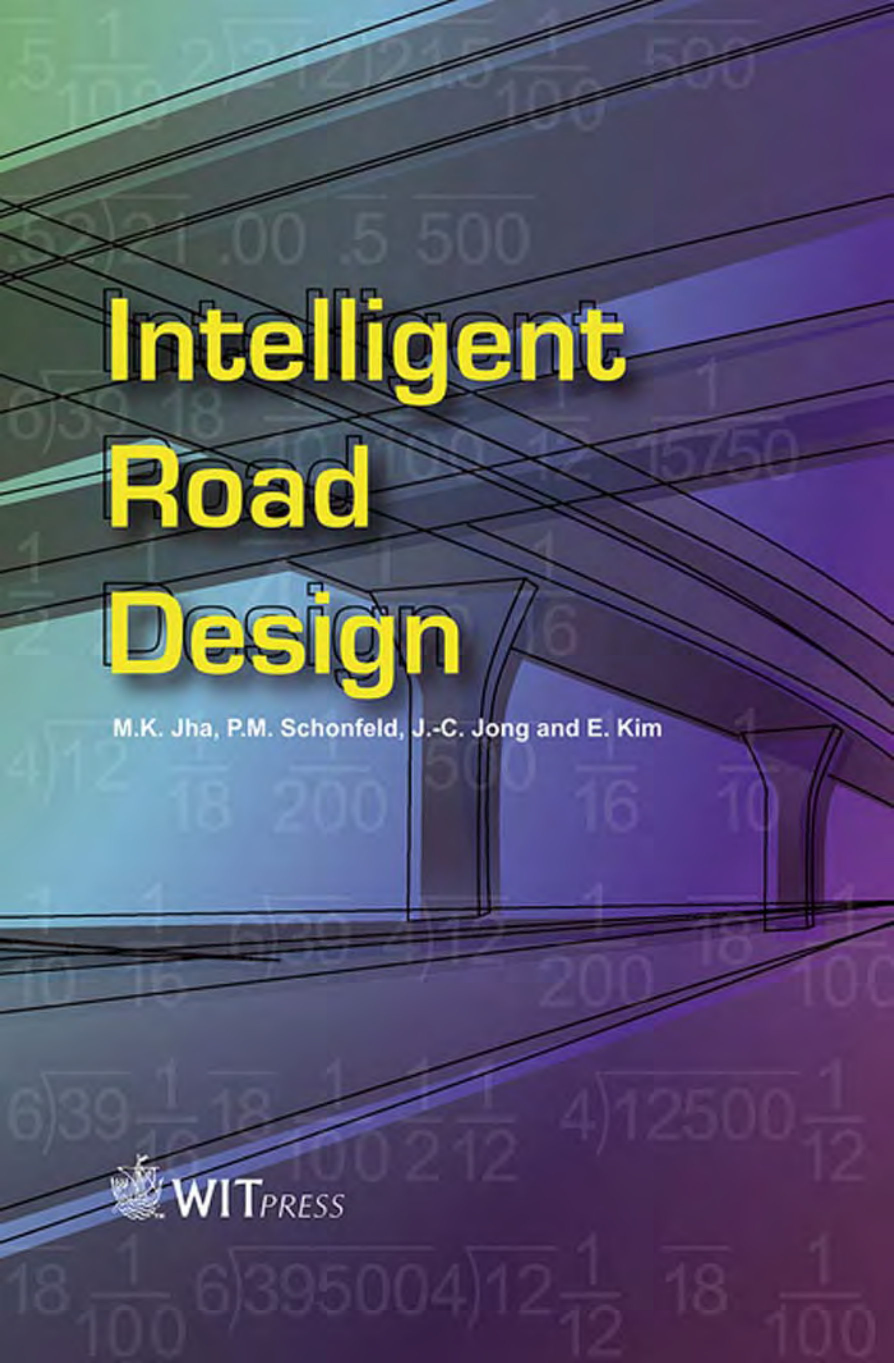


Intelligent Road Design

M.K. Jha, P.M. Schonfeld, J.-C. Jong and E. Kim

WIT PRESS

Intelligent Road Design

International Series on Advances in Transport

Objectives

The objective of this Series is to provide state-of-the-art information on all aspects of transport research and applications. This covers land, water and air systems with emphasis on multi-mode operation. The books in the Series deal with planning operation and management as well as engineering aspects of transport. Environmental topics and sustainability are an important part of the Series. City, national and international transport are covered and encompassing interdisciplinary aspects.

Transport strategies
Planning and funding
Transport and economic issues
Operations and management
Private and public initiatives and policies
Regulation and standardisation
Transport and land use planning
Sustainable transport
Environmental issues
Information technology and electronic aspects
Multi-media and advanced training techniques
Management information systems
Human interface and decision support
Traveller psychology and behaviour
Emerging technologies
Transport and energy
Air transportation

Railway systems
Water and sea transport
Road transport
Urban transport systems
Terminal and interchanges
People movers
Multi-mode systems
Traffic integration
Infrastructure
Scheduling and traffic control
Vehicle technology
Safety and accident prevention
Hazardous transport risk
Hazardous remediation
Transport in extreme conditions
Freight transport

Associate Editors

This page intentionally left blank

Intelligent Road Design

M.K. Jha
Morgan State University, USA

P. Schonfeld
University of Maryland, USA

J.-C. Jong
Sinotech Engineering Consultants Inc., Taiwan (R.O.C.)

E. Kim
University of Incheon, South Korea

WIT*PRESS* Southampton, Boston ™

Intelligent Road Design

Series: Advances in Transport, Vol. 19

M.K. Jha
P. Schonfeld
J.-C. Jong
E. Kim

Published by

WIT Press
Ashurst Lodge, Ashurst, Southampton, SO40 7AA, UK
Tel: 44 (0) 238 029 3223; Fax: 44 (0) 238 029 2853
E-Mail: witpress@witpress.com
http://www.witpress.com

For USA, Canada and Mexico

WIT Press
25 Bridge Street, Billerica, MA 01821, USA
Tel: 978 667 5841; Fax: 978 667 7582
E-Mail: infousa@witpress.com
http://www.witpress.com

British Library Cataloguing-in-Publication Data

A Catalogue record for this book is available
from the British Library

ISBN: 1-84564-003-9
ISSN: 1462-608X

Library of Congress Catalog Card Number: 2005928126

No responsibility is assumed by the Publisher, the Editors and Authors for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

Printed in Great Britain by Cambridge Printing

Contents

PART C: Intelligent intersection and road structure design

This page intentionally left blank

Biography of Dr. Manoj K. Jha

Dr. Manoj K. Jha is Assistant Professor of Civil Engineering at the Morgan State University, USA. He has taught over eight courses in transportation engineering at the Morgan State University and the University of Maryland, College Park. He performs research in transportation system optimization, highway design and maintenance, artificial intelligence, and geographic information systems. He is a member of several professional organizations and technical committees and has given a number of invited seminars. He has published over forty peer-reviewed papers and is very active in transportation research. Currently, he directs a number of doctoral dissertations at the Morgan State University. Dr. Jha received B.E. in Mechanical Engineering from the National Institute of Technology, Durgapur, India in 1991, M.S. in Mechanical Engineering from the Old Dominion University in 1993, and Ph.D. in Civil Engineering from the University of Maryland, College Park in 2000. For additional information please visit Dr. Jha's website at: www.eng.morgan.edu/~mkjha/

Biography of Dr. Jyh-Cherng Jong

Dr. Jyh-Cherng Jong is Senior Research Scientist of Civil, Hydraulic, and Informatics Research Center at Sinotech Engineering Consultants, Inc. He received B.B.A. in Transportation Engineering and Management, and M.S. in the Institute of Traffic and Transportation from National Chiao Tung University, Taiwan. He was granted the fellowship of International Road Federation in 1995 and earned his Ph.D. in Civil Engineering from the University of Maryland, College Park in 1998. Due to his outstanding academic achievements, Dr. Jong was selected as a member of the Phi Tau Phi Scholastic Honor Society and the Honor Society of Phi Kappa Phi. His primary research interests are in the applications of optimization theory, simulation modeling, artificial intelligence, and Information Technology to highway design and railway operation. Dr. Jong is currently the team leader of transportation research group at Sinotech Engineering Consultants, Inc. He has directed several research projects funded by Institute of Transportation and Railway Reconstruction Bureau in Ministry of Transportation and Communications, Taiwan. Dr. Jong is a member of China Road Federation, Chinese Institute of Transportation, Chinese Institute of Engineers, and Rail Engineering Society of Taiwan, where he is also a member of the academic subcommittee.

Biography of Dr. Paul Schonfeld

Dr. Paul Schonfeld is a Professor in the Department of Civil and Environmental Engineering at the University of Maryland, College Park, where he has worked since 1978. With his students he has developed methods for analyzing and optimizing various transportation systems, including highways, public transit systems, inland waterways and air transportation systems. He has advised over thirty Ph.D. students, including the coauthors of this book. He has B.S and M.S. degrees from M.I.T. and a Ph.D. from the University of California at Berkeley.

Biography of Dr. Eungcheol Kim

Dr. Eungcheol Kim is Assistant Professor in the Department of Civil and Environmental System Engineering in the College of Engineering at the University of Incheon, in South Korea, where he teaches courses on highway engineering, traffic engineering, highway planning & alignment design, transportation survey and design, calculus and transportation & logistics. He received a B.A. in Dept. of Urban Planning from the Hanyang University in South Korea and a M.A. in Dept. of Environment Planning, Transportation Major from the Seoul National University in South Korea and Ph.D. in Transportation Engineering, Dept. of Civil and Environmental Engineering from the University of Maryland at College Park. His research interests include highway alignment and design optimization, highway engineering, optimization of transportation systems, capacity analysis and public transit systems operations.

He was the Winner of the Student Paper Competition at the ITE Student Chapter in University of Maryland at College Park in 2000. He has also received two awards for outstanding research from the Mayor of Seoul Metropolitan City in 1996 and from the Chairman of Korean Council of Economic and Social Research Institutes in 2005. He served as an editorial member of the Journal of Korean Society of Transportation from 2002 to 2003. He also serves as a member of Urban Planning Board in such city governments as Incheon, Bucheon and Young-In in South Korea.

Foreword

This book addresses the intelligent concepts of road design not found in other textbooks. Road design is an ancient endeavor, however, the advent of motorized vehicles in the early 1900s required paved roads. Road planners and designers in the 1950s foresaw a tremendous growth in coming decades and therefore developed road planning and design concepts to accommodate vehicles of different dimensions to improve driver and passenger safety, comfort and convenience. In the last two decades most urban roads have experienced tremendous growth in traffic leading to frequent congestion and delays.

Due to shrinking right-of-way and limited highway budgets in recent years roadway planners and designers have been constantly exploring innovative methods of road design. Moreover, in recent years highway agencies have often found it difficult to secure adequate funds for road construction due to conflicting public opinions and political views. Therefore, having an intelligent road design model that can quickly optimize horizontal and vertical alignments will allow rapid evaluations of many competing alignment alternatives which should result in faster political and public approval. This book extensively discusses how such a road alignment optimization model can be developed and applied in real case studies.

The book is based on over eight years of research by the authors in intelligent road design and alignment optimization. It should be considered an advanced textbook in road design and will be appropriate for road planners, designers, senior undergraduate students and graduate students. The authors have extensively published the research results from their intelligent road design and alignment optimization work in leading transportation journals. The readers are strongly encouraged to consult those publications and also keep an eye on the forthcoming publications of the authors to stay up-to-date with the future developments in intelligent road design. The authors wish to acknowledge Dr. David Lovell and Min-Wook Kang for some of the material that they contributed to the book.

For easy reading the book has been conveniently divided into three parts. In Part A we develop theoretical foundations and techniques for intelligent road design. In Part B we develop models and algorithms for optimizing road alignments. In part C we discuss intersection design along with bridges and tunnels. It is hoped that the book will lay the foundations for intelligent road design and will be widely used by researchers and practitioners throughout the world.

The authors
2006

Dedicated to my maternal grandparents, Taranand Jha and Yogmaya Devi, my parents Devendra Jha and Kusum Devi, and my wife Amarjit Kaur.

M.K. Jha

Dedicated to my parents, Marcel and Elise, and to my wife Claudia.

P. Schonfeld

Dedicated to my parents, Kuo-Shang and Pao-Chen, my wife Wen-Yuh, and my two sons, Hou-Ting and Hou-Zhe.

J. -C. Jong

Dedicated to my parents, Taekyung Kim and Okhee Kang, my wife Jongheui Jung, my daughter Chaelin, and my son Seongchan.

E. Kim

This page intentionally left blank

Part A

Theoretical foundations and techniques for intelligent road design

This page intentionally left blank

Chapter 1

Introduction

1.1 Background and motivation

Highway alignment design is a very complex and repetitive process. Designers must select an economical path based on topography, soil conditions, socioeconomic factors, and environmental impacts (such as air pollution and noise). This path must also satisfy a set of design constraints and operational requirements. In the process of constructing or relocating an existing highway, the engineers often face a wide variety of factors and a huge number of alternatives. Due to inadequate computer performance, mathematical models and information, traditional alignment design usually consists of a series of phases, starting from a broad area, then narrowing down to several possible transportation corridors, and finally focusing on the detailed alignment designs in the selected corridor, including horizontal alignment and vertical alignment. This procedure requires professional judgments in various fields including transportation, economics, ecology, geology, environment, and politics. It has proven to be a time-consuming and tedious process.

It would be desirable to formulate any single design phase or several design phases jointly as an optimization problem. With the use of reasonable mathematical models and high-speed computers, highway engineers can considerably speed up the design process and arrive at a very good design rather than a merely satisfactory solution. In fact, optimization of road alignment has attracted much research interest over the past three decades, and several models have been developed. Studies such as OECD (1973), Shaw and Howard (1982), and Fwa (1989) indicate that an optimum alignment derived from mathematical models and computer programs can yield considerable savings in construction cost when compared with a conventional manual design.

Existing models, although performing well in certain aspects, still have considerable defects and are not widely utilized in real-world applications. The difficulties in developing a robust and easily applicable model are due to the complex cost structure associated with an alignment and the requirements for satisfying complex constraints. Moreover, since the problem itself has a continuous search space, there are infinitely many alternative alignments. As the terrain or the cost map over which the alignment is optimized becomes precipitous and irregular, any small change in the alignment will result in a significant change in the total cost. Thus, a reasonable model as well as an efficient solution algorithm are still needed for the optimization of highway alignments.

1.2 The highway planning process

Highway planning process requires a comprehensive evaluation of future conditions in the geographic region which may be impacted by the construction of a new highway. For example, construction of a new highway may change the land accessibility and land use pattern in its area of influence. Such changes should be considered very carefully.

For constructing a new highway we hope to select the minimum cost (or maximum net benefit) path while satisfying all the constraints. Constraints affecting the alignment selection include the following ones:

(1) Design Constraints:

- Design Speed
- Lane Width
- Shoulder Width
- Bridge Width
- Structural Capacity
- Horizontal Alignment
- Vertical Alignment
- Curve Length
- Grade
- Stopping Sight Distance
- Cross Slope
- Superelevation
- Vertical Clearance, and
- Horizontal Clearance

(2) Environmental Constraints:

- Hydrologic constraints such as impacts of floodplains and wetlands
- Air and water contamination as well as changes in noise level

Good planning should take all significant alignment sensitive costs into consideration. Formulation of costs may be a difficult process requiring considerable time and effort. The selection of the final alignment may require comparison of several alternative alignments. Since numerous factors are involved in the decision process, the computation required may be enormously large. Current methods of selecting a best highway alignment are primarily manual, requiring significant time and resource allocation. Therefore, the process of finding the best highway alignment can greatly benefit from automated computerized methods for selecting best highway alignment.

1.3 Intelligent road design

This book is mainly concerned with planning and designing roads in an intelligent process with the aid of computer algorithms. The emphasis will be placed on developing computer models for optimizing horizontal and vertical highway alignments while satisfying design, operational, and environmental constraints.

A typical highway design problem is to select an economical path to connect two cities (or interchanges or points on a map). Even in designing a network, the problem can also be reduced to finding an alignment connecting each successive pair of nodes, as shown, for example, in Figure 1.1.

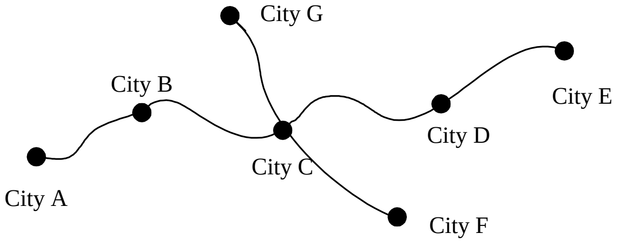


Figure 1.1: A simple highway network.

In it, there are two major alignments: one is to connect city A to city E through cities B, C, and D; the other is to connect city F and city G via city C. Instead of designing these two alignments directly, it is often desirable to optimize the alignment for each adjacent city pair.

In developing intelligent models for roadway design we will extensively examine the characteristics of horizontal and vertical alignments, discuss the applicability of Geographic Information Systems (GISs), and finally discuss how structures such as bridges and tunnels can be modeled in alignment optimization.

Note that in designing highway alignments, there are thousands of cost items associated with an alternative and a great number of constraints and requirements to be satisfied. The attempt in this book is not to quantify all of the cost items and formulate all design factors. Rather, only major cost items and important constraints will be considered here. The scope of this book will include developing a reasonable cost function and an important set of design constraints. The focus will be on the modeling approach and the search algorithms. We can expect that after a good modeling approach is selected and a good algorithm is developed, we will be able to modify and formulate more comprehensive and detailed objective functions and constraints. We should also note that an algorithm which can minimize a total cost function of an alignment alternative can also maximize a net benefit function. Such modifications will be left for future extensions.

1.4 Highway design models

The purpose of highway design models is to assist highway planners and designers in evaluating a number of alignments when considering the construction of a new highway between any two points. Since there are numerous possible alternatives available to connect two points in space, selection of the best alignment is a very difficult task.

1.4.1 The cost models

The cost models generally estimate the total cost of an alignment for a given set of input data. These models are primarily developed for estimating the total cost of a highway alignment by capturing all significant variables that might be used for making policy decisions.

A comprehensive highway cost model was developed by Moavenzadeh *et al* (1973). The model was a result of a study initiated by the World Bank (Watanatada *et al*, 1987) in 1969. This was the first prototype model for interrelating life-cycle costs of highway construction, maintenance, and vehicle operation. The objective of the model was to estimate significant highway costs for a given set of data without performing any optimization. The input to the model was huge (Table 1.1) and not very user-friendly. The model nevertheless had the potential to work with an optimal search algorithm if a proper objective function and set of constraints were developed.

Another study, (Watanatada *et al*, 1987), formulated models for highway design and maintenance standards from European and U.S. experience that could be applied in developing countries. The focus of the study was on comparing economically viable options for highway construction. The basic task in the model was to investigate the relations between total life-cycle costs (including construction, maintenance, and road user costs) and road design and maintenance standards, which could be used in making policy decisions. Table 1.2 describes the costs used in the model and the independent input variables on which the costs depend.

The Watanatada (1987) model used simulation instead of optimization for obtaining a minimum cost route. The arguments for using simulation instead of optimization were that:

- some of the model relations had highly complex non-linear forms that prevented the application of optimization techniques.

- the alternative approach of simplifying the relations to a form more amenable to formal optimization could constitute a critical departure from the complex realities of the physical world.

Table 1.1: Input data for the Moavenzadeh model (1973).

Data Categories	**Data Type**
Roadway configuration data	Alignment
	Profile
	Template
	Pavement
	Shoulders
Policy data	Economic
	Foreign exchange
	Maintenance
Environmental data	Topography
	Soil and ground cover
	Hydrology
	Transport haul distances
Productivity and cost data	Construction
	Maintenance
	Operating and user costs
Vehicle characteristics	Staging strategies
	Traffic demand data

Table 1.2: Types of costs and their relations to independent variables in the Watanatada model (1987).

Costs	**Independent Input Variables**
Construction	Terrain
	Soils
	Rainfall
	Geometric design
	Pavement design
Maintenance	Road deterioration (Pavement design, Climate, Time, Traffic)
	Maintenance standards
Road user costs	Geometric design
	Road surface condition
	Vehicle speed
	Vehicle type

1.4.2 The optimization models

During the 1960’s and 1970’s a number of computerized models were introduced for optimizing horizontal and vertical alignments in the U.S., Australia, and some European Countries (Athanassoulis and Calogero, 1973; Hogan, 1973; OECD, 1973; Puy Huarte, 1973). The progress in the developing countries toward exploiting computerized methods for highway design optimization, however, has been slow (Watanatada *et al*, 1987).

Computer performance limited the effectiveness of solutions to earlier highway design optimization formulations. Data collection was primarily done by field survey since Geographic Information Systems (GIS) were not available at that time. Due to the advent of sophisticated computers and GIS technology, it is now possible to solve problems accurately and efficiently that were unsolvable during the 1960’s and 1970’s.

The commonly used objective in highway economic analysis is to minimize total cost (or maximize total net benefit) associated with the alternatives. It is important to consider all dominating and sensitive costs. A cost is said to be dominating if it accounts for a higher percentage in the total cost function. A sensitive cost is one that is sensitive to the changes of the design features of the alignment.

Due to increasing congestion and roadway accidents, some models have only focused on reducing the number of accidents by choosing among various design alternatives. In the model developed by Vogt and Bared (1998a&b) data have been collected from the states of Washington and Minnesota to develop a regression equation which can be used to predict future accidents along a new highway. Thus a combination of geometric features of the road such as horizontal and vertical curvature and roadside characteristics such as width of shoulders and traveled portion of the road can be selected that would yield the minimum number of future accidents.

In standard highway optimization models, accident costs are part of the objective function which is to be minimized. Other cost components such as right-of-way and construction costs are also part of the objective function and accident costs are not weighed any higher or lower. In fact it is difficult to assess tradeoffs between safety and social, environmental, and economic impacts. Additionally, it is difficult to establish realistic relations between alignment and accidents since the accidents, in many cases, are dominated by human behavior which could be influenced by the alignment.

Chapter 2

Traditional methods for alignment optimization

This chapter briefly reviews the traditional approaches for optimizing highway alignments. The chapter is divided into 7 sections. First, the costs associated with highway transportation are examined for the purpose of establishing a comprehensive evaluation function. In section 2.2, the relations between these cost items and alignment configurations are investigated. The third section presents a set of design constraints and operational requirements that must be followed in designing highway alignments. The advantages and disadvantages of existing models for optimizing highway alignments are compared in sections 2.1 through 2.1. Finally the necessary conditions of a good optimization model for highway alignments are outlined in the last section.

2.1 Costs associated with highway transportation

In highway economic analysis, the most commonly adopted criterion is the costs (or benefits) associated with the alternatives. As can be seen in sections 2.1 through 2.1, when optimizing highway alignment, the objective is usually to minimize the total cost of the proposed alternative. According to previous studies (Winfrey, 1968; OECD, 1973; Wright, 1996), the major costs of highway transportation can be classified into several categories. They are summarized in Table 2.1 and will be discussed below:

2.1.1 Planning, design, and administrative costs

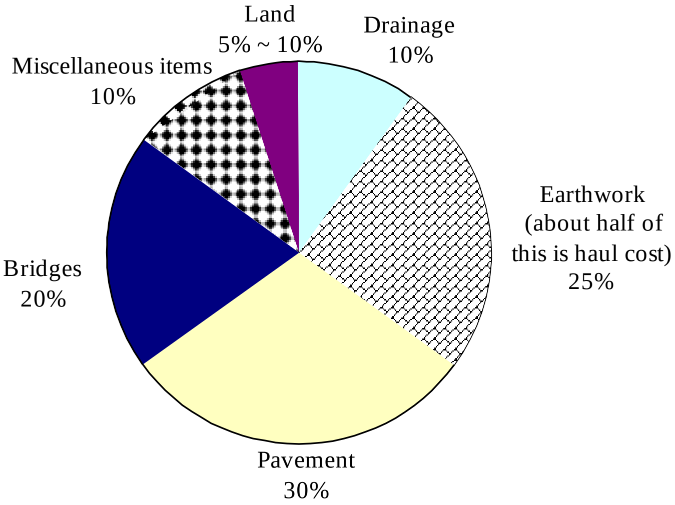
A highway project usually includes a series of steps, initialized by a feasibility study, then followed by planning, design, construction, and finally operation and maintenance. The planning and design costs occur mostly at the early stage of a highway project, while administrative cost may occur throughout the lifetime of the highway. Since these costs are believed to be insensitive to highway alternatives, they are usually not considered in alignment optimization.

Table 2.1: Classification of highway transportation costs.

Classifications		**Examples**
Planning, design, and administrative costs		Consulting and supervision costs
Construction costs		Earthwork, Pavements, Right-of-way
Operation and maintenance costs		Pavement, Mowing, Lighting
User costs	Vehicle operating costs	Fuel, Tire wear, Depreciation of vehicles
	Travel time costs	Vehicle hours times unit value of time
	Accident costs	Estimated accident rates times unit accident cost
Social and environmental costs		Noise, Air pollution, Wetland loss

2.1.2 Construction costs

Construction costs can include many detailed items. For example, the Maryland State Highway Administration (SHA) classifies construction costs into 8 categories, each of which includes various detailed cost items. For final design of a highway project, it is necessary to estimate the cost for each detailed element for the purpose of calculating and controlling budgets. However, at the stage of planning and preliminary design, construction costs are usually divided into fewer major components. In the studies of OECD (1973) and Chew *et al* (1989), the major construction costs are drainage, earthwork, pavement, bridges, miscellaneous items, and land, whose percentages are shown in Figure 2.1.



Source: adopted from OECD (1973) and Chew *et al* (1989)

Figure 2.1: Major components of construction costs.

The percentage of each component in the total construction cost is not fixed but depends on the highway's location. For example, the Maryland SHA is currently constructing a new multi-lane highway from I-95 to MD 295 (Baltimore-Washington Parkway). The cost of right-of way is over 35% of total cost (Glendening, 1997), which is much higher than that shown in Figure 2.1. In general, a highway through an urban area may have a higher percentage of land acquisition cost, whereas a highway over a mountainous area will have a higher percentage of earthwork cost. Construction costs per centerline mile for different types of highways in different areas of the United States can be found in Winfrey (1968). A summary of the ratios of these major components to total construction cost in a list of different countries is presented in OECD (1973).

2.1.3 Maintenance costs

According to AASHTO (1987), highway maintenance costs contain at least 8 classifications, including roadway surfaces, shoulders and approaches, drainage, roadside features, bridges and tunnels, highway appurtenances, snow and ice control, and traffic control devices. OECD (1973) indicates that the net present value of maintenance costs discounted over 30 years is about 5% of the total construction cost. In the United States, highway statistics (Teets, 1997) show that about 26% of 1995 highway expenditures for all levels of governments were spent on maintenance and services.

2.1.4 User costs

The major components of user costs usually include vehicle operating cost, the value of travel time, and traffic accident cost. For vehicle operating cost, basically only those components that depend on the travel mileage are included in highway economic analysis (Wright, 1996). Other operating costs such as registration fees, parking expenses, insurance and the time-dependent portion of depreciation may be excluded. AASHTO (1977) published a manual for estimating user cost of highway and bus transit improvement, where several nomographs are used to estimate user costs.

According to OECD (1973), the net present values of vehicle operating cost discounted over 30 years range from 300% to 1000% of construction costs, depending on the forecast traffic volume.

The travel time value is usually derived by multiplying total vehicle-hours by unit value of time. As indicated in the AASHTO (1977) "Red Book", the unit value of time depends on the type of the trip. However, a travel time value of $3 per vehicle-hour based on a value of $2.4 per person-hour and 1.25 persons per vehicle was generally used for passenger cars, while values of $7 and $8 are recommend for 2A and 3-S2 trucks. Since these data are based on the previous price levels, they must be updated to current price level before being applied.

Traffic accidents represent a great economic loss, and are regarded as a cost of highway transportation. Accident cost is usually estimated by multiplying accident rates and the average cost per accident. Since the causes of accidents are many and complex (refer to Garber, 1996 for examples), it is very difficult to specify the exact relation between accident rates and alignment configurations. For this reason and due to the lack of empirical data, accident costs are usually excluded in the evaluation for a newly built highway. However, in highway improvement projects, the reduction in accident cost is considered a large part of user benefits. When a section of highway is labeled as a hazardous location, the improvements of that particular location will be implemented, and a before-after analysis is usually conducted based on the empirical data.

Pline (1992) indicates that about 70% to 90% of all accidents involve some form of driver error, but highway design can significantly improve the safety and reduce driver error. Sufficient sight distance, clear shoulders, gentle side slopes and absence of fixed objects give the driver enough time to react to a potential accident.

2.1.5 Social and environmental costs

The construction of the highway system will change the land use patterns in the adjacent area, and cause environmental impacts such as air pollution, water pollution, and noise. There is an increasing public awareness of the impacts of highways and other public projects on the environment. In some extreme situations, environmental issues may become the most critical factors in highway design, even dominating the other cost items associated with the alignment (see Monchak, 1996 for an example). In the United States, the Federal Highway Administration (FHWA) has issued regulations which highway agencies must follow to reduce the environmental impacts to a minimum while developing federal-aid highway projects. The applicants are required to prepare an environmental impact statement (EIS) or environmental assessment (EA) in order to get approval for their projects.

Social and environmental costs are usually considered at planning stages for selecting the transportation corridor or determining preliminary alignments. At the detailed design phase (especially for vertical alignment), they may be excluded because the configuration of the alignment has fewer effects on environments. Since social and environmental costs are difficult to quantify, it is hard to incorporate them into the cost function for evaluation. For this reason, various studies have tried to estimate unit environmental costs. DeCorla-Souza and Jensen-Fisher (1994) summarized from their references that the unit environmental costs per vehicle mile traveled (VMT) are in the range shown in Table 2.2. With these estimates, it would be possible to incorporate social and environmental costs into the objective function of a highway design model. However, as we can imagine, social costs are not only influenced by VMT but also by a road’s location. Thus, an optimization model should be able to avoid environmentally sensitive locations, such as flood plains, coastal zones, wet lands, etc.

Table 2.2: Unit environmental costs per VMT.

Classifications	Unit Cost (Cent/VMT)
Air pollution	1.0 to 7.2
Noise pollution	0.1 to 0.3
Water pollution	0.16 to 0.2
Oil extraction, distribution and use	1.5 to 4.0
Land use	3.5 to 6.3
Solid and chemical waste disposal	0.2

Source: Adopted from DeCorla-Souza, P. and Jensen-Fisher, R. (1994)

2.2 Relations between highway costs and alignment configurations

In modeling optimization of highway alignment, it is very important to capture the relations between the costs and the alignment configurations because all costs must be suitably quantified in terms of road design features so that the total cost function can be minimized with respect to the alignment characteristics. OECD (1973) classifies some highway costs into plan-dependent, profile-dependent, and route-dependent costs, as summarized in Table 2.3.

In Table 2.3, plan-dependent costs are those costs that depend on the location of the alignment in the horizontal plan. Profile-dependent costs depend on the elevation of the alignment at individual points along the route. Route dependent costs at a particular point along the route are a function of the features of the design at many other points along the route.

From another point of view, highway costs can be further classified as location-dependent, length-dependent, area-dependent, volume-dependent, and VMT-dependent costs. Location-dependent costs depend on the location and the covered area of the alignment at a particular place. Land acquisition cost, environmental costs, and the highway base construction cost, which is affected by the soil conditions, are examples of location-dependent cost. Guard rails, drainage, and fences can be regarded as length-dependent costs. Area-dependent costs are those costs that depend on the coverage area of the alignment. Pavement cost is a typical area-dependent cost regardless of base construction. If the width or the number of traffic lanes remains the same along the alignment, it would be possible to convert area-dependent costs into length-dependent costs. A good example of volume-dependent cost is earthwork cost, which depends on the volume of cut and fills. Vehicle operating cost and some portion of social and environmental cost are examples of VMT-dependent costs. With these classifications, it would be easier to formulate highway costs in terms of the design features of the alignments.

Table 2.3: Classifications of costs as plan-, profile-, and route-dependent.

Plan-dependent	Profile-dependent	Route-dependent
Drainage (proximity to outfalls)	Drainage (slopes and pipe sizes)	Drainage (flow)
Earthworks (soil type)	Earthworks (volumes)	Earthworks (out of balance volumes and haul)
Pavements (soil type)		
Bridges (nature of obstacles, soil type)	Bridges (height)	
Miscellaneous items (nature of ground to be cleared and fenced)	Miscellaneous items (area of land to be cleared)	
Land (type, use and value)	Land (area depends on width between fence lines)	Land (severance costs)
Vehicle operation (length of route)	Vehicle operation cost (gradient and height rise)	Vehicle operation cost (speed and fuel)
Social costs (areas of aesthetic and political value)	Social costs (road above or below ground - noise and visual intrusion	

Sources: Adopted from OECD (1973)

Different types of costs will favor different alignment configurations. For example, length-dependent costs and VMT-dependent costs tend to straighten the alignment, while location-dependent costs tend to favor more indirect and circuitous alignment. The optimal alignment configuration will balance different types of costs.

Another important aspect of highway costs is whether a cost is dominating or sensitive. A cost is said to be dominating if it accounts for a high percentage in the total cost function. For example, vehicle operating cost might be a dominating cost because it ranges from 300% to 1000% of construction cost, according to the discussion in section 2.1. A sensitive cost is one that is sensitive to the changes of the design features of the alignment. Otherwise, it is said to be insensitive.

A dominating cost is not necessarily a sensitive cost. OECD (1973) indicates that in optimizing vertical alignment, vehicle-operating cost is insensitive to design changes even though it is a dominating cost because vehicle descents roughly compensate for climbs. Obviously, in modeling alignment optimization, we want to capture all dominating and sensitive costs. In some extreme cases, however, if a cost is believed to be minor or insensitive, it may be excluded from the optimization model.

2.3 Constraints and operational requirements in highway alignments

There are a great number of constraints and operational requirements to be followed when designing a highway. These constraints have been developed for a long time, and published in many textbooks and reports (for example, AASHTO, 2001; Underwood, 1991; Schoon, 1993; Wright, 1996; and Pline, 1992). Here only important constraints are outlined.

2.3.1 Traffic considerations

It is fundamental that the design of roads be based on the behavior of all road users. For vehicles, some important traffic considerations and their purposes are listed in Table 2.4.

In this book, *AADT*, number of traffic lanes, and design speed are assumed to be determined in advance so that the main concern is the optimization of total cost with respect to the design features of alignment.

Table 2.4: Traffic considerations in highway designs.

Measure	**Purpose/Annotation**
Annual Average Daily Traffic (*AADT*)	As a basis for economic analysis
Design Hour Volume (*DHV*)	As a basis for determining no. of lanes
K Factor	For analyzing *DHV*, given *AADT*
Traffic composition (*T* Factor)	For analyzing the percentage of heavy vehicles in the traffic stream
Directional distribution (*D* Factor)	For analyzing one-way volume, given two-way traffic
Design Speed V_d	• Maximum safe speed that can be maintained over a specific section of highway • Numerically it is equivalent to the 85th percentile of speed profile

2.3.2 Horizontal alignment

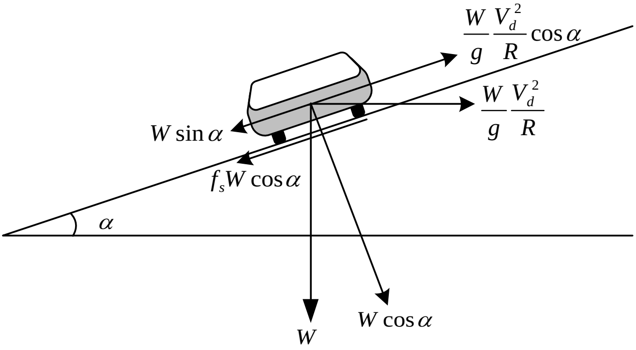
The horizontal alignment of a road usually consists of a series of straight (tangent) lines, circular curves, and possible spiral transition curves. The most important constraints on horizontal alignments are:

(1) Minimum Radius

Minimum radius depends on the design speed, superelevation, and coefficient of side friction. Their physical relations are illustrated in Figure 2.2.

(2) Sight Distance on a Curve

Sight distance on a horizontal curve is affected by the radius of the curve and the lateral distance between the obstruction and road side, as illustrated in Figure 2.3.


$\frac{W}{g}\frac{V_d^2}{R}\cos\alpha$
$\frac{W}{g}\frac{V_d^2}{R}$
$W\sin\alpha$
$f_s W\cos\alpha$
α
W
$W\cos\alpha$

$$W\sin\alpha + f_s W\cos\alpha = \frac{W}{g}\frac{V_d^2}{R}\cos\alpha$$

$$e + f_s = \frac{V_d^2}{gR}$$

$e =$ superelevation
$f_s =$ coefficien t of side friction
$R =$ curve radius
$V_d^2 =$ design speed

Figure 2.2: The physical relations among design speed, superelevation, coefficient of side friction, and curve radius.

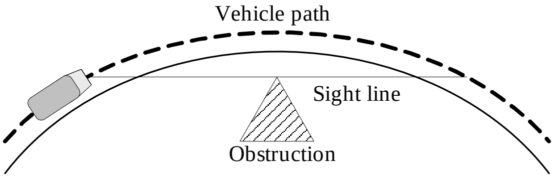


Figure 2.3: Geometry of horizontal sight distance.

2.3.3 Vertical alignment

The vertical alignment of a road usually consists of a series of grades joined to each other by parabolic curves. The most important constraints on vertical alignments are:

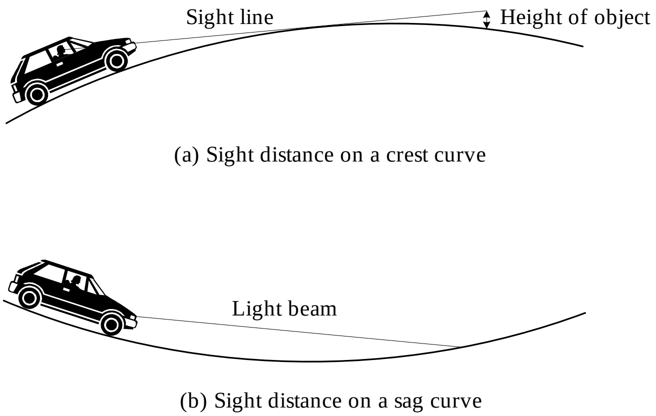
(1) Maximum gradient

Maximum gradient depends on the nature and importance of the road, design speed, topography, design controls (e.g., the elevations of underpass or overpass of existing structures), and whether the road is in a rural or urban area.

(2) Sight distance on crest vertical curve

On crests, the vertical alignment itself causes sight restriction. As illustrated in Figure 2.4, the sight distance is controlled by the height of driver’s eyes and the height of the object.

(3) Head light sight distance and motorist comfort on sag vertical curves
On sag curves, the worst situation occurs at night when the sight line is limited within the visible area of headlight, as shown in Figure 2.4.



(a) Sight distance on a crest curve

(b) Sight distance on a sag curve

Figure 2.4: Geometry of vertical sight distance.

2.4 Models for optimizing horizontal alignment

The models for optimizing horizontal alignment are more complex and require substantially more data than those for optimizing vertical alignment (OECD, 1973). The main reasons are that the design of horizontal alignment involves more political, socioeconomic, and environmental issues, and that the costs associated with a horizontal alignment are significantly affected by the vertical alignment. However, the cost reductions due to the optimization of horizontal alignment are substantially higher because most of major costs such as land cost, construction cost, social and environmental cost are very sensitive to the configuration of the horizontal alignment.

The progress in developing models for optimizing horizontal alignment is slow and their number is relatively small. In the literature, three basic approaches are noted: calculus of variations, network optimization, and dynamic programming. These are discussed in turn.

2.4.1 Calculus of variations

The calculus of variations has been developed to quantify the optimization process for a certain class of optimization problems. Many applications have led to the theory of optimal control. The basic problem in calculus of variations is to seek a curve connecting two end points in space which minimizes the integral of a function (Wan, 1995). To some extent, the optimization of highway alignment is similar to the typical problem in calculus of variations. Howard, Bramnick, and Shaw (1968) borrowed the idea to derive the Optimum Curvature Principle (OCP) for highway horizontal alignment. A similar application of the calculus of variations in transportation can be seen in Thomson and Sykes (1988), where a maritime route is optimized through a dynamic ice field.

The basic approach of the OCP is to hypothesize the existence of a criterion function defined in the region concerned between two cities or points where it is contemplated to build a highway. In other words, it is assumed that there exists a continuous cost surface above the 2 dimensional region of interest. The OCP then generates a route that winds along the low cost valleys and skirts the high cost mountain to connect two end points.

The OCP states that the curvature of an optimally located highway at each point is equal to the logarithmic directional derivative (percentage rate of change) of the criterion function (local cost function) perpendicular to the route. During optimization, several routes are initialized from the start point in different directions. The sub-optimal routes are obtained by numerical integration of the OCP for each starting angle, and by selecting the paths that terminate at the end point. The optimal route is the best of this discrete set.

Shaw and Howard (1981) proposed two numerical integration methods for applying the OCP. One is to integrate the criterion function along a sequence of circular arcs joined together (arc of circle algorithm). The other uses a Taylor series expansion of degree 3 to approximate the path (intrinsic equation procedure). An example was utilized to test the performance of the two methods. The result shows that the intrinsic equation algorithm is superior to the arc of circle algorithm. Shaw and Howard (1982) further applied the OCP to find the horizontal alignment of an expressway in flat south Florida, where construction cost was optimized. They also conducted an error analysis to establish error bounds when the true answer is unknown. The numerical example shows that a saving of nearly 5% in the cost over the straight line was gained by an increase in length of only about 0.5%.

Although Howard, Bramnick, and Shaw (1968) stated that the determination of a local cost function is not a weakness of the OCP, that determination still seems to be a major problem in applying the OCP. In practice, the authors suggested that Bicubic Spline interpolation over several discrete points be used to derive the continuous smooth surface of the local cost function. However, the local cost function may not be continuous since at least the cost of right-of-way is usually not continuous between different zones or land use patterns. Thus additional processing is required. Moreover, the local cost function in the OCP implies that the highway cost at any individual point is determined, which is somewhat unrealistic. As discussed in section 2.2, the highway costs depend on various factors, including length, location, area, volume, and VMT. It seems that there are some approximations and assumptions behind the determination of the local cost function in the OCP.

Regardless of the above disadvantages, the optimum route derived by the OCP is continuous and a global optimum is guaranteed. These can be regarded as the main advantages of this approach.

2.4.2 Network optimization

The basic idea of this approach is to formulate the optimization of horizontal alignment as a network problem, in which the alignment is represented by the arcs connecting from the start point to the end point. Then some well-developed network optimization techniques such as a shortest path algorithm are used to solve the problem.

According to OECD (1973), early studies at MIT and Miami University belong to this category. A similar concept was used by Turner (1971) to develop the Generalized Computer Aid Route Selection (GCARS) system. The GCARS system is initialized by establishing cost models (a surface covering the region of interest) for right-of-way, pavement, and earthwork (maintenance and operating costs are not considered and left to engineers for judgments). Total cost is then summed by a linear combination of different cost components with relative weightings. A grid network is finally formed from the cost model matrix by joining all nodes and assigning the cost of traversing it to each link.

Due to some obvious drawbacks in the early version, the GCARS system was further improved by allowing movement on the diagonals of the network matrix, modifying the shape and size of network grids, and incorporating the environmental impact into the model (Turner, 1978).

Athanassoulis and Calogero (1973) also employed network optimization techniques to approach the optimal route problem. Unlike Turner's model, where link costs are calculated by averaging the costs of two end nodes, Athanassoulis defined cost lines (like river, bridge) and cost areas (such as lake, wetland) as a basis for calculating link costs. Then the cost between any pair of nodes was accurately calculated by the summation of the length in each cost area multiplying the associated unit cost. Athanassoulis's model is the only one found in this search, which explicitly and precisely estimates link costs.

For finding the optimal route location, instead of utilizing the shortest path algorithm, Athanassoulis formulated it as a modified transportation problem, where one unit of goods is shipped from the start point to the end point. (In fact, this is the so-called transshipment problem or generalized network flow problem.) All nodes in the resulting network are considered as possible sources as well as possible destinations so that the optimal route can move in any direction to join any two nodes without passing through the other nodes. The same network will take more time to be solved by Athanassoulis's approach than by Turner's and more storage space is required due to the dimension of the embedded cost matrix.

It must be mentioned that neither Turner nor Athanassoulis considered the vertical profile in their models, although Turner's GCARS system did consider earthwork cost. Later, Parker (1977) developed a two-stage approach to optimize a route corridor subject to a gradient constraint. Although that approach was claimed to simultaneously optimize both horizontal and vertical alignments, it is more reasonable to categorize it as a 2-dimensional alignment optimizer because the vertical alignment was predetermined before optimizing the horizontal route. In Parker's approach, the region of interest is first divided into sub-squared zones where the mean elevation of each zone is taken to be the ground elevation at the zone centroid. A smooth surface is then constructed through the zone centroid elevations such that any horizontal alignment from the origin to the destination will intersect the surface with a feasible vertical alignment with respect to gradient constraints. The smoothed surface is derived by a binomial expansion of a predetermined order in both dimensions of horizontal plan. Then a linear programming regression model is run to minimize the residuals (deviations between projected surface and ground surface) with respect to the coefficients in the binomial function of the projected surface, while subject to gradient constraints and some other system equations. To some extent, the surface smoothing procedure is slightly similar to Turner's work, where the projected surface is derived with a linear regression model without considering gradient constraints.

In the second stage of Parker's approach, the centroids of the sub zones form a grid network. Then a multiple-path program is used to derive a set of best routes which minimize the absolute residuals (which can be regarded as a measure of earthwork cost). By its nature, the second stage is also similar to Turner's approach, but Turner considers more cost components in the cost function.

Another study by Trietsch (1987a) presented a family of methods for the preliminary design of highway alignment. The methods were proposed for optimizing 3-dimensional alignment, but the models are 2-dimensional for the same reasons mentioned above. Trietsch's paper includes 16 models with different structures and cost estimation procedures. Some of them fall into the network optimization category, while the others belong to the class of dynamic programming models (discussed in the next subsection). Trietsch (1987b) further applied his results to a comprehensive design of highway networks, which is capable of satisfying all bilateral (two-way) transportation demands and minimizing total cost.

Four basic search grids were proposed by Trietsch: rectangular, square, ellipse, and honeycomb. The rectangular grid is the basic search grid mostly used in a dynamic programming approach, but can also be formulated as a network problem. The apparent drawback of the rectangular grid in a dynamic programming model is that it is incapable of handling backtracking alignments, which are sometimes necessary in mountainous terrain. The other three network structures are proposed for possible backward bends with different detailed level of moving angles. The most sophisticated one, honeycomb grid, allows twelve different moving directions at each node. Since early studies in alignment optimization do not take the curvature constraints into account, Trietsch presented a modified shortest path algorithm to force the search direction at each node within a limited range of angles.

The optimal alignment derived by the network approach is a piecewise linear trajectory, which is very rough for highway alignment. In fact it is a corridor rather than an alignment. In Turner's model, the node size is about 1 km × 1 km, whereas in Trietsch's honeycomb approach, the diameter of the hexagons is around 500 meters. If a more accurate alignment is desired, the number of nodes must be increased and the size of the node must be decreased so that the network can cover the whole region of interest. However, the resulting network problem quickly becomes too large to solve.

Other implied shortcomings of this approach are the calculation and storage requirements for link costs. To apply shortest path algorithm, all link costs must be determined before performing the search. If the resulting network is large, the calculation efforts and the computer storage space (usually for a matrix) are considerable.

2.4.3 Dynamic programming

Dynamic programming is developed for solving the optimization problems of complex and large-scale systems. The principal assumption of dynamic programming is that a problem can be divided into a number of sub-problems (stages) and that the contributions to the objective function value from each sub-problem are independent and additive.

Dynamic programming was widely used in optimizing highway alignments, especially for vertical alignment (see section 2.5). The stages in a dynamic programming model for optimizing horizontal alignment are usually evenly spaced lines perpendicular to the axis connecting the start and end points of the alignment. At each stage, the states are the nodes or grids on the perpendicular line. During the search, the objective function is usually evaluated from the last stage back to the first stage. To ensure the resulting alignment satisfies the curvature constraints, only a limited number of nodes in the next stage are permitted to connect to the node at the current stage. Studies using dynamic programming for optimizing horizontal alignment and 3-dimensional alignment (which can also be applied to optimize horizontal alignment) include Trietsch (1987a), France (see OECD, 1973), Hogan (1973), and Nicholson (1976).

By nature, a dynamic programming model for optimizing horizontal alignment can also be formulated as a shortest path problem, but the reverse is not always true. When compared with the shortest path algorithm, the advantage of using dynamic programming is its efficiency and lower storage requirements, because it assumes that the decision cost (for alignment, it is the link cost) at each state only depends on the remaining stages but is independent of previous stages. (This idea is known as the principle of optimality.) Therefore, the associated link costs can be deleted from memory after finishing the search of a stage. However, an obvious drawback of dynamic programming is that it is difficult to handle alignments with backward bends due to its basic assumption.

As with the network optimization approach, the optimal alignment derived by dynamic programming is not precise unless the number of stages and the number of states at each stage are sufficiently large. However, these are undesirable elements, especially when the distance between successive stages is very small, because it becomes almost impossible to fit smooth curves even though we control the exit direction (angle) at each state. Therefore, dynamic programming models usually utilize a coarse grid and then possibly a refined grid in a second search.

2.5 Models for optimizing vertical alignment

Many more models are found for optimizing vertical alignment than for horizontal alignment. The main reason is that only a few costs (for example earthwork cost) are significantly influenced by the vertical alignment so that other insensitive cost items can be ignored during optimization. Moreover, in optimizing vertical alignment, there is no need to worry about backward bends because the vertical alignment is designed along the horizontal alignment. In the literature, the existing models fall into four categories. These are discussed below.

2.5.1 Enumeration

Easa (1988) presents a model which selects road grades that minimize earthwork cost while satisfying the geometric specifications. His model is initialized by setting the stations along the horizontal alignment at equal distance. Then the elevation increments and ranges of intersection points at each station are specified. The search procedure is quite straightforward. At each iteration, a combination of intersection points is selected to check whether the resulting alignment is feasible. Note that the alignment is derived in a conventional design manner, i.e., by fitting a parabolic curve for each successive linear segment connecting the intersection points. The constraints to be checked include minimum slope, maximum gradient, minimum distance between reverse curves, range of elevation at each station, and so forth. If any one of the constraints is violated, then the set of intersection points is discarded and another set of points is selected to repeat the procedure. If the resulting alignment is feasible, then the program calculates the earthwork volumes for each cut or fill section. It then further checks whether the amount of borrow or disposal volume exceeds the capacity of the borrow pit or landfill. If the net borrow volume exceeds the borrow pit capacity or the net disposal volume exceeds the landfill capacity, then the set of intersection points is infeasible, and another set is selected. Otherwise, a linear programming model is run to optimize the amount of earth moved between different sections, borrow pits and land fills. In other words, the linear programming model is used to derive the most economic earth-moving plan. The above procedure is repeated until all combinations of intersection points have been investigated. The one that has the least earth-moving cost provides the optimal alignment.

Easa’s model considers most of the important constraints and the calculation of earthwork is based on the realistic alignment rather than piecewise linear segments used in most dynamic programming models. Moreover, the resulting solution guarantees a globally minimal earthwork cost. However, the approach is inefficient because enumeration is a very exhaustive search approach. Furthermore, the intersection point at each station is only allowed to lie on a discrete set of points, which is only a subset of the problem’s search space. If a more detailed search space is desired, then the elevation increments of intersection points must be sufficiently small. However, the resulting problem is quite large and the enumeration method cannot handle it. Another weakness is that only earthwork cost is considered in Easa's model. Other costs such as vehicle operating cost and pavement cost are ignored.

2.5.2 Dynamic programming

Dynamic programming is the most widely used method for optimizing vertical alignment because the problem fits quite easily into the structure of this approach. Each station of the alignment is considered as a stage in a dynamic programming model and the states at each stage are the set of points with different elevations at each station.

Puy Huarte (1973) presented a dynamic programming model incorporating linear programming and transportation problem models to optimize vertical alignment. The program presented by Puy Huarte is called OPYGAR (Optimisation and Automatic Generation of Longitudinal Profiles) and was developed in Spain. The program consists of two phases. In phase one, dynamic programming is used to generate the optimal profile, given the hauling cost. The gradient constraint is processed by limiting a set of feasible nodes corresponding to a given node in the current stage. Based on the optimal solution found by dynamic programming, a new solution to the hauling cost is obtained by applying the transportation problem model, which is solved with the Hungarian Algorithm (Winston, 1994). Then, given the new hauling cost solution, dynamic programming is used again to find an optimal profile. This procedure is repeated until convergence is reached. In phase two, the final alignment found in phase one is approximated by a set of cubic polynomial functions. A linear programming model is then run to minimize the absolute value of curvature at each intermediate point.

In Puy Huarte's model, the final alignment is not consistent with that during the dynamic programming search. The alignment in the dynamic programming model is represented by piecewise linear segments, which may be quite different from the eventual alignment. Note that Puy Huarte's model is somewhat similar to Easa's (1988) approach, where the mass-haul plan is derived by linear programming and the alignment is obtained by enumerating intersection points. The alignment in the OPYGAR program is first derived by dynamic programming and then smoothed by a linear programming model, while the hauling cost is optimized by solving a transportation problem (which can also be solved by linear programming).

It must be mentioned that the OPYGAR program considers possible retaining walls and bridges if the difference between the ground profile and the proposed road profile exceeds certain values. This feature improves the applicability and flexibility of the OPYGAR program.

Murchland (1973) also employed a dynamic programming approach (with interpolation) to optimize vertical profiles by minimizing earthwork cost for the improvement of existing roads. That program is called VALOR. Unlike other dynamic programming models in which the alignment is represented by piecewise linear trajectories, Murchland used a set of quadratic spline functions with equal intervals to specify the alignment. The advantage of this approach is that the first and second derivatives of the alignment can be obtained at any point along the alignment so that the gradient and gradient change constraints can be easily formulated. The author chose quadratic spline functions because the alignment is smooth everywhere (i.e., the first derivative is continuous) and because a quadratic spline function is the one in the spline families satisfying this property with the lowest degree. In addition to gradient, curvature, and elevation restriction constraints, the model also considers a minimum slope constraint (for drainage purpose) and undulation restrictions (to prevent the alignment from changing the gradient too often). Inevitably, the resulting model (a non-convex constrained optimization problem with many local minima) is very complex and difficult to solve. Half a dozen methods were employed to solve the problem, including linear programming, mixed integer programming, a smoothing method, random search, direct search, and dynamic programming. Among them, dynamic programming with interpolation can solve the problem with assurance of a good solution. The decisions at each stage of dynamic programming include not only the elevation but also the gradient of the road profile. The gradient decision is required because the first derivatives of the quadratic spline functions between two consecutive intervals must be continuous at the intermediate stations. Then through the quadratic condition, the set of spline functions can be solved in a sequence.

The advantage of Murchland’s model is that the continuous functions for the alignment representation provide a smooth alignment with the first and second derivatives. However, the alignment is still restricted to pass through a limited finite set of points at each station. Moreover, the large computational requirements are another weak point of this model.

Goh, Chew, and Fwa (1988) presented a discrete and a continuous model for optimizing vertical alignment. The discrete model employs dynamic programming to solve the problem, whereas the continuous model uses numerical search method (will be discussed in the next subsection). The costs considered in their model include earthwork and vehicle operating costs, and the alignment is represented by piecewise linear segments. In addition, the authors provided detailed formulations for calculating earthwork cost and an explicit formula (which is generally obtainable from any textbook) to explain the dynamic programming procedure.

Fwa (1989) presented the same dynamic programming model for optimizing vertical alignment, where the same numerical example was used again with extra work on sensitivity analysis. The main finding was that the best choice of horizontal interval is 60 m in length and the best vertical grid increment is 0.5 m. The sensitivity analysis also shows that the relaxation of maximum gradient and gradient change constraints significantly reduce the total cost. However, the total cost is insensitive to the relaxation of gradient change after it exceeds a certain value. The reason is that when the gradient change is slackened up to a certain value, it becomes non-binding and thus the gradient constraint dominates the behavior of the alignment.

In general, the application of dynamic programming for optimizing vertical alignment is quite successful when compared with optimizing horizontal alignment. However, it restricts the alignment (or intersection points) at each station to a finite set of points. Thus, only a subset of the problem's search space is examined, and it would be impossible to extend it to cover the whole search space. Furthermore, most dynamic programming models tend to generate the alignment in piecewise linear segments, which may be too rough for applications. Even though a refined alignment can be obtained by fitting parabolic curves to the optimal solution found by dynamic programming, that is inconsistent with the alignment during search. This is an inherent limitation of dynamic programming because before the optimal solution is found, it is impossible to fit the curves due to the assumption of independent sub-problems in dynamic programming. From this viewpoint, Murchland's (1973) model is more useful because it determines the behavior of the alignment within each sub problem. However, as mentioned above, the decision at each stage becomes more complex since the gradient at each point must be determined. Of course, it has infinite values within the feasible range of gradient. That is why Murchland's model needs interpolation to cope with dynamic programming.

2.5.3 Linear programming

ReVelle, Whitlatch, and Wright (1997) employed a linear programming model to optimize vertical alignment so that the earthwork cost is minimized. Quite unlike other models, where the decision variables are usually the elevations of the alignment at each station, this utilized a 5th order polynomial function to depict the alignment without needing prior knowledge about the elevations at each station. A linear programming model is then formulated to optimize the coefficient of the polynomial function so that the total earthwork cost, including cut and fill, is minimized while subject to gradient and gradient change constraints. Since the vertical profile is a fifth order continuous function, the first and second derivatives can be obtained without difficulty and thus the constraints can be readily formulated.

One of the advantages of this approach is that the alignment is a single smooth function, where the elevation at any points along the alignment can be derived by simply plugging the horizontal distance measured from the start point. The other benefit is that there exist some well-developed algorithms such as the simplex method to solve the problem. However, there is no strong evidence to show that the alignment can be specified by a 5th order polynomial function. As discussed in the next chapter, this is not a real alignment. Moreover, as indicated in Chapra (1988), a higher order polynomial function tends to abruptly bend the curve, which is undesirable in a highway alignment. Secondly, the earthwork volume in the proposed model is calculated in a simplified way without considering side slopes. If the side slopes, pavement or vehicle operating costs are considered, the resulting objective function becomes nonlinear and is not solvable by linear programming. Finally, the gradient and gradient change constraints are formulated at the points of each station. However, there is no evidence that other points along the alignment will also satisfy the constraints.

2.5.4 Numerical search

Models using numerical search for optimizing vertical alignment are proposed to overcome some obvious shortcomings in other approaches, such as enumeration and dynamic programming. The search space defined in this approach is continuous rather than a discrete solution set. Therefore, it provides more flexibility for the alignment configuration. Of course, it will be more difficult to solve because the resulting problems usually turn out to be nonlinear, or even non-convex.

Hayman (1970) suggested a model, where the decision variables are defined as the elevations at each station and are continuous in nature. The alignment is then generated by connecting these points with straight line segments. In his model, the gradient and curvature constraints are formulated in a way similar to Goh *et al* (1988) and Fwa (1989). However, Hayman considered more constraints such as slope stability and material balance constraints. In calculating earthwork volume, Hayman’s formulation allows the existence of a tilting ground profile (still a straight line) at the road cross-section, whereas Goh and Fwa assumed that the ground profile at the road cross-section is flat.

The search method employed in Hayman’s study can be characterized as a line search method. First, an initial guess of the solution is defined. Then a new point is formed by moving the original point toward its gradient direction with a step size which will be optimized. The procedure is repeated until no non-zero step size is found. The computational sequence is then altered to solve an auxiliary problem that seeks a new feasible direction. (This is called the method of feasible directions.) The entire algorithm will finally end up with a solution better than the other nearly points in the search space. Obviously the solution found by this approach cannot guarantee a global optimum. In practice, several different initial solutions are used to increase the possibility of finding a relatively satisfactory solution.

The program MINERVA, developed at the Transportation and Road Research Laboratory (TRRL), also applies a numerical search method to optimize vertical alignment. Discussions of the MINERVA program can be found in OECD (1973) and Pearman *et al* (1973). Unlike several other models, where the alignment is represented by linear approximation, or spline, or polynomial functions, the MINERVA program is designed to mimic an engineer’s design process and result in a solution that can be directly put into practice. In other words, the alignment derived from the program consists of straight-line segments with parabolic curves fitted between them. Since MINERVA endeavors to get close enough to reality, the resulting objective function and constraints are very complex and the problem is then difficult to solve.

The decision variables defined in the MINERVA program are the coordinates of each intersection point on the vertical surface along the horizontal alignment, and the length of the vertical curve for each point of intersection. The optimization technique is a univariate direct search method. It works from the initial feasible solution. Then it alters one decision variable at one time with others fixed until no further improvements in the objective function can be achieved. Clearly the final solution is only a local optimum and the algorithm is not efficient enough. Another characteristic of this approach is that the resulting solution may follow a certain pattern, depending on the orders of variables changed. To overcome the problem of multiple local optima, the program is usually run several times with different initial solutions. The one that yields the lowest objective function value is selected as the final solution.

For a nonlinear optimization model like the MINERVA or Hayman's model, instead of arbitrarily trying different initial solutions, it is better to understand the behavior of the objective function. Pearman (1973) suggested that random sampling be used to obtain a set of feasible solutions, and then use statistical analysis to estimate an adequate frequency distribution of the objective function. This may give engineers a clue about the goodness or performance of the current solution obtained by the program.

Robinson (1973) presented a program called VENUS, which can automatically generate a good initial solution that may be used for further search. The algorithm is initialized by obtaining a quasi-vertical alignment which is derived by smoothing ground profile. Then the quasi-vertical profile is converted into the preliminary vertical alignment, where the tangent lines are formed by fitting straight lines to the quasi-vertical alignment at inflection points by a least squares process. Next, the alignment is further adjusted so that all constraints are satisfied. The resulting alignment is called a feasible vertical alignment and can be used as the initial solution in any search algorithm.

In the paper by Goh, Chew, and Fwa (1988), a continuous model for optimizing vertical alignment was also presented. The model is first formulated as a calculus of variations problem. Since it is very difficult to solve the necessary conditions (i.e., the Euler- Lagrangian equations), the original model is further converted into an optimal control problem by some mathematical techniques from optimal control theory (Teo and Goh, 1987; Goh and Teo, 1988). The alignment in their model is parameterized by a set of cubic spline functions. Thus the gradient and curvature constraints can be easily formulated due to the availability of the first and second derivatives of cubic spline functions. It is noted that these two constraints must be satisfied at an infinite number of points along the alignment. It is impossible to list the constraints for all points. To facilitate computation, they are transformed into one-dimensional constraints via a constraint transcription used in optimal control theory. The final model then becomes a general constrained nonlinear optimization problem with the coefficients of spline functions as its decision variables. The model can be solved by a numerical search method and has several local minima.

The authors used this continuous model and its counterpart, a discrete dynamic programming model (mentioned in previous sub section), to solve the same problem. The results show that the dynamic programming model is superior in terms of computation efforts and ease of formulation. However, the continuous model has better flexibility and potential for upgrading to solve 3-dimensional problems.

Generally, for optimizing highway alignment, a well-formulated continuous model provides more flexibility to the alignment configurations, and has the potential to yield a realistic alignment. However, its drawbacks are the difficulties in both formulation and solution algorithms. Furthermore, the resulting problems are usually nonlinear and non-convex. Thus, many local optima exist in the search space, which inconveniences both developers and engineers.

2.6 Models for simultaneously optimizing horizontal and vertical alignments

Although much progress has been made in developing models for optimizing vertical alignment, the development of models that simultaneously optimize horizontal and vertical alignments is not yet successful because there are more factors involved and more complexities in the geometric specification of a 3-dimensional alignment. There are two basic approaches found in the literature: dynamic programming and numerical search. Although some of network optimization models such as those developed by Parker (1977) and Trietsch (1987a) claim the capability of simultaneously selecting both horizontal and vertical alignments, they employ a 2-stage optimization approach (i.e., vertical alignment is determined prior to horizontal alignment), and are essentially 2-dimensional optimizers. These models have already been discussed in section 2.1.

2.6.1 Dynamic programming

The basic structure of dynamic programming models for optimizing 3-dimensional alignment is to set the stages as equally spaced planes between the start and end points, which are, from a top view, perpendicular to the line segment connecting two end points of the alignment. At each stage, the search grids (i.e., states) are located on a 2-dimensional plan. Linking straight-line segments from the origin to the destination through the grids at each stage will generate a 3-dimensional alignment.

Hogan (1973) presented a dynamic programming model, OPTLOC, which is used by U.S. Forest Service for optimizing road alignment and profile. Generally a coarse search grid is used initially. Then successive iterations, accomplished by refinement of the search grid, can be made to choose a route at any desired tolerance level.

Nicholson (1976) also employed similar approach to optimize route location. At the first stage, the model searches a relative coarse grid of points for a preliminary alignment (or corridor). Then a discrete variational calculus is adopted to refine the alignment so that the resulting alignment can deviate from the grid points.

The application of dynamic programming in optimizing 3-dimensional alignment has several defects. First, it has difficulty in handling backtracking alignments. Secondly, the resulting solution is very rough for both horizontal and vertical alignments, and there are difficulties in dealing with both horizontal and vertical curvatures. Finally, the storage requirements may hinder this approach from searching a finer grid initially.

2.6.2 Numerical search

The only model found in the literature that simultaneously optimizes a "smooth" 3-dimensional alignment was developed by Chew, Goh, and Fwa (1989). This model is the extension of their continuous model for optimizing vertical alignment (Goh, Chew, and Fwa, 1988).

The problem is initially formulated as a calculus of variations problem. As mentioned in the last section, the necessary conditions are very difficult to solve. The authors thus utilized a set of cubic spline functions to interpolate the alignment, then transformed the constraints into one-dimensional constraints by the method of constraint transcription used in the optimal control theory. Finally the model becomes a constrained nonlinear program structure with the coefficient vectors of spline functions as its decision variables.

It is noted that the objective function involves integrals, which are not easy to compute. Thus a numerical integration is suggested by the authors to facilitate the computation during search. The solution algorithm employed in their paper is the quasi Newton descent algorithm. Variable scaling is also considered to improve the convergence performance.

Like other models for optimizing vertical alignment by numerical search, the solution found in Chew's model only guarantees a local optimum. In practice, different initial solutions with human judgments are used for running the model. Furthermore, the variable scaling procedure requires interactions between the program and users during optimization. Therefore, the model is not fully automatic. Another potential defect embedded in this model is that it seems difficult to incorporate discontinuous location dependent costs (e.g., land acquisition cost) into the objective function because the algorithm requires a differentiable objective function.

Regardless of the above disadvantages and difficulties of formulation, this model possesses some good features. At least it successfully optimizes a smooth 3-dimensional alignment, which is not possible with the dynamic programming approach or with any other known models besides those presented in the following chapters of this book.

2.7 Summary

In the previous three sections, several existing models for optimizing highway alignments are investigated. The potential advantages and disadvantages in each approach are summarized in Table 2.5 to Table 2.7 in a somewhat general way. For example, the numerical search approach for optimizing vertical profile by Hayman’s model (1970) only produces a set of linear piecewise segments rather than a smooth alignment. However, this can be ameliorated if Goh, Chew, and Faw’s (1988) model is employed. Thus, the advantages and disadvantages listed for each approach are general to each method rather than a specific model. Furthermore, the note “possibly finds the global optimum” appearing in the tables means that a particular approach can find the global optimum under its assumptions, which are, however, unrealistic for optimizing highway alignments.

Apparently, none of the approaches shown in Table 2.5 to Table 2.7 dominates the others, and there is always some trade off between them. The problem turns out to be: what approach (or model) is most promising for optimizing highway alignments. To answer this question, the features of a good model may be outlined. These are readily merged from Table 2.5 to Table 2.7.

Necessary conditions of a good model for optimizing highway alignments

(1) Consider all dominating and sensitive costs
(2) Formulate all important constraints
(3) Yield a realistic alignment
(4) Be able to handle alignment with backward bends
(5) Simultaneously optimize 3-dimensional alignments
(6) Find globally or near globally optimal solutions
(7) Have an efficient solution algorithm
(8) Have low storage requirements
(9) Have a continuous search space
(10) Automatically avoid inaccessible regions
(11) Be compatible with GIS (Geographical Information System) databases

The first condition usually depends on the availability of data, but a good model should be able to deal with various cost items, including continuous and discrete cost functions. The second to the fifth conditions will be considered in more detailed and mathematical forms in the next chapter. The seventh and eighth conditions are less critical than in previous times. With technology developments, the speed and memory of modern computers has significantly increased. The last condition, “be compatible with GIS databases”, is the trend in transportation planning and analysis because most spatial data will be stored in GIS databases.

One may find that the aforementioned conditions are somewhat conflicting. Emphasizing some aspects may neglect the others. It is very difficult and challenging to trade off these conditions. The history of research in this field shows that recent development has become very slow, or even stagnant. In fact, most of the models were developed in late 60’s and early 70’s, when many highways were built all over the world. After that only a few papers were published. The most promising one (Chew *et al*, 1989) was developed almost two decades later. Why? This seems due not only to the shortage of new highway projects, but also to the lack of theoretical breakthroughs. This book attempts to develop models that yield better tradeoffs among the above conditions. However, we are not going to solve the entire problem since each of these conditions deserves a major research effort.

Table 2.5: Potential advantages and disadvantages of existing approaches for optimizing horizontal alignment.

Method	**Advantages**	**Disadvantages**
Calculus of variations	- Yields smooth alignment - Possibly finds the global optimum - Has continuous search space	- Cannot deal with discontinuous cost items (requires well-behaved objective function) - Complex modeling and computation efforts
Network optimization	- Simple and easy to use - Well-developed algorithms for solving the problem exist - Possibly finds the global optimum	- The resulting alignment is not smooth - Discrete solution set rather than continuous search space - Large memory requirements
Dynamic programming	- Simple and easy to use - Well-developed algorithms for solving the problem exist - Possibly finds the global optimum	- The resulting alignment is not smooth - Discrete solution set rather than continuous search space - Difficulty in handling backward bends

Table 2.6: Potential advantages and disadvantages of existing approaches for optimizing vertical alignment.

Method	Advantages	Disadvantages
Enumeration	- Can yield a realistic alignment - Possibly finds the global optimum - Can consider most of the important constraints	- Inefficient - Discrete solution set rather than continuous search space
Dynamic programming	- Simple and easy to use - Well-developed algorithms for solving the problem exist - Possibly finds the global optimum	- The resulting alignment is not smooth - Discrete solution set rather than continuous search space
Linear programming	- Simple and easy to use - Well-developed algorithms for solving the problem exist - Possibly finds the global optimum - Can yield smooth alignment - Has continuous search space	- Only limited cost items and constraints can be formulated (must be linear) - Gradient and curvature constraints are formulated for a limited number of points
Numerical search	- Can yield a realistic alignment - Can consider most of the important constraints and various costs - Has continuous search space	- Multiple local optima exist - Complex modeling and computation efforts

Table 2.7: Potential advantages and disadvantages of existing approaches for simultaneously optimizing horizontal and vertical alignments.

Method	Advantages	Disadvantages
Dynamic programming	- Simple and easy to use - Well-developed algorithms for solving the problem exist - Possibly finds the global optimum	- The resulting alignment is not smooth - Discrete solution set rather than continuous search space - Large memory requirements - Difficulty in handling backward bends
Numerical search	- Can yield a realistic alignment - Can consider most of the important constraints and various types of costs - Has continuous search space	- Multiple local optima exist - Complex modeling and computation efforts - Difficulty in modeling discontinuous cost items

Chapter 3

Basic definitions and properties of alignment optimization

The basic definitions and properties of alignment optimization are investigated before the models are developed to provide a general insight to the problem. All definitions and properties are described in mathematical forms, which facilitate the development of the models. Unlike traditional highway design in which all elements are explained in separate mathematical forms, this chapter uses a generic parametric function to represent alignments. The mathematics used here to develop definitions, theorems, and proofs are general tools from Analytical Geometry, Differential Geometry, Calculus, Vector Analysis, and Linear Algebra. For notational convenience, throughout this book we shall use boldface letters to denote vectors in space. Sometimes, where it is difficult to represent a vector by a single boldface letter, a notation such as $\overrightarrow{AB}$ (equivalent to $\mathbf{B}-\mathbf{A}$) may be employed to denote the vector from point A to point B .

3.1 Basic definitions and theorems of highway alignments

An alignment is a smooth curve (or path) in space. To depict a space curve, usually a parametric representation is employed. There are many reasons for using parametric curves. In particular, not all space curves can be expressed in single-valued explicit functions such as $y = f(x)$. (The other reasons are discussed in more detail in Mortenson, 1997.) An example for parametric form of road alignments can be found in Lovell (1999) and Lovell *et al* (2001).

The alignment in a generic road design problem is a space curve bounded by its end points. Let $\mathbf{S} = [x_S, y_S, z_S]'$ and $\mathbf{E} = [x_E, y_E, z_E]'$ denote the position vectors of the start and end points of the alignment, respectively. Further assume that the parameter of the alignment, denoted by u , has been normalized so that $u \in [0,1]$. Then the alignment can be defined as follows:

Definition 3.1 *Let* L *be a set of points defined by the position vector* $\mathbf{P}(u) = [x(u), y(u), z(u)]'$, *where* $u \in [0,1]$. *If* L *is an alignment connecting* $S(x_S, y_S, z_S)$ *and* $E(x_E, y_E, z_E)$, *then the position vector* $\mathbf{P}(u)$ *must satisfy*

$$(1)\ \mathbf{P}(0) = \mathbf{S} \text{ and } \mathbf{P}(1) = \mathbf{E}, \tag{3.1}$$

$$(2)\ \mathbf{P}(u) \text{ is continuous in the interval } u \in [0,1], \tag{3.2}$$

$$(3)\ \mathbf{P}'(u) \text{ is continuous in the interval } u \in [0,1], \tag{3.3}$$

where $\mathbf{P}'(u) = \lim_{\Delta u \to 0} \dfrac{\mathbf{P}(u+\Delta u) - \mathbf{P}(u)}{\Delta u} = [x'(u), y'(u), z'(u)]'$ *is a tangent vector to* L *at point* $P(u)$.

The third condition in the above definition restricts the alignment to be “smooth”. This is necessary because a highway must be built in such a way that vehicles can smoothly move along it safely and comfortably.

Definition 3.1 basically provides necessary but not sufficient conditions of an alignment. More specifically, the projection of the alignment L onto the XY plane, denoted by L_{xy} , is composed of straight-line sections and circular curves (see Wright 1996 for example). Formally, L_{xy} is called the horizontal alignment of L. For modern highway design, a smooth transition curve between a straight-line section and a circular curve is required unless a large radius curve or low design speed is adopted (Underwood, 1991). Transition curves are used to accommodate the changes in radius and superelevation, and provide a safer and more comfortable driving environment. Clothoid spirals are the most commonly used transition curves in highway design. (The mathematical forms and detailed calculations for the clothoid spiral are provided by Walton and Meek, 1989). Figure 3.1 shows an example of a 3-dimensional alignment projected onto the XY plane. The basic geometric specifications of the resulting horizontal alignment (which are simply derived by ignoring the Z component in the position vector $\mathbf{P}(u)$) is shown in Figure 3.2.

Another geometric characteristic of an alignment is its projection onto the surface orthogonal to the XY plane along the horizontal alignment. Assume the surface is stretched so that it is flat (see Figure 3.3 for an example). Let us name it the HZ plane, where H is the distance measured along the horizontal alignment L_{xy} . Then the projection of the alignment L onto the HZ plane, denoted by L_{hz} , is composed of straight-line sections and parabolic curves (see Wright 1996 for example). Formally L_{hz} is called the vertical alignment or vertical profile of L. The parabolic curve is employed to provide a gradual change in road grade for safety and comfort reasons. Figure 3.4 illustrates the geometric specification of a typical vertical alignment.

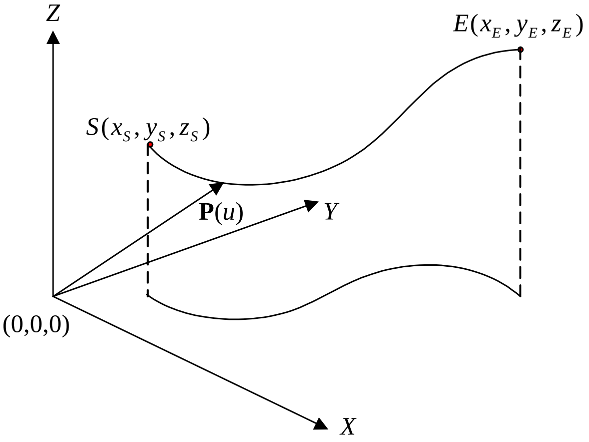


Figure 3.1: The projection of a highway alignment on the XY plane.

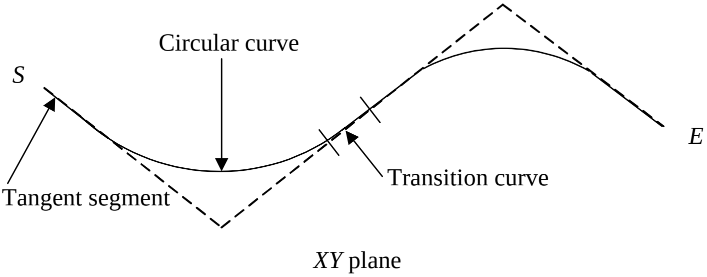


Figure 3.2: A typical horizontal alignment on the *XY* plane.

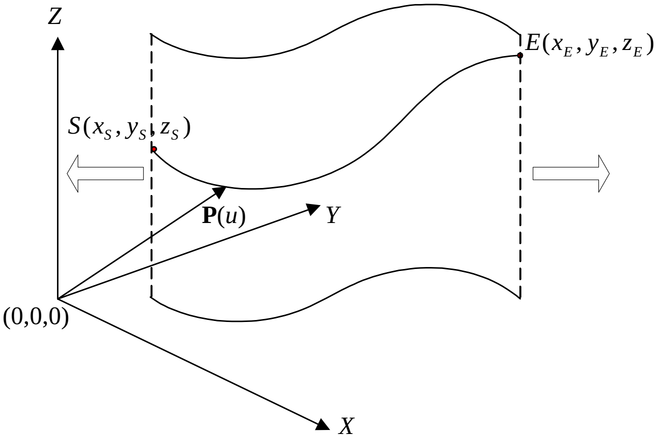


Figure 3.3: Projecting a highway alignment onto the surface orthogonal to the XY plane and stretching the surface to make it flat.

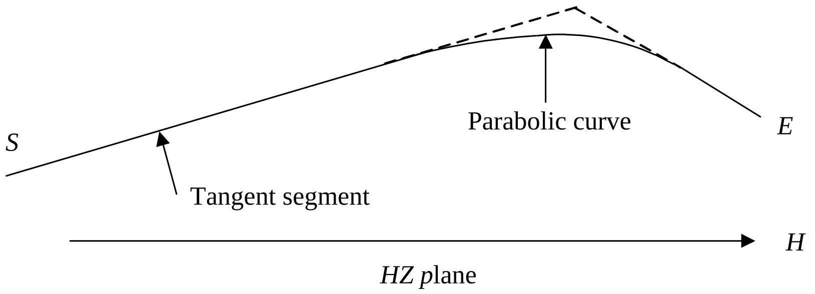


Figure 3.4: A typical vertical alignment on the *HZ* Plane.

We further define the alignment as follows:

Definition 3.2 *If L is a 3-dimensional alignment, then*

(1) L_{xy} contains no elements other than tangent sections, circular curves, and spirals.
(2) L_{hz} contains no elements other than tangent sections and parabolic curves.

Mathematically we can rewrite the above statements as

$$(1')\ L_{xy} \cap \overline{C_t \cup C_c \cup C_s} = \phi\,, \tag{3.4}$$

$$(2')\ L_{hz} \cap \overline{C_t \cup C_p} = \phi\,, \tag{3.5}$$

where L_{xy} and L_{hz} are the corresponding horizontal and vertical alignments of L, C_t, C_c, C_s, and C_p denote the set of tangent sections, circular curves, spiral curves, and parabolic curves in $\Re^2$ (2-dimensional space).

Next, we will examine a very important property of alignment configuration, which is essential in optimizing horizontal alignment and 3-dimensional alignment.

Definition 3.3 *Let L be the alignment defined by the position vector $\mathbf{P}(u) = [x(u), y(u), z(u)]'$, $u \in [0,1]$, where $\mathbf{S} = \mathbf{P}(0)$ and $\mathbf{E} = \mathbf{P}(1)$ are the position vectors of start and end points of L. Assume L_{xy} is the corresponding horizontal alignment of L. Then L and L_{xy} are said to be backtracking if there exists some $u \in [0,1]$ such that the position vector $\mathbf{P}(u)$ satisfies the following property on the XY plane.*

$$\cos^{-1}\left(\frac{\mathbf{P}'(u)\cdot(\mathbf{E}-\mathbf{S})}{\|\mathbf{P}'(u)\|\|\mathbf{E}-\mathbf{S}\|}\right) \geq 90^\circ \text{ , or equivalently } \mathbf{P}'(u)\cdot(\mathbf{E}-\mathbf{S}) \leq 0\,, \tag{3.6}$$

where $\|\cdot\|$ *denotes the norm of the vector. Otherwise,* L *and* L_{xy} *are called non-backtracking.*

Literally the above definition states that if at some points along the alignment, the angle between $\overrightarrow{SE}$ and the tangent vector to $\mathbf{P}(u)$, denoted by $\mathbf{P}'(u)$, is greater than or equal to $90°$, then the alignment is said to be backtracking. Otherwise it is non-backtracking. Figure 3.5 gives a graphic interpretation of Definition 3.3, where the alignment is backtracking.

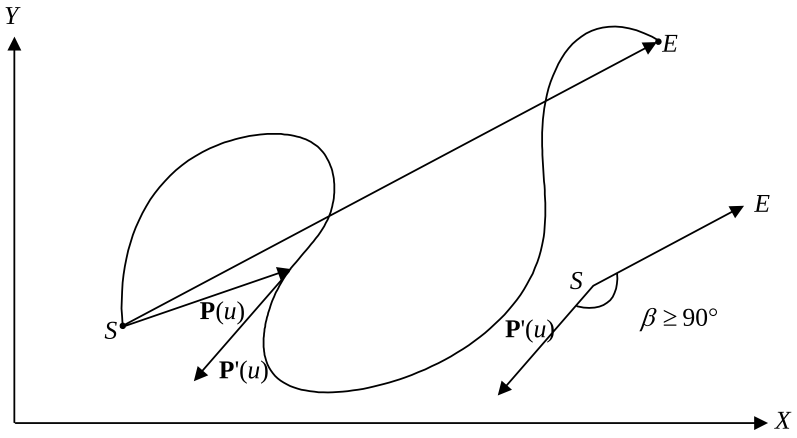


Figure 3.5: An example of backtracking alignment.

Intuitively a backtracking alignment is undesirable because it is indirect and makes the road longer. However, in some cases it might be necessary for reducing the cost of constructing a highway. Especially in mountainous areas, the alignment may bend along the mountain to avoid heavy excavation or backfill.

The backtracking property makes alignment optimization more difficult. To explain why it complicates the analysis, we must first inspect the following theorem.

Theorem 3.1 *Let* L *be an alignment from* $S(x_S, y_S, z_S)$ *to* $E(x_E, y_E, z_E)$ *and* $N = \{(x, y, z) | y = a + bx\}$ *define a plane orthogonal to the XY plane. If* $N \cap \overline{SE} \neq \phi$, *then* $N \cap L \neq \phi$, *where* $\overline{SE}$ *is the line segment between the points* S *and* E.

Proof: Assume that we transform the XYZ coordinate system (rotated along the Z axis) into $X^T Y^T Z$ coordinate system so that the new axis X^T is perpendicular to N on the XY (or equivalently, $X^T Y^T$) plane, as shown in Figure 3.6. Since the Z coordinate remains unaltered and N is orthogonal to the X^TY^T plane, we can neglect the Z coordinate in the proof.

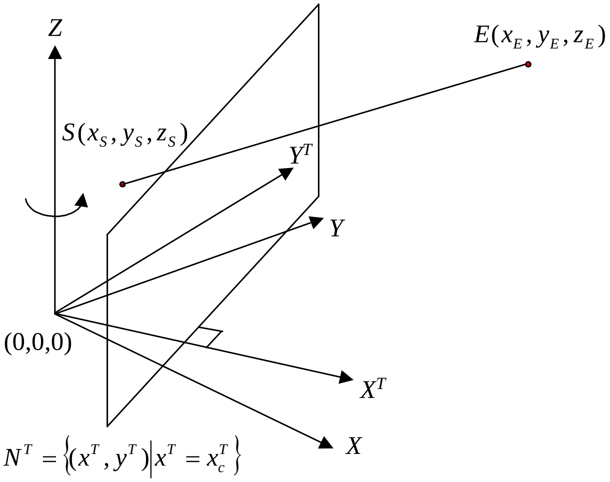


Figure 3.6: The coordinate transformation from XYZ to X^TY^TZ .

We now look at the horizontal alignment $L_{x^Ty^T}$, which is defined by the position vector $\mathbf{P}^T(u) = [x^T(u), y^T(u)]', u \in [0,1]$ on the X^TY^T plane, where $(x^T(0), y^T(0)) =$ (x_S^T, y_S^T) and $(x^T(1), y^T(1)) = (x_E^T, y_E^T)$ are the corresponding start and end points of the alignment. Assume $x_S^T \le x_E^T$ (If $x_E^T \le x_S^T$, we can still complete the proof in a similar way.). Then by Definition 3.1, the position vector $\mathbf{P}^T(u)$ is continuous over the interval $u \in [0,1]$, or equivalently

$$\lim_{u \to c} \mathbf{P}^T(u) = \mathbf{P}^T(c), \quad \forall c \in [0,1]. \tag{3.7}$$

Eqn (3.7) also implies that $\mathbf{P}^T(u)$ is continuous over all $x_S^T \le x^T \le x_E^T$.

We next examine the behavior of N on the X^TY^T plane. Since the coordinate system has been rotated in such a way that the X^T axis is perpendicular to N , the resulting line of N on the X^TY^T plane, denoted by N^T , can be rewritten as

$$N^T = \{(x^T, y^T) | x^T = x_c^T\}. \tag{3.8}$$

Since $N \cap \overline{SE} \ne \phi$, it implies that $x_S^T \le x_c^T \le x_E^T$ and there exists some $u_c \in [0,1]$ such that $x^T(u_c) = x_c^T$ (known as “Intermediate Value Theorem” (Swokowski, 1980)). From eqn (3.7), we also get

$$\lim_{u \to u_c} \mathbf{P}^T(u) = \mathbf{P}^T(u_c) = [x_c^T, y^T(u_c)]' \qquad (3.9)$$

and thus $N \cap L = \{x_c^T, y^T(u_c)\} \neq \phi$.

The theorem states that if the plane N intersects the line segment $\overline{SE}$ ($N \cap \overline{SE} \neq \phi$), then N also intersects the alignment L ($N \cap L \neq \phi$). But the reverse is not always true. Figure 3.7 is a counter example, which shows that the plane N intersects the alignment L without intersecting the line segment $\overline{SE}$.

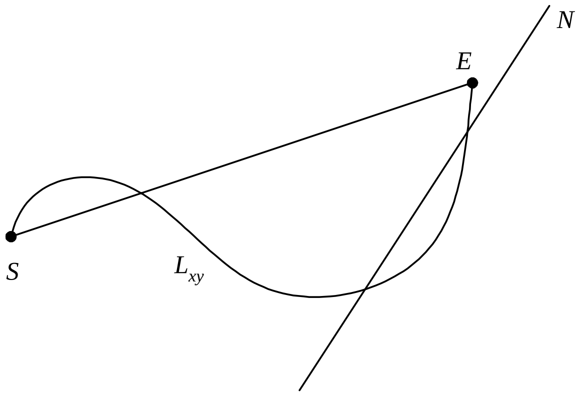


Figure 3.7: An example in which N intersect L_{xy} without intersecting $\overline{SE}$.

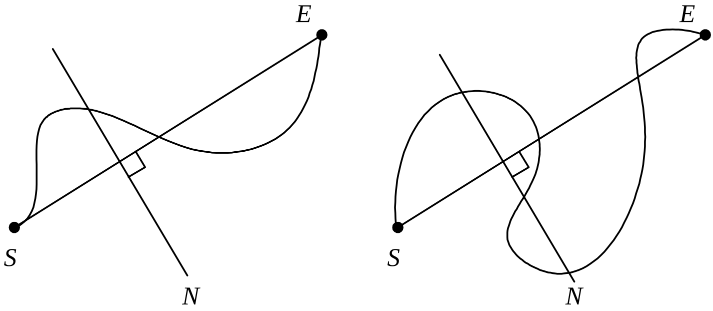
Combining Definition 3.1, Definition 3.3 and Theorem 3.1, we observe the following corollary:

Corollary 3.1 *Let* L *be an alignment from* $S(x_S, y_S, z_S)$ *to* $E(x_E, y_E, z_E)$ *and* $N = \{(x, y, z) | y = a + bx\}$ *define a plane orthogonal to the* XY *plane. Assume* N *is perpendicular to the projection of* $\overline{SE}$ *on the* XY *plane and* $N \cap \overline{SE} \neq \phi$ *, where* $\overline{SE}$ *is the line segment between the points* S *and* E *. Then the fact that* L *is non-backtracking implies that* N *intersects* L *at only one point in* $\Re^3$ *(3-dimensional space).*

Proof: By Theorem 3.1, if $N \cap \overline{SE} \neq \phi$, then $N \cap L \neq \phi$. Assume that N intersects L at n points occurring at $u_1, \ldots\ldots u_n$, where $u_i \in [0,1]$, $\forall i = 1, \ldots. n$. Since L is non-backtracking, from Definition 3.3 we know that

$$\cos^{-1}\left(\frac{\mathbf{P}'(u_i) \cdot (\mathbf{E} - \mathbf{S})}{\|\mathbf{P}'(u_i)\| \|\mathbf{E} - \mathbf{S}\|}\right) < 90°, \ \forall i = 1, \ldots., n \ .$$

Also note that N is perpendicular to $\overline{SE}$. Therefore, n must be equal to 1. Otherwise $\mathbf{P}(u)$ is discontinuous, which violates the second condition in Definition 3.1. The above corollary can also be illustrated by Figure 3.8.



(a) Non-backtracking case (b) Backtracking case

Figure 3.8: Backtracking and non-backtracking alignments.

Corollary 3.1 is very useful in developing models for optimizing alignments. Assume we know that the actual alignment is non-backtracking. Then each vertical plane between S and E will intersect the alignment at exactly one point. All dynamic programming models found in the literature for optimizing highway alignments are based on this concept. Unfortunately, as shown in Figure 3.6, if the actual optimal alignment is backtracking, there may exist some vertical planes which intersect the alignment at more than one point. This may prevent dynamic programming models from locating the optimal alignment.

3.2 The properties of models for optimizing alignments

In this section, the alignment optimization problem will be described in a mathematical form. Although the problem will not be solved in this format, it helps us to realize the nature and difficulties of the problem, and eventually to develop a solvable model. The notations used here basically follow those in the previous section with additional variables, which are summarized in Table 3.1.

As discussed in chapter 2, highway costs can be categorized into location–dependent, length-dependent, area-dependent, volume-dependent, and VMT-dependent costs. Assume that the forecast traffic demands throughout the lifetime of the proposed highway are given, and the environmental impacts per VMT are available. Then the VMT-dependent cost can be incorporated into length-dependent cost. (The detailed transformation will be presented in the next chapter.) The user costs of a highway alternative depends on various features of the alignment and thus can not be classified into a single category. For example, travel time cost depends on the length of the alignment and travel speed, which is further influenced by curvature, gradient, and other design features. Therefore, user cost will stand alone as a cost item.

Table 3.1: Notations for general alignment optimization problem.

Variables	**Descriptions**
$A_C(u)$	Highway cross-section area of cut at u
$A_F(u)$	Highway cross-section area of fill at u
C_A	Area-dependent cost
C_L	Length-dependent cost
C_N	Location-dependent cost
C_T	Total cost
C_U	User cost
C_V	Volume-dependent cost
$G_{\max}$	Maximal allowable gradient
$H_{\max}$	Maximal allowable curvature for horizontal alignment
I	The set of inaccessible regions
K_A	Unit area-dependent cost
K_C	Unit cutting cost
K_F	Unit filling cost
K_L	Unit length-dependent cost
K_N	Unit location-dependent cost
$V_{\max}$	Maximal allowable curvature for vertical alignment
$W(u)$	Highway width at u

Now alignment optimization problems can be written in the following mathematical format. Note that some formulas are common in general calculus textbooks (for example Swokowski, 1980), and thus are not further derived here.

Model 0 – Basic model for optimizing highway alignment

(1) Objective function

$$\underset{\mathbf{P}(u)}{\text{Minimize}}\ C_T = C_N + C_L + C_A + C_V + C_U \tag{3.10}$$

subject to:

(2) Boundary conditions

$$\mathbf{P}(0) = \mathbf{S} \tag{3.11}$$

$$\mathbf{P}(1) = \mathbf{E} \tag{3.12}$$

(3) Alignment necessary conditions

$$L_{xy} \cap \overline{C_t \cup C_c \cup C_s} = \phi \tag{3.13}$$

$$L_{hz} \cap \overline{C_t \cup C_p} = \phi \tag{3.14}$$

(4) Horizontal curvature constraint

$$\frac{\left| \frac{dx(u)}{du} \frac{d^2 y(u)}{du^2} - \frac{d^2 x(u)}{du^2} \frac{dy(u)}{du} \right|}{\left[\left(\frac{dx(u)}{du} \right)^2 + \left(\frac{dy(u)}{du} \right)^2 \right]^{\frac{3}{2}}} \le H_{\max}, \forall u \in [0,1] \tag{3.15}$$

(5) Gradient constraint

$$\left| \frac{dz(u)}{dh} \right| \le G_{\max}, \ \forall u \in [0,1] \tag{3.16}$$

(6) Vertical curvature constraint

$$\frac{\left| \frac{dh}{du} \frac{d^2 z(u)}{du^2} - \frac{d^2 h}{du^2} \frac{dz(u)}{du} \right|}{\left[\left(\frac{dh}{du} \right)^2 + \left(\frac{dz(u)}{du} \right)^2 \right]^{\frac{3}{2}}} \le V_{\max}, \ \forall u \in [0,1] \tag{3.17}$$

(7) Inaccessible region

$$\left\{ x(u), y(u) \middle| u \in [0,1] \right\} \cap I = \phi \tag{3.18}$$

(8) System equations

$$C_N = \int_0^1 K_N \left(x(u), y(u) \right) W(u) \sqrt{x'(u)^2 + y'(u)^2} \, du \tag{3.19}$$

$$C_L = K_L \int_0^1 \sqrt{x'(u)^2 + y'(u)^2 + z'(u)^2} \, du \tag{3.20}$$

$$C_A = K_A \int_0^1 W(u) \sqrt{x'(u)^2 + y'(u)^2 + z'(u)^2} \, du \tag{3.21}$$

$$C_V = K_C \int_0^1 A_C(u) \sqrt{x'(u)^2 + y'(u)^2} \, du + K_F \int_0^1 A_F(u) \sqrt{x'(u)^2 + y'(u)^2} \, du \tag{3.22}$$

$$C_U = f\left(\int_0^1 \sqrt{x'(u)^2 + y'(u)^2 + z'(u)^2}\,du, \frac{\left| \frac{dx(u)}{du}\frac{d^2y(u)}{du^2} - \frac{d^2x(u)}{du^2}\frac{dy(u)}{du} \right|}{\left[\left(\frac{dx(u)}{du}\right)^2 + \left(\frac{dy(u)}{du}\right)^2 \right]^{\frac{3}{2}}}, \right. \tag{3.23}$$

$$\left. \frac{dz(u)}{dh}, \frac{\left| \frac{dh}{du}\frac{d^2z(u)}{du^2} - \frac{d^2h}{du^2}\frac{dz(u)}{du} \right|}{\left[\left(\frac{dh}{du}\right)^2 + \left(\frac{dz(u)}{du}\right)^2 \right]^{\frac{3}{2}}} \right)$$

$$dh = \sqrt{dx(u)^2 + dy(u)^2} \tag{3.24}$$

The above formulations define the problem more rigorously. The objective is to find the position vector $\mathbf{P}(u)$ (a vector-valued function) of the alignment, which minimizes the sum of various cost items, while subject to a set of constraints. Note that it is very difficult to solve the problem in this format because some equations do not have closed forms. However, it sheds light on the nature and difficulties of the problem, and allows us to justify existing models. This will eventually help the development of a valid model.

Eqns (3.11) to (3.14) are already introduced in the previous section and will not be repeated here. The horizontal curvature in eqn (3.15) implies that the first and second derivatives of $y(u)$ and $x(u)$ with respect to u exist. The first derivatives $dx(u)/du$ and $dy(u)/du$ are continuous because the alignment is continuously differentiable over the interval $u \in [0,1]$ as stated in Definition 3.1. However, the second derivatives $d^2x(u)/du^2$ and $d^2y(u)/du^2$ are not necessarily continuous everywhere in the interval $u \in [0,1]$ for roads in rural and mountainous areas, where transition curves between different design elements may be omitted. In particular, on tangent sections, the second derivatives of the position vector become zeros and the left hand side of eqn (3.15) is undefined. Therefore, the horizontal curvature constraint can be checked only at the points where the second derivative is non-zero (i.e., at curve sections). A similar argument also applies to the vertical curvature constraint.

The vertical curvature constraint given in eqn (3.17) has a similar form to eqn (3.15). Since vertical alignments are non-backtracking by their nature and consist of tangents and parabolic curves, as defined in eqn (3.14), the Z component (i.e., $z(u)$) of the position vector $\mathbf{P}(u)$ can be expressed as a function of h . It is also possible to reconvert h as a linear function of u and therefore d^2h/du^2 in the second term of the numerator of eqn (3.17) becomes 0. Accordingly, eqn (3.17) can be rewritten as

$$\left| \frac{d^2 z(u)/dh^2}{\sqrt{\left[1+\left(dz(u)/dh\right)^2\right]^3}} \right| \le V_{\max}, \ \forall u \in [0,1] \tag{3.25}$$

In some existing models found in the literature (for example Fwa, 1989; Hayman, 1970; ReVelle *et al*, 1997; Murchland, 1973), the vertical curvature constraint is formulated as the change in gradient, which is approximately true. However, none of the above papers mentioned this. In fact eqn (3.25) shows that the vertical curvature is equivalent to the change in gradient only when the gradient reaches zero. Since the highway gradient is generally small, the denominator in eqn (3.25) may be approximated as 1.

The curvature (both horizontal and vertical) and gradient constraints are very difficult to formulate because there are infinitely many points along the alignment. It is very inefficient to formulate eqns (3.15) to (3.17) for every point. Accordingly, some existing models only set the constraints at a limited number of spots or stations (for example Parker, 1977 and ReVelle *et al*, 1997). The only model found in the literature which takes this into account is that by Chew *et al* (1989) and Gho *et al* (1988). In their model, some mathematical techniques from optimal control theory were employed to convert the constraints into one-dimensional constraints. However, the fact that the second derivatives in eqns (3.15) and (3.17) are not necessarily continuous everywhere along the alignment was not considered in the model. In fact, we do not know where the second derivative is continuous until we find the alignment. Without considering this fact, the authors approximated the alignment by a set of cubic spline functions, which imply that the alignment is twice continuously differentiable. As we know, this is not true because the necessary conditions in eqns (3.13) and (3.14) are violated, although cubic spline functions provide very good approximations for most curves.

Regardless of user cost, the computations of other cost items are straightforward. They are the combinations of the multiplication of length, width, and cross-section area. Note that it is very difficult to calculate the location-dependent cost in eqn (3.19). The unit cost K_N is a function of the X and Y coordinates of the alignment (i.e., the cost depends on the location of the alignment). As discussed in chapter 2, location-dependent cost is discontinuous between different zones and thus it is impossible to use a single function of u to represent K_N . In fact, none of the models found in the bibliography explicitly take this factor into account except that of Athanassoulis and Calogero (1973), where a network optimization model is employed to optimize horizontal alignment. However, a network model (a transshipment problem in their case) yields a non-smooth alignment, which violates the third condition in Definition 3.1 (eqn (3.3)), and the search space in their model is discrete rather than continuous. Therefore, this approach is undesirable.

The calculation of user cost is also complex. In fact, it is very difficult to derive a closed form of the user cost. Eqn (3.23) only gives a general relation between user cost and design features. For computation, however, it is not sufficient. Unlike other costs, which can be expressed as a functional integration, user cost can not be calculated until the alignment is completely determined. This can explain why most existing models did not incorporate user cost into the objective function. The difficulty is especially embedded in the OCP approach, the models by Chew *et al* (1989), and by Goh *et al* (1988), and all network optimization and dynamic programming models.

The nature and difficulties of the problem have been investigated. But what is the best approach to the problem? To answer this question, we must first investigate the following two sub-questions:

(1) How shall we represent the alignment?
(2) What data format provides the best description of the problem?

In the literature, the representations of the alignment have five formats: piecewise linear segments (linear spline), quadratic spline, cubic spline, polynomial function, and real alignment as in the MINERVA program (Pearman 1973). As we know, the first four approaches do not completely satisfy eqns (3.14) to (3.17). The last format is the best. If a model can mimic the design process (i.e., starting from tangent segments and then fitting curves at each intersection point), then the boundary conditions and alignment necessary conditions are satisfied. The curvatures and gradient constraints will also be satisfied elsewhere along the alignment as long as the model can carefully fit the circular and parabolic curves into the alignment. The main remaining problem is to develop a good algorithm for optimization since we know that the resulting objective function is not well behaved.

As to the second question, it is very difficult, or even impossible, to store all information for all points in the study region (continuous space). Therefore, some simplified formats are used to store the spatial data for further calculating the costs associated with an alignment alternative. Existing approaches include storing cost data at nodes, or links (network optimization and dynamic programming), fitting cost surface (calculus of variations) and direct calculation (enumeration, linear programming, and numerical search approach). Among them, direct calculation seems to have the lowest storage requirements for a given precision level. Therefore, it will be the one employed in this book.

This page intentionally left blank

Part B

Intelligent road alignment models

This page intentionally left blank

Chapter 4

Modeling the optimization problem for non-backtracking horizontal alignments

In optimizing a highway alignment, if the study region is relatively flat and the cost of building the highway at different locations within the region is not too irregular, we may expect that the optimal alignment is non-backtracking. In this chapter, a model for optimizing non-backtracking horizontal alignment will be formulated. According to the necessary conditions of a good optimization model for highway alignment (discussed in section 2.7), the model developed in this chapter is not perfect. However, it is a starting point, from which a more sophisticated model can be evolved by relaxing assumptions and expanding capabilities. More specifically, the model developed in this chapter does not satisfy conditions (4) and (5). However, other conditions will be pursued as far as possible.

This chapter starts with the data format for describing the region of interest. The modeling approach for representing alignments is presented in section 4.2. The cost function in the proposed model is briefly outlined in section 4.3 and discussed in more detail in sections 4.4 to 4.6. The properties of the model are finally discussed in section 4.7.

4.1 Data format for describing the region of interest

As indicated in the end of the previous chapter, it is impossible to store all information at all points in the region of interest. That is not only because there are infinitely many points in the region, but also due to the lack of available information at all points. Therefore, a data format is required to minimize the needed memory and carry important information for the entire region. A matrix format is employed in this book. The basic assumptions are:

(1) The study region is in a rectangular shape. (Other shapes can be modified as rectangles as will be discussed later.)

(2) The study region can be partitioned into several equal-dimensional cells within which the land acquisition cost, land-use characteristics and soil conditions are practically uniform, and the internal variation of elevations is relatively small compared to that between different cells.

Figure 4.1 provides an example of the format for describing the study area, in which the coordinates of the origin (bottom left corner) are labeled as $O(x_O, y_O)$ and the dimensions of each cell are D_x and D_y. We further denote $x_{\max}$ and $y_{\max}$ as the maximal X and Y coordinates of the study region. It is desirable that the coordinate system of the study area be consistent with the real coordinate system for easy location of the route origin and destination on the map. Another advantage of adopting a real coordinate system is that the optimal solutions obtained in the model can be directly related to the real coordinates.

The proposed format has several advantages. First, it can comply with the real coordinate system as mentioned above. Next, it is convenient for programming because a matrix is a commonly used data structure in computers. Finally, any GIS data can be easily converted to this format. The last advantage is also a necessary condition (11) of a good model, as discussed at the end of chapter 2.

It should be noted that the optimal alignment could pass through any points in the region of interest rather than be limited to a finite set of points as in graph optimization and dynamic programming models. In the next section, we will discuss how the alignment can be represented.

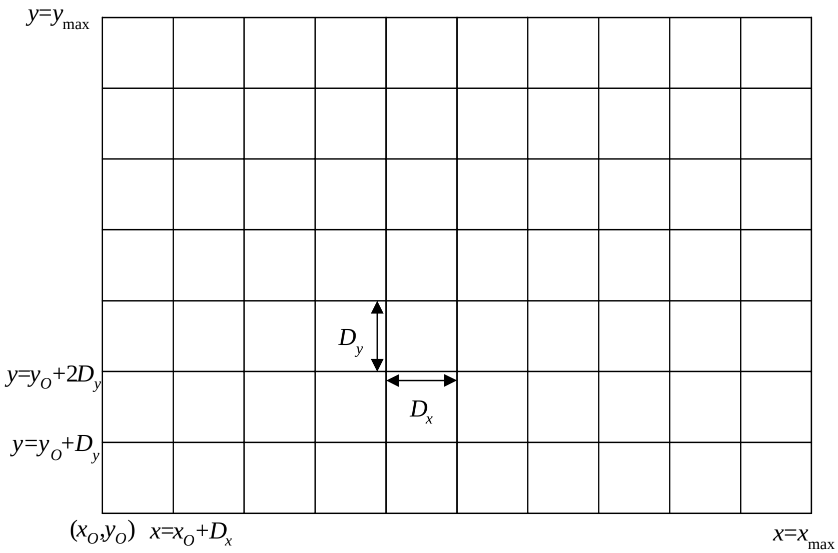


Figure 4.1: An example of study area for alignment optimization.

4.2 Representation of alignment

The concept on which the alignment representation is based is motivated by Corollary 3.1. Let $S(x_S, y_S)$ and $E(x_E, y_E)$ be the start and end points of the alignment and $N = \{(x, y) | y = a + bx\}$ denote a line in the XY plane. Suppose that the optimal alignment exists somewhere else in the study region. From Theorem 3.1 we know that if N intersects the line segment $\overline{SE}$, then N also intersects the optimal alignment. Since we assume that the optimal alignment is non-backtracking, from Corollary 3.1, if N is perpendicular to $\overline{SE}$ (we refer N as a vertical cutting line (in 2 dimensional space) or vertical cutting plane (in 3 dimensional space)), then N will intersect the optimal alignment at exactly one point. Suppose we place several vertical cutting lines to the line segment $\overline{SE}$. Then the intersecting points on those vertical cuts along with S and E will yield a rough appearance of the alignment. The more vertical cuts we apply to $\overline{SE}$, the more detailed is the alignment we obtain.

The idea of defining the decision variables to depict the horizontal alignment is based on this "cutting" concept. Suppose that we cut the line segment $\overline{SE}$ n times at equal distance between contiguous cuts as shown in Figure 4.2. Then, the intersecting points of the optimal alignment at each vertical cut are those points which we want to find out during the optimization. Instead of directly searching the XY coordinates of the intersecting points, we simply define the decision variables as the coordinates along the vertical cutting lines in order to minimize the number of variables in the optimization model. Note that in the proposed approach, any plane curves connecting two continuous vertical cuts will have the same orthogonal projection onto the line segment $\overline{SE}$. Colloquially, it is said that all non-backtracking alignments make "continual progress" from S to E.

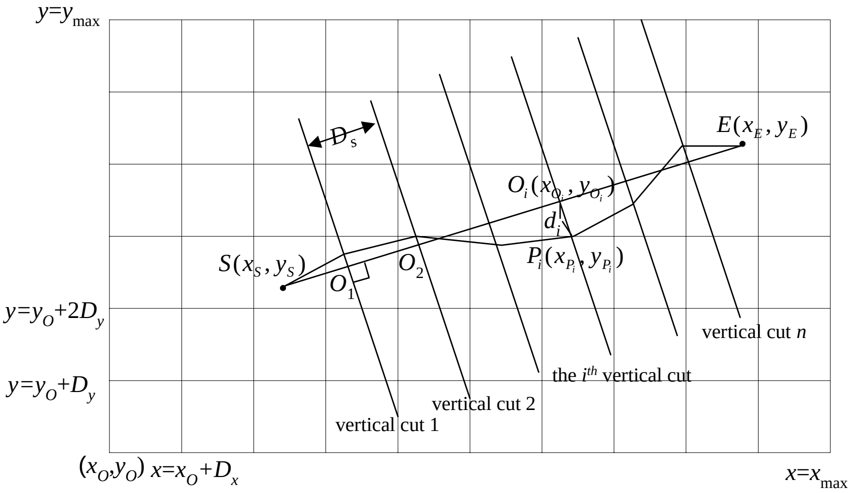


Figure 4.2: Decision variables at each vertical cut.

The coordinate system at each vertical cut is not well defined yet because the direction is not specified. For maximum generality, we define the upward direction as the positive direction and the downward direction as the negative direction. The only exception occurs when the vertical cuts are orthogonal to the X axis, in which case the positive direction is defined rightward and the negative direction is leftward.

For each vertical cut, the origin is defined at the intersection point of the cut and the line segment $\overline{SE}$. Let $\mathbf{u}_{xy} = [x_u, y_u]'$ denote the unit vector from S to E on the XY plane. Then

$$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (x_E - x_S)/[(x_E - x_S)^2 + (y_E - y_S)^2]^{1/2} \\ (y_E - y_S)/[(x_E - x_S)^2 + (y_E - y_S)^2]^{1/2} \end{bmatrix}. \tag{4.1}$$

Using the unit vector one can easily calculate the $X\,Y$ coordinates of the origin at each vertical cut. Let O_i be the origin at the i^{th} vertical cut, $\forall i = 1,......,n$. Then the coordinates of O_i, denoted as (x_{O_i}, y_{O_i}), are derived as

$$\begin{bmatrix} x_{O_i} \\ y_{O_i} \end{bmatrix} = \begin{bmatrix} x_S \\ y_S \end{bmatrix} + iD_s \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} x_S \\ y_S \end{bmatrix} + \frac{i}{n+1} \begin{bmatrix} x_E - x_S \\ y_E - y_S \end{bmatrix}, \tag{4.2}$$

where $D_s = \left[(x_E - x_S)^2 + (y_E - y_S)^2\right]^{1/2} /(n+1)$ is the distance between contiguous vertical cutting lines.

Let d_i be the coordinate of the intersection point at the i^{th} vertical cut. Then we must identify the upper and lower bounds of $d_i, \forall i = 1,......,n$ because they act as constraints in the optimization model. Four different cases arise in determining these bounds, depending on the angle of the cutting line. Let θ denote the angle between the cutting line and the X axis. Then θ ranges from $0°$ to $180°$, and can be calculated by

$$\theta = \tan^{-1}\left(\frac{y_E - y_S}{x_E - x_S}\right) + 90°. \tag{4.3}$$

In most computer software, $\tan^{-1}$ returns a value ranging from –90° to 90° and therefore the range of θ is $0°$ and $180°$. We now let d_{iU} and d_{iL} be the upper and lower bounds of d_i respectively. Then d_{iU} and d_{iL} are determined according to the following four cases:

(1) Case 1: $\theta = 0°$ or $\theta = 180°$

In this case, the positive direction of the coordinate axis is rightward. Thus the upper bound of d_i is reached at $x_{\max}$, while the lower bound occurs at x_O (see

Figure 4.3). Since the study site is rectangular, the upper and lower bounds for all vertical cuts are the same, i.e.,

$$\begin{aligned} d_{iU} &= x_{\max} - x_{O_i} \\ d_{iL} &= x_O - x_{O_i} \end{aligned} . \tag{4.4}$$

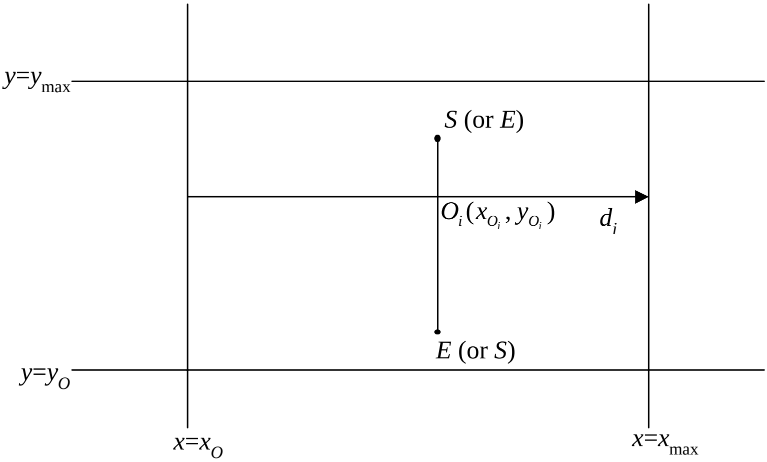


Figure 4.3: d_{iU} and d_{iL} when $\theta = 0°$ or $\theta = 180°$.

(2) Case 2: $0° < \theta < 90°$

Figure 4.4 shows that if $0° < \theta < 90°$, then the upper bound of d_i is reached when the coordinate axis reaches either $x_{\max}$ or $y_{\max}$, whichever is reached first. Likewise, the lower bound of d_i is reached at either x_O or y_O:

$$\begin{aligned} d_{iU} &= \min\left\{\frac{x_{\max} - x_{O_i}}{\cos\theta}, \frac{y_{\max} - y_{O_i}}{\sin\theta}\right\} \\ d_{iL} &= \max\left\{\frac{x_O - x_{O_i}}{\cos\theta}, \frac{y_O - y_{O_i}}{\sin\theta}\right\} \end{aligned} . \tag{4.5}$$

(3) Case 3: $\theta = 90°$

If $\theta = 90°$, then the upper bound of d_i is reached when the coordinate axis reaches $y_{\max}$, whereas the lower bound is reached when it touches y_O (see Figure 4.5). As in case 1, all d_i have the same bounds:

$$\begin{aligned} d_{iU} &= y_{\max} - y_{O_i} \\ d_{iL} &= y_O - y_{O_i} \end{aligned} . \tag{4.6}$$

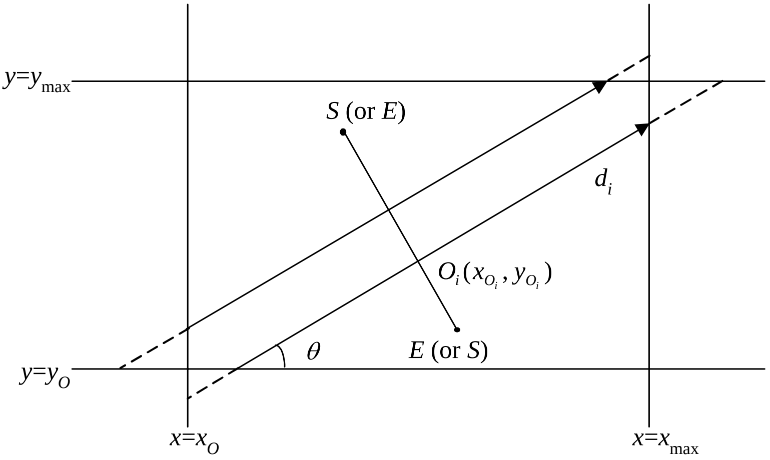


Figure 4.4: d_{iU} and d_{iL} when $0° < \theta < 90°$.

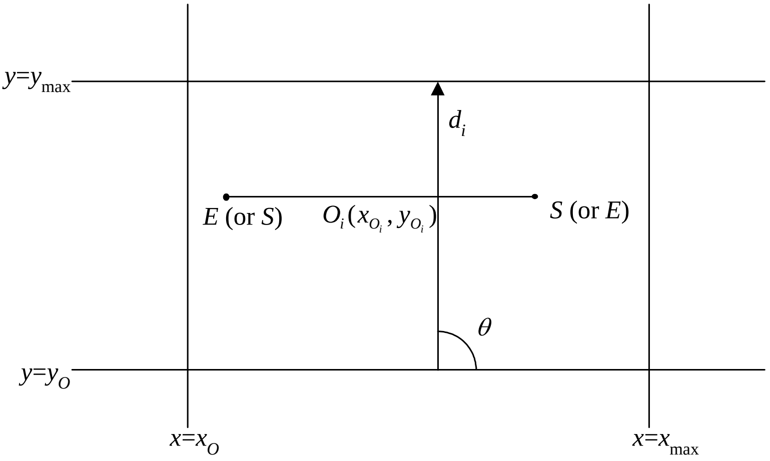


Figure 4.5: d_{iU} and d_{iL} when $\theta = 90°$.

(4) Case 4: $90° < \theta < 180°$

As shown in Figure 4.6, if $90° < \theta < 180°$, then the upper bound of d_i is reached when the coordinate axis reaches either x_O or $y_{\max}$, whichever is reached first. Similarly the lower bound of d_i is reached at either $x_{\max}$ or y_O:

$$\begin{aligned} d_{iU} &= \min\left\{ \frac{x_O - x_{O_i}}{\cos\theta}, \frac{y_{\max} - y_{O_i}}{\sin\theta} \right\} \\ d_{iL} &= \max\left\{ \frac{x_{\max} - x_{O_i}}{\cos\theta}, \frac{y_O - y_{O_i}}{\sin\theta} \right\} \end{aligned} . \tag{4.7}$$

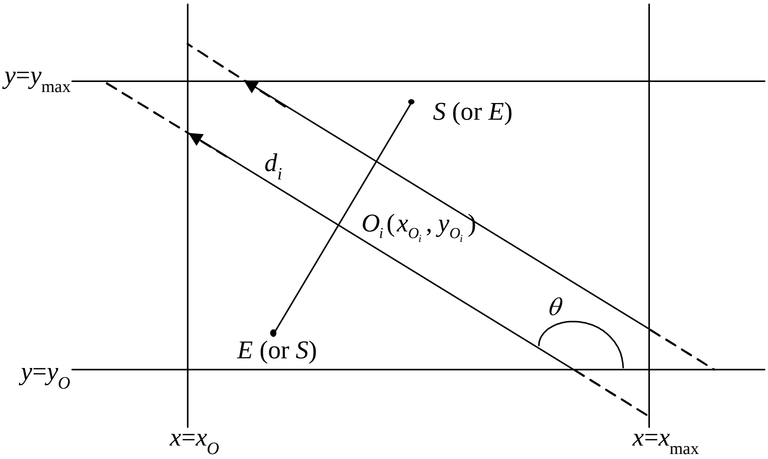


Figure 4.6: d_{iU} and d_{iL} when $90° < \theta < 180°$.

The decision variables for delineating the alignment and their associated boundaries have been defined. A given set of d_i 's represents a set of points on different coordinate axes. For consistency, they must be converted into the *X Y* coordinate system. Let P_i be the intersection point located at the i^{th} vertical cut, whose position is determined by d_i . Then the *X Y* coordinates of P_i , denoted by (x_{P_i}, y_{P_i}) , can be obtained as follows:

$$\begin{bmatrix} x_{P_i} \\ y_{P_i} \end{bmatrix} = \begin{bmatrix} x_{O_i} \\ y_{O_i} \end{bmatrix} + d_i \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} . \tag{4.8}$$

The set of points P_i , $i = 1,....,n$ generally outlines the track of the alignment. Linking each pair of successive points by a straight-line section will generate a piecewise linear trajectory. Next, circular curves must be fitted into the tangent sections at the intersection points, unless the intersection angle is 0. Without violating the design standard, we assume that the minimum radius for a given design speed is used to fit the horizontal curves between the tangent sections, understanding that the minimum radius may result in a higher superelevation and less comfort than a longer curve. According to AASHTO (2001), the minimum radius, denoted by $R_{\min}$, for a given design speed V_d can be computed as

$$R_{\min} = \frac{V_d^2}{15(e + f_s)}, \tag{4.9}$$

where e = superelevation
f_s = coefficient of side friction.

Note that minimum radius constraints are not the most restrictive constraints in designing alignments. The continuity condition (second condition in Definition 3.1) is much stronger because if a discontinuity occurs, then the resulting plane curve is no longer an alignment, whereas if the continuity condition holds but the minimum radius constraint is violated, then the resulting plane curve is still an alignment (although a poor one). For this reason, we must develop an algorithm to fit the circular curves. Before the algorithm is outlined, we need the following mathematical foundations.

For notational convenience, let P_0 and P_{n+1} denote S and E respectively. (i.e., $(x_{P_0}, y_{P_0}) = (x_S, y_S)$ and $(x_{P_{n+1}}, y_{P_{n+1}}) = (x_E, y_E)$.) We further define the following variables (the geometric meaning of each variable is shown in Figure 4.7).

Table 4.1: Variable definitions for a circular curve.

Variables	**Descriptions**
$C_i(x_{C_i}, y_{C_i})$	The point of curvature (beginning of curve) pertaining to $P_i,\ i = 1,....,n$
$T_i(x_{T_i}, y_{T_i})$	The point of tangency (end of curve) pertaining to $P_i,\ i = 1,....,n$
$\delta_i(x_{\delta_i}, y_{\delta_i})$	The center of the circular curve pertaining to $P_i,\ i = 1,....,n$
R_i	The radius of the circular curve pertaining to $P_i,\ i = 1,....n$
Δ_i	Intersection angle at $P_i,\ i = 1,....,n$
$L_T(i)$	Tangent length from C_i to P_i (P_i to T_i)

By vector analysis and trigonometry, we obtain

$$\begin{aligned}\Delta_i &= \cos^{-1}\left(\frac{(\mathbf{P}_i - \mathbf{P}_{i-1})\cdot(\mathbf{P}_{i+1} - \mathbf{P}_i)}{\|\mathbf{P}_i - \mathbf{P}_{i-1}\|\|\mathbf{P}_{i+1} - \mathbf{P}_i\|}\right)\\ &= \cos^{-1}\left(\frac{(x_{P_i} - x_{P_{i-1}})(x_{P_{i+1}} - x_{P_i}) + (y_{P_i} - y_{P_{i-1}})(y_{P_{i+1}} - y_{P_i})}{\sqrt{(x_{P_i} - x_{P_{i-1}})^2 + (y_{P_i} - y_{P_{i-1}})^2}\sqrt{(x_{P_{i+1}} - x_{P_i})^2 + (y_{P_{i+1}} - y_{P_i})^2}}\right),\end{aligned} \tag{4.10}$$

$$L_T(i) = R_i \times \tan\frac{\Delta_i}{2}. \tag{4.11}$$

To determine the radius at each intersection point, we must calculate the intersection angle Δ_i first. Then we use $R_{\min}$ to compute the tangent length. If the length between any two consecutive intersection points (say P_i and P_{i+1}) is less than the sum of tangent length ($L_T(i) + L_T(i+1)$), then a discontinuity occurs, as in the example shown in Figure 4.8. Therefore, the radii for these two intersection points must be reduced so that the continuity condition can hold. For consistency, assume that the radii at these two intersection points are reduced to the same value derived with the following formula:

$$R_i = R_{i+1} = \left\|\mathbf{P}_{i+1} - \mathbf{P}_i\right\| \Big/ \left(\tan(\Delta_i / 2) + \tan(\Delta_{i+1} / 2)\right). \tag{4.12}$$

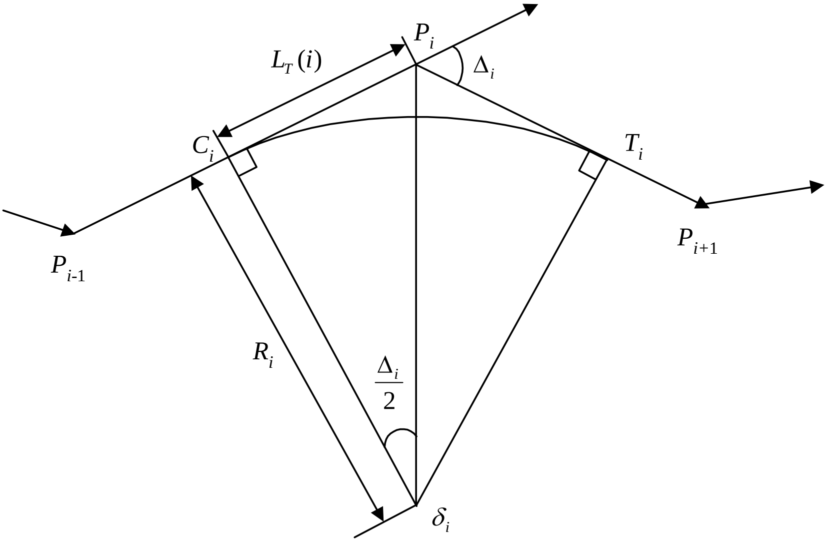


Figure 4.7: The geometric specification of a circular curve.

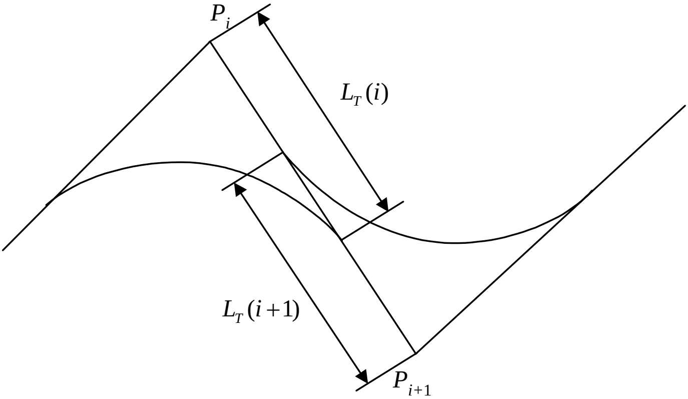


Figure 4.8: An example of discontinuity in horizontal curve.

Note that if discontinuities exist in a series of successive tangent sections, then it is best to first reduce the radii at one or both end points of the most critical tangent section. Based on the revised radii, the next critical tangent is identified and the radii at its two (or one) end points are then updated to their maximal possible value (at most $R_{\min}$) while satisfying the continuity condition. This procedure is continued until all tangents satisfy the continuity condition. We now summarize the radius determination algorithm as follows:

Algorithm 4.1 Radius determination procedure

(0) Step 0: Variable definitions

Variables	Descriptions
D_i	The deficiency in the length of the i^{th} tangent, $i = 0,....,n$
R_c	The radii at the end points of the most critical tangent
R_t	Temporary radius
I_c	The index to indicate the most critical tangent
f_i	Boolean variable to indicate whether the radius at P_i must be fixed ($f_i = 1$) or not ($f_i = 0$)

```
(1) Step 1: Initialization
    Set $R_i = R_{\min}, \forall i = 1,....,n$
    Calculate intersection angle $\Delta_i, \forall i = 1,...,n$ with eqn (4.10)
    Calculate tangent length $L_T(i), \forall i = 1,...,n$ with eqn (4.11)
    Set $L_T(0) = L_T(n+1) = 0$
    Set $D_i = (L_T(i) + L_T(i+1)) - \|\mathbf{P}_i - \mathbf{P}_{i+1}\|, \forall i = 1,...,n$
    Set $f_0 = f_{n+1} = 1$
    Set $f_i = 0, \forall i = 1,....,n$
    Set $I_c = 0$
(2) Step 2: Identify discontinuous tangent sections
    Set $R_c = R_{\min}$
    Set $i = 0$
    2-1 If $i \le n$, then continue 2-2; Otherwise go to step 3
    2-2 If $D_i > 0$, then continue; Otherwise go to step 2-4
        2-2-1 If $f_i = f_{i+1} = 0$, then
              Set $R_t = \|\mathbf{P}_i - \mathbf{P}_{i+1}\| \big/ (\tan(\Delta_i/2) + \tan(\Delta_{i+1}/2))$
              Go to step 2-3
        2-2-2 If $f_i = 0$ and $f_{i+1} = 1$, then
              Set $R_t = (\|\mathbf{P}_i - \mathbf{P}_{i+1}\| - \tan(\Delta_{i+1}/2)) \big/ \tan(\Delta_i/2)$
              Go to step 2-3
        2-2-3 If $f_i = 1$ and $f_{i+1} = 0$, then
              Set $R_t = (\|\mathbf{P}_i - \mathbf{P}_{i+1}\| - \tan(\Delta_i/2)) \big/ \tan(\Delta_{i+1}/2)$
              Go to step 2-3
    2-3 If $R_t < R_c$, then
        Set $R_c = R_t$
        Set $I_c = i$
    2-4 Set $i = i + 1$; go to step 2-1
```

```
(3) Step 3: Termination rule
    If $R_c = R_{\min}$, then stop; Otherwise continue step 4.
(4) Step 4: Update radii and other variables
    4-1 If $f_i = f_{i+1} = 0$, then
        Set $R_{I_c} = R_c$
        Set $R_{I_c+1} = R_c$
        Set $L_T(I_c) = R_{I_c} \tan(\Delta_{I_c}/2)$
        Set $L_T(I_c+1) = R_{I_c+1} \tan(\Delta_{I_c+1}/2)$
        Set $D_{I_c} = 0$
        If $I_c \neq 0$, set $D_{I_c-1} = \left(L_T(I_c-1) + L_T(I_c)\right) - \left\|\mathbf{P}_{I_c-1} - \mathbf{P}_{I_c}\right\|$
        If $I_c \neq n$, set $D_{I_c+1} = \left(L_T(I_c+1) + L_T(I_c+2)\right) - \left\|\mathbf{P}_{I_c+1} - \mathbf{P}_{I_c+2}\right\|$
        Set $f_{I_c} = f_{I_c+1} = 1$
        Go to step 2
    4-2 If $f_i = 0$ and $f_{i+1} = 1$, then
        Set $R_{I_c} = R_c$
        Set $L_T(I_c) = R_{I_c} \tan(\Delta_{I_c}/2)$
        Set $D_{I_c} = 0$
        If $I_c \neq 0$, set $D_{I_c-1} = \left(L_T(I_c-1) + L_T(I_c)\right) - \left\|\mathbf{P}_{I_c-1} - \mathbf{P}_{I_c}\right\|$
        Set $f_{I_c} = 1$
        Go to step 2
    4-3 If $f_i = 1$ and $f_{i+1} = 0$, then
        Set $R_{I_c+1} = R_c$
        Set $L_T(I_c+1) = R_{I_c+1} \tan(\Delta_{I_c+1}/2)$
        Set $D_{I_c} = 0$
        If $I_c \neq n$, set $D_{I_c+1} = \left(L_T(I_c+1) + L_T(I_c+2)\right) - \left\|\mathbf{P}_{I_c+1} - \mathbf{P}_{I_c+2}\right\|$
        Set $f_{I_c+1} = 1$
        Go to step 2
```

The above algorithm is similar to a manual engineering design process. With it, we can generate a unique alignment for a given set of intersection points. The resulting alignment is composed of tangent sections and circular curves, and thus satisfies the conditions defined in Definition 3.1. Although we omit the spiral transition curves here, there is still enough precision for evaluation purposes.

4.3 An overview of the cost function

As mentioned in chapter 2, the costs associated with highway transportation include planning and administrative costs, construction costs, maintenance costs, user costs, and environmental costs. Since planning and administrative costs are insensitive to alignment alternatives, they will be excluded from the proposed model. Note that most existing models only consider in their objective function construction costs, or even just one of its components such as earthwork cost. The absence of a comprehensive cost function in existing models seems due to the lack of studies on the relations between these cost items and the design variables and also, possibly to the mathematical difficulties of minimizing intricate cost functions with previously available optimization methods. In this book, we attempt to establish a relatively comprehensive objective function (i.e., to satisfy condition (1) of a good model, as described in section 2.7). The solution algorithm is then developed based on the nature of the final model Thus, we endeavor to solve the actual problem, rather than fit the problem into an existing solution tool.

The way we formulate the objective function is to categorize all cost items into location-dependent, length-dependent, area-dependent, volume-dependent and VMT-dependent costs. With this approach, all cost items can be easily related to the design variables. This is very important because the objective function must be able to reflect the features of the alignment, which is described by the set of decision variables. In the previous section, we proposed Algorithm 4.1 to generate the alignment based on a given set of decision variables. (In the proposed model, these are the coordinates of the set of intersection points.) With the resulting alignment, we must quantify all dominating and sensitive costs in terms of decision variables. Assuming that the projected traffic demand is given and the number of traffic lanes is fixed, then area-dependent cost (e.g., pavement cost) and VMT-dependent cost (e.g., environmental cost) can be further converted into length-dependent cost. Without the detailed vertical profile of the alignment, we are not able to precisely calculate the volume-dependent cost (e.g., earthwork cost) because we are optimizing the horizontal alignment here. However, to some extent, earthwork cost is also location-dependent. For example, an alignment alternative passing through a mountainous cell will incur higher earthwork cost than one passing through a flat zone. Although the relation is loose, it does exist and we will provide some mechanisms to clarify it. The only major cost that cannot be easily classified into a single category is user cost. As discussed in chapter 2, user costs include vehicle operating, accident, and travel time cost, which are affected by different features of the alignment. Therefore, we need an additional section to discuss them. Note that the estimations of user costs are highly dependent on future traffic, which cannot be precisely forecast. For design, Annual Average Daily Traffic ($AADT$) is usually employed. The prediction of traffic demand is beyond the scope of this book. We assume that $AADT$ is obtained from other studies and is considered to be given information for our proposed model. The precision of $AADT$ is not considered an issue here.

Since each of the aforementioned cost items consists of complex computations, we will discuss them separately in different sections. In the next one, the estimation of location-dependent costs is presented. Section 4.5 will explain the estimation of length-dependent costs. The estimation of user costs is then discussed in section 4.6.

4.4 Location-dependent costs

Each cell in the study region represents a zone of different construction cost. Building a highway through different cells may incur different costs. Let $C(i, j)$ denote the cell bounded by $x = x_O + iD_x$, $x = x_O + (i+1)D_x$, $y = y_O + jD_y$, and $y = y_O + (j+1)D_y$ (see Figure 4.9). Then for each $C(i, j)$, the associated cost $K_N(i, j)$ is determined by

$$K_N(i, j) = K_R(i, j) + K_S(i, j) + K_I(i, j) + K_E(i, j)\ , \tag{4.13}$$

where $K_N(i, j)$ = unit location-dependent cost of $C(i, j)$ $(\$/\mathrm{ft}^2)$
$K_R(i, j)$ = unit right-of-way (land acquisition) cost of $C(i, j)$ $(\$/\mathrm{ft}^2)$
$K_S(i, j)$ = unit soil stabilization cost of $C(i, j)$ $(\$/\mathrm{ft}^2)$
$K_I(i, j)$ = index of availability or unit environmental impact of $C(i, j)$ $(\$/\mathrm{ft}^2)$
$K_E(i, j)$ = unit earthwork cost of $C(i, j)$ $(\$/\mathrm{ft}^2)$.

In eqn (4.13), $K_R(i, j)$ can be calculated according to the unit land value of $C(i, j)$, while $K_S(i, j)$ is estimated based on the soil type of $C(i, j)$. $K_I(i, j)$ is an index used to identify inaccessible cells, environmentally sensitive areas, or historical preservation areas, where the alignment is not allowed to pass. If somehow a cell is inaccessible or sensitive, then $K_I(i, j)$ will be assigned to a very large positive number. Since our objective is to achieve the minimal cost, a high cost cell will be avoided during the optimum search.

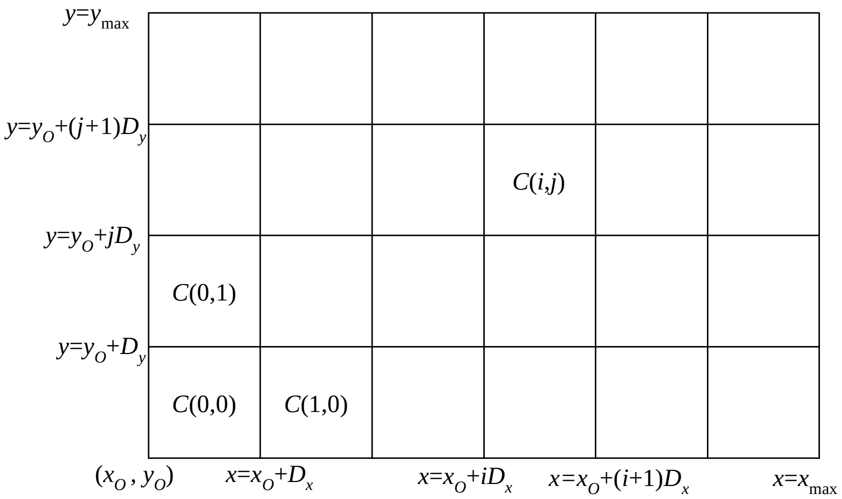


Figure 4.9: Cell definition for the study region.

$K_I(i, j)$ is very useful in formulating the problem. For example, if the region of interest is not rectangular, it can be modeled as a collection of cells in a rectangular grid, where inaccessible regions are represented by cells with high availability cost via $K_I(i, j)$. Thus, any study region could be similarly transformed into a format acceptable to the proposed model. Figure 4.10 shows an example of such mapping.

The unit earthwork cost $K_E(i, j)$ cannot be precisely calculated without the actual vertical alignment. However, one can imagine that if the alignment passes through a cell whose elevation is much different from the elevation at start and end points, then the resulting earthwork cost will be higher. Based on this observation, Parker (1977) has developed a linear programming regression model to construct the projected road surface and estimate the earthwork volume for each cell. (For more detailed discussion, refer to section 2.4.2.) Another approach to estimate earthwork cost without presenting a vertical profile is proposed by Sthapit and Mori (1994). In their model, the earthwork cost is estimated by linear regression analysis, given the ground elevations along the horizontal alignment. The independent variables are the average slope of the ground profile (called the slope factor) and the average difference between the actual change in ground elevations and the maximum allowable change in ground elevations (called the hill factor). The coefficient of determination (R^2) shows that their model has significant statistical performance.

Both methods for estimating earthwork cost should be valid for the proposed model; However, Parker's approach is preferred because it also uses a data format which is consistent with the proposed model.

Here we assume that $K_N(i, j)$ only includes land acquisition cost, soil stabilization cost, availability or environmental impact, and earthwork cost. In fact, more cost items can be added as long as the cost information is available and all components are properly weighted according to their relative importance.

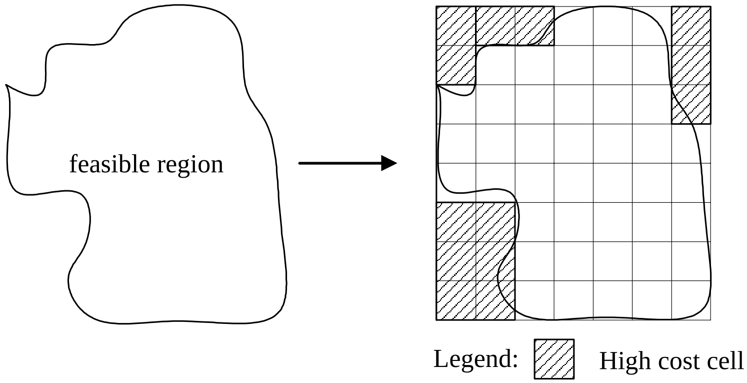


Figure 4.10: The transformation of a non-rectangular shape into a rectangle.

To compute location-dependent cost for an alignment alternative, we must first locate the cells through which the alignment passes and then calculate the costs incurred in each cell. To simplify, we assume that the highway width is fixed. Then the associated location-dependent cost C_N for a highway alignment is

$$C_N = W\left[\sum_{i=0}^{I_{\max}-1}\sum_{j=0}^{J_{\max}-1} L(i,j)K_N(i,j)\right], \tag{4.14}$$

where $I_{\max x} = (x_{\max} - x_O)/D$ is the maximal cell index in X coordinate

$J_{\max y} = (y_{\max} - y_O)/D$ is the maximal cell index in Y coordinate

$L(i,j)$ = the length of the alignment in $C(i,j)$

W = the width of the highway, which is assumed to be fixed along the alignment.

In eqn (4.14), $L(i,j)$ is a function of the set of decision variables d_i ’s. However, the function form is not explicit. In fact, we are unable to express $L(i,j)$ explicitly in terms of d_i ’s. $L(i,j)$ is computable only after d_i ’s are specified. Once d_i ’s are given, a human could easily calculate $L(i,j)$ by looking at the map to see what cells the alignment passes through, and then calculate the location-dependent costs incurred in each cell. Finally, C_N can be obtained by applying eqn (4.14). Unfortunately, computers do not possess this visual intuition, and therefore, must be programmed more precisely to identify the cells that the alignment passes through, and then calculate the associated location-dependent cost.

As discussed in the previous section, the alignment generated by Algorithm 4.1 contains tangent sections and circular curves. For notational convenience, we further denote $T_0 = P_0 = S$ and $C_{n+1} = P_{n+1} = E$ as the start and end points of the alignment. Then we observe that T_i and C_{i+1} are linked by a straight-line section for all $i = 0,......,n$, whereas C_i and T_i are connected by a circular curve with radius R_i for all $i = 1,......,n$ (see Figure 4.11 for an example). In some extreme cases, where the tangent section between two intersection points is completely eliminated by two circular curves, the point of tangency pertaining to one intersection point will coincide with the point of curvature pertaining to the next intersection point. For example in Figure 4.11, T_4 and C_5 are the same point.

The coordinates of C_i can be obtained by translating P_i along the vector $\overrightarrow{P_iP_{i-1}}$ (or equivalently $\mathbf{P}_{i-1} - \mathbf{P}_i$) through a distance $L_T(i)$. Likewise, the coordinates of T_i are determined by translating P_i along the vector $\overrightarrow{P_iP_{i+1}}$ (or equivalently $\mathbf{P}_{i+1} - \mathbf{P}_i$) through a distance $L_T(i)$. In mathematical forms, we obtain

$$\mathbf{C}_i = \begin{bmatrix} x_{C_i} \\ y_{C_i} \end{bmatrix} = \mathbf{P}_i + L_T(i)\frac{\mathbf{P}_{i-1} - \mathbf{P}_i}{\left\| \mathbf{P}_{i-1} - \mathbf{P}_i \right\|}$$

$$= \begin{bmatrix} x_{P_i} + L_T(i)(x_{P_{i-1}} - x_{P_i}) \Big/ \sqrt{(x_{P_{i-1}} - x_{P_i})^2 + (y_{P_{i-1}} - y_{P_i})^2} \\ y_{P_i} + L_T(i)(y_{P_{i-1}} - y_{P_i}) \Big/ \sqrt{(x_{P_{i-1}} - x_{P_i})^2 + (y_{P_{i-1}} - y_{P_i})^2} \end{bmatrix}, \qquad (4.15)$$

$$\mathbf{T}_i = \begin{bmatrix} x_{T_i} \\ y_{T_i} \end{bmatrix} = \mathbf{P}_i + L_T(i)\frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{\left\| \mathbf{P}_{i+1} - \mathbf{P}_i \right\|}$$

$$= \begin{bmatrix} x_{P_i} + L_T(i)(x_{P_{i+1}} - x_{P_i}) \Big/ \sqrt{(x_{P_{i+1}} - x_{P_i})^2 + (y_{P_{i+1}} - y_{P_i})^2} \\ y_{P_i} + L_T(i)(y_{P_{i+1}} - y_{P_i}) \Big/ \sqrt{(x_{P_{i+1}} - x_{P_i})^2 + (y_{P_{i+1}} - y_{P_i})^2} \end{bmatrix}. \qquad (4.16)$$

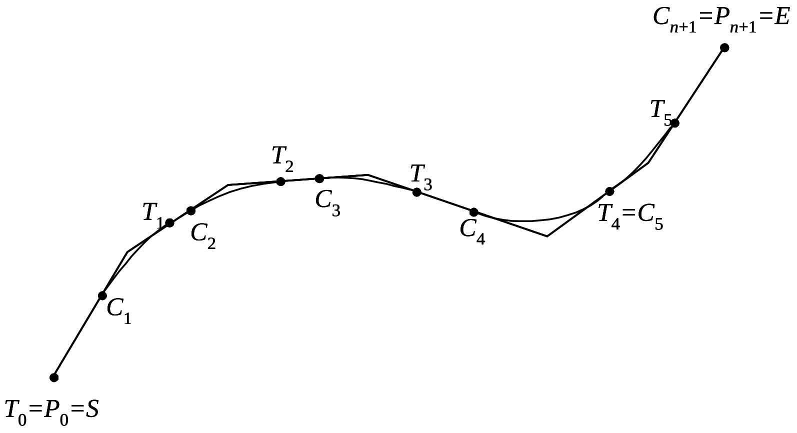


Figure 4.11: An example of points of tangency and curvature.

Since the logical and mathematical requirements for determining the length of the alignment in each cell are different for tangent sections and circular curves, they will be discussed separately in the following subsections:

4.4.1 Location-dependent costs of tangent sections

Since each of the cells is a convex set, any linear combination of two points in a cell will fall into the same cell. It is noted that each tangent section will intersect horizontal cell grids (parallel to *X* coordinate) and/or vertical cell grids (parallel to *Y* coordinate), or none, depending on the slope and length of the section. Therefore, if we can find all of these intersection points and sort them in either the *X* or *Y* coordinate, then the mid-point of any two consecutive points will give a clue about the cell where the line segment occurs.

Let $Link(i)$ be the link connected by T_i and C_{i+1} for all $i = 0,.......,n$. Then the slope of $Link(i)$, denoted by m_i, is

$$m_i = (y_{p_{i+1}} - y_{p_i})/(x_{p_{i+1}} - x_{p_i})\,. \tag{4.17}$$

The value of m_i determines how $Link(i)$ intersects the cell grids and the sorting basis. If $m_i = 0$ (i.e., $y_{P_{i+1}} = y_{P_i}$), then $Link(i)$ only intersects vertical grids. The intersection points are then sorted by the magnitude of their X coordinates. If m_i approaches $\pm\infty$ (i.e., $x_{P_{i+1}} = x_{P_i}$), then $Link(i)$ only intersects horizontal grids. Therefore, the intersection points are sorted by the magnitude of their Y coordinates. If $x_{P_{i+1}} \neq x_{P_i}$ and $y_{P_{i+1}} \neq y_{P_i}$ (the most general case), we should find out all intersection points of $Link(i)$ at both vertical and horizontal grids, and then sort them according to their X coordinates.

Let $Q_1^{(i)}$, $Q_2^{(i)}$,……., $Q_{l_i}^{(i)}$ be the resulting points after sorting, including two end points of $Link(i)$ (see Figure 4.12 for a general example), and assume that the coordinates of $Q_j^{(i)}$ are represented by $(x_{Q_j}^{(i)}, y_{Q_j}^{(i)})$. Then the line segment between two consecutive points $Q_j^{(i)}$ and $Q_{j+1}^{(i)}$ will exist in exactly one cell with the following indexes:

1st index:
$$\left\lceil \frac{(x_{Q_j}^{(i)} + x_{Q_{j+1}}^{(i)})/2 - x_O}{D_x} \right\rceil^{-}, \tag{4.18}$$

2nd index:
$$\left\lceil \frac{(y_{Q_j}^{(i)} + y_{Q_{j+1}}^{(i)})/2 - y_O}{D_y} \right\rceil^{-}, \tag{4.19}$$

where $\lceil \cdot \rceil^{-}$ denotes the truncated integer value of its argument.

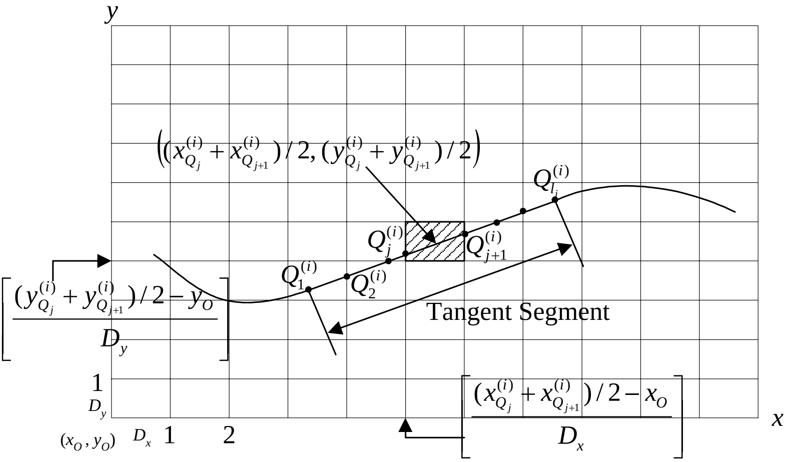

y
$\left((x_{Q_j}^{(i)} + x_{Q_{j+1}}^{(i)})/2, (y_{Q_j}^{(i)} + y_{Q_{j+1}}^{(i)})/2\right)$
$Q_{l_i}^{(i)}$
$Q_j^{(i)}$
$Q_{j+1}^{(i)}$
$Q_1^{(i)}$
$Q_2^{(i)}$
Tangent Segment
$\left\lceil \frac{(y_{Q_j}^{(i)} + y_{Q_{j+1}}^{(i)})/2 - y_O}{D_y} \right\rceil$
1
D_y
(x_O, y_O)
D_x
1
2
$\left\lceil \frac{(x_{Q_j}^{(i)} + x_{Q_{j+1}}^{(i)})/2 - x_O}{D_x} \right\rceil$
x

Figure 4.12: Sorted intersection points of a tangent section.

The indexes calculated with eqns (4.18) and (4.19) indicate the cell where the j^{th} segment of $Link(i)$ is located. To simplify notation, let the unit location-dependent cost of the cell be further denoted by ${}_L K_{N\cdot j}^{(i)}$. Then the location-dependent costs of the alignment along all tangent sections can be calculated as

$$C_N^T = W\left[\sum_{i=0}^{n}\sum_{j=1}^{l_i-1} {}_L K_{N\cdot j}^{(i)} L_j^{(i)}\right], \tag{4.20}$$

where C_N^T = the location-dependent costs of the alignment along all tangent sections

${}_L K_{N\cdot j}^{(i)}$ = the unit location-dependent cost of the cell where the j^{th} segment of $Link(i)$ is located

$L_j^{(i)} = \left((x_{Q_j}^{(i)} - x_{Q_{j+1}}^{(i)})^2 + (y_{Q_j}^{(i)} - y_{Q_{j+1}}^{(i)})^2\right)^{1/2}$ is the length of j^{th} segment of $Link(i)$, i.e., the distance between $Q_j^{(i)}$ and $Q_{j+1}^{(i)}$

l_i = number of intersection points of $Link(i)$.

With the above approach, we would be able to calculate the location-dependent costs for the tangent sections of a given alignment. The algorithm for calculating C_N^T is summarized as follows:

Algorithm 4.2 Computation of location-dependent costs for tangent sections

```
(1) Step 1: Initialization
      Set $C_N^T = 0$
      Set $i = 0$
(2) Step 2: Obtain the sorted intersection points $Q_1^{(i)}$, $Q_2^{(i)}$,……., $Q_{l_i}^{(i)}$
      2-1 If $i > n$, then stop and return $C_N^T$; Otherwise continue to 2-2.
      2-2 Find all points at which $Link(i)$ intersects horizontal and vertical grids
      2-3 If $x_{P_{i+1}} = x_{P_i}$, sort the set of points on the $Y$ coordinates; Otherwise,
          sort them on the $X$ coordinates
      2-4 Obtain $Q_1^{(i)}$, $Q_2^{(i)}$,……., $Q_{l_i}^{(i)}$
(3) Step 3: Calculate the cost for each line segment and add it to $C_N^T$
      Set $j = 1$
      3-1 If $j > l_i - 1$, then set $i = i + 1$ and go to step 2; Otherwise continue to
          3-2
      3-2 Use eqns (4.18) and (4.19) to determine the indexes of the cell where
          the $j^{th}$ segment of $Link(i)$ is located
      3-3 Let $C_N^T = C_N^T + K_{N\cdot j}^{(i)} L_j^{(i)} W$
      3-4 Let $j = j + 1$; go to 3-1
```

4.4.2 Location-dependent costs of circular curves

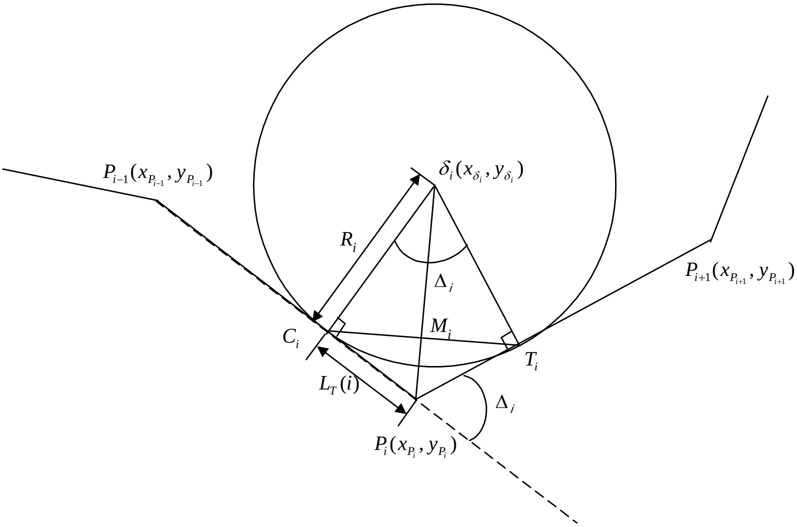
The computation of location-dependent costs for the circular curves of a given alignment is relatively difficult because the linear combination of any two points on a curve may deviate from the curve, and thus we are unable to apply the convex property of the cells. Alternatively, we may need other mechanisms to identify the cells where each segment of the circular curves is located.

Let $Arc(i)$ be the circular curve from C_i to T_i for all $i = 1, \ldots\ldots, n$. Then three parameters are required for completely describing the behavior of $Arc(i)$. They are the point of curvature $C_i(x_{C_i}, y_{C_i})$, the point of tangency $T_i(x_{T_i}, y_{T_i})$, and the center of the circular curve $\delta_i(x_{\delta_i}, y_{\delta_i})$. The coordinates of C_i and T_i can be obtained with eqns (4.15) and(4.16). As to δ_i, given the radius of the circular curve R_i (obtained from Algorithm 4.1), there are many ways to locate δ_i. The simplest approach uses vector analysis. Let $M_i(x_{M_i}, y_{M_i})$ be the middle point of the line segment between C_i and T_i as shown in Figure 4.13. Then it can be verified by trigonometry that M_i is also located on the line segment connecting P_i and δ_i. Accordingly $\delta_i(x_{\delta_i}, y_{\delta_i})$ can be derived by extending the vector $\overrightarrow{P_i M_i}$ from P_i to δ_i. In mathematical forms, we obtain

$$\mathbf{M}_i = \begin{bmatrix} x_{M_i} \\ y_{M_i} \end{bmatrix} = \frac{1}{2}\left(\mathbf{C}_i + \mathbf{T}_i\right) = \begin{bmatrix} (x_{C_i} + x_{T_i})/2 \\ (y_{C_i} + y_{T_i})/2 \end{bmatrix}, \tag{4.21}$$

$$\boldsymbol{\delta}_i = \begin{bmatrix} x_{\delta_i} \\ y_{\delta_i} \end{bmatrix} = \mathbf{P}_i + R_i \sec\frac{\Delta_i}{2} \frac{\mathbf{M}_i - \mathbf{P}_i}{\left\|\mathbf{M}_i - \mathbf{P}_i\right\|}. \tag{4.22}$$

where Δ_i is obtained from eqn (4.10).



$P_{i-1}(x_{P_{i-1}}, y_{P_{i-1}})$
$\delta_i(x_{\delta_i}, y_{\delta_i})$
R_i
Δ_i
M_i
C_i
T_i
$P_{i+1}(x_{P_{i+1}}, y_{P_{i+1}})$
$L_T(i)$
Δ_i
$P_i(x_{P_i}, y_{P_i})$

Figure 4.13: The geometric relations among C_i, T_i, δ_i, and M_i.

We now derive the function of the circular curve as

$$(x - x_{\delta_i})^2 + (y - y_{\delta_i})^2 = R_i^2 \,. \tag{4.23}$$

In order to calculate the location-dependent cost, we must determine the cells through which the circular curve passes. To do so, first we have to find the points at which the curve intersects the grids. As we know, if the circle intersects the grids, it will intersect each grid at two distinct points due to the symmetric property of the circle unless it is just tangent to the grids. The ranges to be considered in solving the above equation are

$\left[\min(x_{P_i}, x_{C_i}, x_{T_i}),\quad \max(x_{P_i}, x_{C_i}, x_{T_i})\right]$ for the X interval, and
$\left[\min(y_{P_i}, y_{T_i}, y_{T_i}),\quad \max(y_{P_i}, y_{C_i}, y_{T_i})\right]$ for the Y interval.

Let $B_j^{(i)}$ ’s be the set of points at which $Arc(i)$ intersects the cell grids within the above ranges. Then we must sort $B_j^{(i)}$ ’s in sequence so that we can identify the cell in which the curve segment connecting two successive intersection points is located. To sort $B_j^{(i)}$ ’s, the following measurement is employed as the sorting base

$$D_j^{(i)} = \left\| \mathbf{B}_j^{(i)} - \mathbf{C}_i \right\| , \tag{4.24}$$

where $D_j^{(i)}$ = the distance between $B_j^{(i)}$ and C_i.

Let $O_1^{(i)}$, $O_2^{(i)}, \ldots, O_j^{(i)}, \ldots., O_{a_i}^{(i)}$ be the set of points after sorting, including the two end points of $Arc(i)$, where $O_j^{(i)}$ denotes the j^{th} intersection points of $Arc(i)$. (See Figure 4.14 for a general example.) Further assume that the coordinates of $O_j^{(i)}$ are represented by $(x_{O_j}^{(i)}, y_{O_j}^{(i)})$. It is then clear that the arc segment between $O_j^{(i)}$ and $O_{j+1}^{(i)}$ will exist in exactly one cell. Suppose that the middle point of the arc segment, denoted by $F_j^{(i)}$, is selected to indicate the cell. Then the coordinates of the middle point can be obtained with eqn (4.25). The vector representation of the analysis is illustrated in Figure 4.15.

$$\mathbf{F}_j^{(i)} = \begin{bmatrix} x_{F_j}^{(i)} \\ y_{F_j}^{(i)} \end{bmatrix} = \boldsymbol{\delta}_i + R_i \frac{\mathbf{O}_j^{(i)} + \mathbf{O}_{j+1}^{(i)} - 2\boldsymbol{\delta}_i}{\left\| \mathbf{O}_j^{(i)} + \mathbf{O}_{j+1}^{(i)} - 2\boldsymbol{\delta}_i \right\|} \,. \tag{4.25}$$

With the coordinates of $F_j^{(i)}$, the indexes of the cell through which an arc segment connects $O_j^{(i)}$ and $O_{j+1}^{(i)}$ are as follows:

1st index:
$$\left\lceil \frac{x_{F_j}^{(i)} - x_O}{D_x} \right\rceil^{-} , \tag{4.26}$$

2nd index: $$\left\lceil \frac{y_{F_j}^{(i)} - y_O}{D_y} \right\rceil^{-},$$ (4.27)

where $\lceil \cdot \rceil^{-}$ denotes the truncated integer value of its argument.

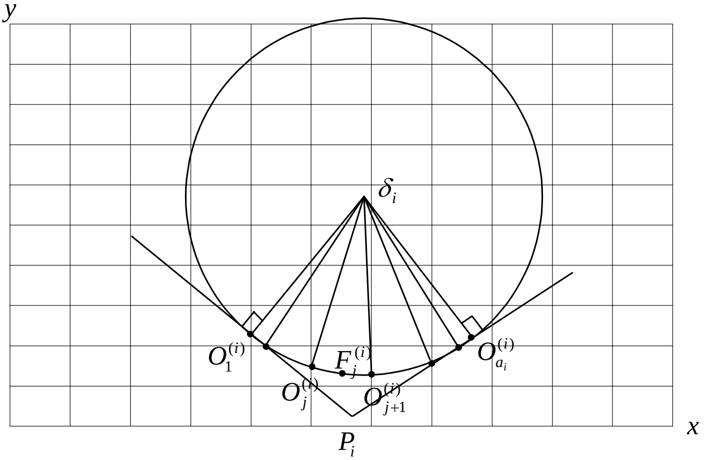


Figure 4.14: Sorted intersection points of a circular curve.

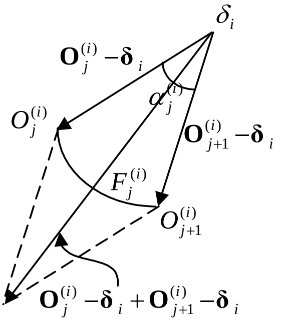

δ_i
$\mathbf{O}_j^{(i)} - \boldsymbol{\delta}_i$
$O_j^{(i)}$
$\alpha_j^{(i)}$
$\mathbf{O}_{j+1}^{(i)} - \boldsymbol{\delta}_i$
$F_j^{(i)}$
$O_{j+1}^{(i)}$
$\mathbf{O}_j^{(i)} - \boldsymbol{\delta}_i + \mathbf{O}_{j+1}^{(i)} - \boldsymbol{\delta}_i$

Figure 4.15: The vector presentation of eqns (4.25) and (4.29).

The indexes calculated with eqns (4.26) and (4.27) indicate the cell where the j^{th} segment of $Arc(i)$ is located. Again, for notational convenience, let the unit location-dependent cost of the cell be denoted by ${}_A K_{N \cdot j}^{(i)}$. Then the location-dependent costs of the alignment along all circular curves are

$$C_N^A = W\left[\sum_{i=1}^{n}\sum_{j=1}^{a_i-1} {}_A K_{N\cdot j}^{(i)} A_j^{(i)}\right], \tag{4.28}$$

where C_N^A = the location-dependent costs of the alignment along all circular curves

${}_A K_{N\cdot j}^{(i)}$ = the unit location-dependent cost of the cell in which the j^{th} segment of $Arc(i)$ is located

$A_j^{(i)}$ = the length of j^{th} segment of $Arc(i)$

a_i = number of intersection points of $Arc(i)$.

The length of the j^{th} segment of $Arc(i)$ in eqn (4.28) is determined by

$$A_j^{(i)} = R_i\alpha_j^{(i)} = R_i \cos^{-1}\left(\frac{(\mathbf{O}_j^{(i)}-\boldsymbol{\delta}_i)\cdot(\mathbf{O}_{j+1}^{(i)}-\boldsymbol{\delta}_i)}{\left\|\mathbf{O}_j^{(i)}-\boldsymbol{\delta}_i\right\|\left\|\mathbf{O}_{j+1}^{(i)}-\boldsymbol{\delta}_i\right\|}\right), \tag{4.29}$$

where $\alpha_j^{(i)}$ is the angle subtended by the j^{th} segment of $Arc(i)$, (see Figure 4.15).

We now summarize the algorithm for computing the location-dependent costs for the circular curves of a given alignment as follows:

Algorithm 4.3 Computation of location-dependent costs for circular curves

```
(1) Step 1: Initialize
      Set $C_N^A = 0$
      Set $i = 1$
(2) Step2: Obtain the sorted intersection points $O_1^{(i)}$, $O_2^{(i)}$,……., $O_{a_i}^{(i)}$
      2-1 If $i > n$, then stop and return $C_N^A$; Otherwise continue to 2-2.
      2-2 Use eqns (4.21) and (4.22) to compute the coordinates of the center of
          each circular curve.
      2-3 Use eqn (4.23) to find all points at which $Arc(i)$ intersects with
          horizontal and vertical grids
      2-4 Use eqn (4.24) to calculate $D_j^{(i)}$ for all points
      2-5 Obtain $O_1^{(i)}$, $O_2^{(i)}$,……., $O_{a_i}^{(i)}$ by sorting the points on their $D_j^{(i)}$ values
(3) Step 3: Calculate the cost for each arc segment and add it to $C_N^A$
      Set $j = 1$
      3-1 If $j > a_i - 1$, then set $i = i + 1$ and go to step 2; Otherwise continue
      3-2 Use eqns (4.25) to (4.27) to determine the indexes of the cell where the
          $j^{th}$ segment of $Arc(i)$ is located
      3-3 Use eqn (4.29) to compute $A_j^{(i)}$
      3-4 Let $C_N^A = C_N^A + K_{N\cdot j}^{(i)} A_j^{(i)} W$
      3-5 Let $j = j + 1$; go to 3-1
```

With Algorithms 4.2 and 4.3, we can obtain the location-dependent costs for the tangent sections and circular curves of an alignment. Then the total location-dependent cost C_N is given by

$$C_N = C_N^T + C_N^A . \tag{4.30}$$

Note that in both Algorithms 4.2 and 4.3, a sorting procedure is required. Sorting is a very common problem for which many good algorithms have been developed. The one adopted in this book is called "Quicksort". It is discussed in detail by Cormen *et al* (1996).

4.5 Length-dependent costs

The computation of length-dependent cost for a highway is straightforward. Basically it is the multiplication of the highway length and the unit length-dependent cost. Recall that for an alignment generated by Algorithm 4.1, T_i and C_{i+1} are linked by a straight-line section for all $i = 0,......,n$, whereas C_i and T_i are connected by a circular curve with radius R_i for all $i = 1,......,n$ (see Figure 4.11). Therefore, the total length of the alignment, denoted by L_n, is expressed as

$$L_n = \sum_{i=0}^{n} \sqrt{(x_{T_i} - x_{C_{i+1}})^2 + (y_{T_i} - y_{C_{i+1}})^2} + \sum_{i=1}^{n} R_i \Delta_i \ . \tag{4.31}$$

where L_n = total length of the alignment

(x_{C_i}, y_{C_i}) = the coordinates of C_i, which are given in eqn (4.15)

(x_{T_i}, y_{T_i}) = the coordinates of T_i, which are given in eqn (4.16)

R_i = the radius of the circular curve from C_i to T_i. It is obtained with Algorithm 4.1

Δ_i = the intersection angle at intersection point P_i, obtained with eqn (4.10)

The unit length-dependent cost includes unit construction cost, unit maintenance cost, and unit environmental cost. The construction cost per unit length can be broken down into detailed miscellaneous items, such as fences and guardrails, provided the information is available. Here only pavement cost is considered because it is a dominating and sensitive cost. Although pavement cost is supposed to be area-dependent, it can be further converted to be length-dependent as long as the highway width is fixed. The environmental cost is VMT-dependent. However, if the projected traffic demand is given (it is usually forecast in the planning stage), the environmental cost can be further transformed to a length-dependent cost. Let $AADT$ be the projected two-way traffic volume, which we assume will increase at a yearly rate r_t throughout the whole analysis period n_y. Then the net present value of the environmental cost per unit length is

$$K_{E\cdot L} = \frac{1}{5280} K_{E.V} \cdot 365 \cdot AADT \cdot \sum_{k=1}^{n_y} \left[\frac{(1+r_t)}{(1+\rho)} \right]^k , \tag{4.32}$$

where $K_{E\cdot L}$ = the net present value of unit length-dependent cost for environment impact ($/ft)
$K_{E\cdot V}$ = the unit environmental cost per VMT ($/VMT)
$AADT$ = Annual Average Daily Traffic (two-way) (vehicles/day)
r_t = the annual growth rate of $AADT$ (decimal fraction)
ρ = the assumed interest rate (decimal fraction).

In the above equation the unit environmental cost per VMT is derived by summing up the unit costs for different environmental impacts such as air pollution, noise pollution, water pollution and so forth. The information is provided in Table 2.2, where the units must be converted from cents per VMT to dollars per VMT so that eqn (4.32) has consistent units. It is noted that the computation of eqn (4.32) is slightly laborious. A form of (4.32) which incorporates smoothly compounded traffic growth at the same annual rate of r_t is given below (Wright, 1996):

$$K_{E\cdot L} = \frac{1}{5280} K_{E.V} \cdot 365 \cdot AADT \cdot \frac{e^{(r_t-\rho)n_y} - 1}{r_t - \rho} . \tag{4.33}$$

The comprehensive unit length-dependent cost is then

$$K_L = K_P + K_M + K_{E.L} , \tag{4.34}$$

where K_L = the total unit length-dependent cost ($/ft)
K_P = the pavement cost per unit length ($/ft)
K_M = the maintenance cost per unit length ($/ft)
$K_{E\cdot L}$ = the net present value of unit length-dependent cost for environment impact ($/ft), given in eqn (4.33).

Finally the total length-dependent cost is

$$C_L = K_L L_n , \tag{4.35}$$

where C_L = the total length-dependent cost of an alignment ($)
K_L = unit length-dependent cost, as given in eqn (4.34) ($/ft)
L_n = the total length of the alignment (ft), obtained with eqn (4.31).

4.6 User costs

As mentioned in chapter 2, user costs include vehicle-operating costs, travel time costs, and accident costs. The computation of user costs is less direct because the relations between user costs and highway design features are not explicit. In general, the relations were calibrated by statistical analysis. Note that for evaluating the user costs of a proposed highway, a macroscopic model is appropriate because in the long run, user costs are not sensitive to tiny changes in traffic behavior during a short period. In fact, the errors in estimating user costs are mainly ascribed to the precision of the traffic demand forecast rather than the aggregation of a macroscopic model. Note that the formulations of user costs developed in this section are more appropriate for two-lane rural highways. For other types of roadways, such as multi-lane highways or urban streets, the formulation for each component of user costs may have to be modified to properly reflect the relations between the cost items and the decision variables.

Besides the geometric features of a highway, such as curvature, gradient, and the length of the alignment, the relations between user costs and alignment configuration are linked by the average running speed. For example, in deriving travel time cost, we need the length of the alignment and the average running speed, which is a function of the geometric design features. Many models have been developed for estimating operational speed based on highway geometric design. For example Lamm *et al* (1987), and Krammes *et al* (1995) presented models for estimating the 85th percentile speed. However, their models may not be appropriate for our analysis because an average running speed may be more suitable for evaluation purpose. In the literature, the model that estimates average running speed based on various design features and traffic situations is the one developed by Polus *et al* (1984). The same model was also presented in Polus and Eck (1987) for evaluating geometric design alternatives. That model was developed for estimating average running speed on two-lane rural highways. Since models for estimating average running speed on other types of highways were not found, Polus's work will be adopted as a general case in this book. After other models are developed and justified, Polus's formula can be replaced.

Polus's (1984 and 1987) employed multiple linear regression to derive the relation between average running speed (dependent variable) and geometric features as well as traffic characteristics (independent variables). Several functional forms were investigated. Among them, the two-regime model (for traffic flow $\leq$ 200 and traffic flow > 200) best fits the data. However, Polus stated that the one-regime model is likely to be sufficiently precise for practical applications. The model form is

$$\begin{aligned}\overline{V} &= 88.714 - 0.094\overline{C} - 0.282\overline{H} - 0.069G_N - 0.022T \\ &\quad - 3.981D - 0.27Q\end{aligned}, \qquad (4.36)$$

where $\overline{V}$ = average running speed (km/h)

$\overline{C}$ = average curvature (degrees/km) (some authors call it average bendiness)
$\overline{H}$ = average hilliness (m/km)
G_N = net gradient (m/km)
T = T factor, i.e., the percentage of heavy vehicles in the traffic stream (decimal fraction)
D = D factor, i.e., the directional distribution of traffic (decimal fraction),
Q = hourly one-way traffic volume (vph).

Eqn (4.36) is in metric units. If English units are employed, we can convert it to

$$\begin{aligned}\overline{V} = 55.124 - 0.0363\overline{C} - 0.0332\overline{H} - 0.0081G_N - 0.0137T \\ - 2.4737D - 0.1678Q\end{aligned}, \qquad (4.37)$$

where $\overline{V}$ = average running speed (miles/hr)
$\overline{C}$ = average curvature (degrees/mile)
$\overline{H}$ = average hilliness (ft/mile)
G_N = net gradient (ft/mile)
T, D, and Q maintain the same units as in eqn (4.36).

For consistency, eqn (4.37) will be adopted in this book. To apply this equation, we must first determine the values of independent variables based on the alignment configuration and the forecast traffic demand. The first three independent variables are the geometric features of the alignment. Their values are obtained with the following equations:

$$\overline{C} = \sum_{i=1}^{n} \frac{\Delta_i (180/\pi)}{(L_n / 5280)} = \frac{950400}{\pi L_n} \sum_{i=1}^{n} \Delta_i , \qquad (4.38)$$

where Δ_i = the intersection angle at intersection point P_i (radians), given in eqn (4.10)
L_n = the total length of the alignment (ft), obtained with eqn (4.31).

$$\overline{H} = \sum_{i=1}^{m} \frac{h_i}{(L_n / 5280)} = \frac{5280}{L_n} \sum_{i=1}^{m} h_i , \qquad (4.39)$$

where h_i = the vertical distance between the i^{th} crest and the following $(i+1)^{th}$ sag or vice versa (ft)
m = total number of successive crests and sags.

$$G_N = \frac{|z_S - z_E|}{(L_n / 5280)}, \tag{4.40}$$

where z_S = the Z coordinate (elevation) of start point S (ft)
z_E = the Z coordinate (elevation) of end point E (ft).

Note that the proposed model is intended to optimize horizontal alignment, and thus provides no information about the vertical profile. Accordingly, $\overline{H}$ may have to be omitted in the analysis. However, $\overline{H}$ will be taken into account in the 3-dimensional models developed later in this book.

The last three independent variables in eqn (4.37) are employed to describe the traffic characteristics. They are given values for highway design. Since traffic patterns vary from time to time, AASHTO (1977) recommended that traffic within peak and off-peak periods be analyzed separately, and a model day of 18 hours is suggested, with traffic in six hours from 12 midnight to 6 a.m. added to the off-peak period.

Let H_p be the number of peak hours per day. Then, based on the AASHTO assumptions (i.e., in a year, there are 253 weekdays, 104 weekends and 8 holidays, which are equivalent to 56 weekdays), the total numbers of peak hours H_{Tp} and off-peak hours H_{To} per year are

$$H_{Tp} = 309 H_p \text{ (hours)}, \tag{4.41}$$

$$H_{To} = 6570 - 309 H_p \text{ (hours)}. \tag{4.42}$$

To determine peak hour volume in the prevalent direction, the following equation is employed (see McShane and Roess, 1990):

$$Q_{pp} = AADT \cdot K \cdot D, \tag{4.43}$$

where Q_{pp} = peak hour traffic in the prevalent direction (vph)
$AADT$ = Annual Average Daily Traffic (two-way) (vehicles/day)
K = the proportion of daily traffic occurring during the peak hour (decimal)
D = the directional distribution of traffic in peak hour (decimal)

Accordingly, the peak hour volume in the non-prevalent direction, denoted by Q_{pn}, is

$$Q_{pn} = AADT \cdot K \cdot (1 - D). \tag{4.44}$$

For the off-peak period, assume that the traffic volume is the same for both directions (i.e., $D = 0.5$). Then the directional one-way traffic volume Q_o is

$$Q_o = \frac{0.5 \cdot AADT \cdot (1 - H_p K)}{18 - H_p}. \tag{4.45}$$

Note that if the directional distribution for the off-peak period is available, then we can derive the off-peak volume for each direction. However, $D = 0.5$ for off-peak period is a valid assumption because the off-peak traffic is typically the same in both directions. It is also assumed that hourly traffic volume in the morning peak will be equal but opposite in direction to that in the evening peak period. Therefore, for each direction, the hourly traffic has three patterns, Q_{pp} , Q_o , and Q_{pn}. Inserting each traffic volume with other independent variables into eqn (4.37), we can obtain the average running speed for different periods and directions:

$$\begin{aligned}\overline{V}_{pp} &= 55.124 - 0.0363\overline{C} - 0.0332\overline{H} - 0.0081G_N - 0.0137T \\ &\quad - 2.4737D - 0.1678Q_{pp}\end{aligned}, \tag{4.46}$$

$$\begin{aligned}\overline{V}_o &= 55.124 - 0.0363\overline{C} - 0.0332\overline{H} - 0.0081G_N - 0.0137T \\ &\quad - 1.2369 - 0.1678Q_o\end{aligned}, \tag{4.47}$$

$$\begin{aligned}\overline{V}_{pn} &= 55.124 - 0.0363\overline{C} - 0.0332\overline{H} - 0.0081G_N - 0.0137T \\ &\quad - 2.4737(1-D) - 0.1678Q_{pn}\end{aligned}, \tag{4.48}$$

where $\overline{V}_{pp}$, $\overline{V}_o$, and $\overline{V}_{pn}$ are average running speeds under traffic conditions (D, Q_{pp}) , $(0.5, Q_o)$, and $((1-D), Q_{pn})$, respectively.

Eqns (4.46) to (4.48) will be used in the evaluations of vehicle-operating costs and travel time costs later in this section.

4.6.1 Vehicle operating costs

According to AASHTO (1977), Chesher *et al* (1987), and Watanatada *et al* (1987), in highway economic analysis, vehicle-operating costs may include fuel and oil consumption, tire wear, and vehicle depreciation. However, it is found that fuel consumption cost is the most dominating and sensitive cost item among all components. Hence, only fuel consumption cost is considered here.

AASHTO (1977) has developed several nomographs for estimating vehicle-operating costs based on geometric designs and traffic conditions. However, our goal is to computerize the optimization model. Computers cannot check the graphs to get the corresponding costs. Therefore, this approach is less preferable even though we can convert the graphs into tables and use interpolation techniques to obtain the required information. Alternatively, Watanatada (1987) presented a detailed model for estimating fuel consumption cost. The model starts by deriving various resistance forces incurred in running a vehicle and calculating the required vehicle power. Then it estimates fuel consumption cost with the required vehicle power and roadway geometric characteristics. Watanatada's model is believed to yield the most precise estimation of fuel costs. However, it is too detailed and thus may not be appropriate for our case.

The most relevant approach is to derive a functional form to estimate fuel consumption cost based on the geometric designs of highway. Unfortunately, the relation between fuel consumption cost and alignment configuration is not direct. Instead, we may have to establish a model to estimate fuel consumption cost through a variable whose relation with the alignment is explicitly specified. Obviously the best choice is average running speed because the relation between fuel consumption and average running speed is significant, and the average running speed is associated with geometric designs through eqns (4.46) to (4.48). It is noted that the variability of speed influences fuel consumption. However, we are estimating fuel consumption for all traffic and thus the average running speed is employed in the model.

If average running speed is the only variable for predicting the fuel consumption, it is observed that at both lower and higher speed, the fuel consumption is relatively high. In other words, the relation between fuel consumption and average running speed is U-shaped and thus there exists an economical speed for running a vehicle. Chesher *et al* (1987) indicated that the following functional form could be used:

$$F = \gamma_0 + \gamma_1 \frac{1}{\overline{V}} + \gamma_2 \overline{V}^2 , \tag{4.49}$$

where F = fuel consumption per unit distance
$\overline{V}$ = vehicle speed
γ_0, γ_1, and γ_2 are coefficients.

To establish a fuel consumption function, we now need empirical data for calibrating the coefficients. The data employed for regression analysis are adopted from Zaniewski (1983), where the fuel consumption rates for eight types of vehicles at different grades and speeds were presented in table forms. AASHTO (1977) indicated that the representative vehicle classes for economic analysis of highway projects are medium car (for passenger car), 2A single-unit truck and 3-S2 diesel truck (for heavy trucks). Therefore, only these three vehicle classes are considered in this analysis.

Note that in addition to average running speed, the grade is also considered as an independent variable in the regression function. After investigating different structures, it is found that the following function form best fits the data for all three vehicle classes:

$$F = \gamma_0 + \gamma_1 \overline{G} + \gamma_2 \overline{V} + \gamma_3 \overline{V}^2 , \tag{4.50}$$

where F = fuel consumption (gallons/1000 miles)

$\overline{G}$ = grade of road section (%)
$\overline{V}$ = vehicle running speed (mph)

The regression functions for the three vehicle classes are given in eqns (4.51) to (4.53). The detailed statistical analyses are summarized in Table 4.2 to Table 4.4. To visualize the results, the original data and the regression functions are then plotted in Figure 4.16 to Figure 4.21.

$$F_{MC}(\overline{G},\overline{V}) = 72.8769 + 4.1909\overline{G} - 1.7322\overline{V} + 0.0241\overline{V}^2, \tag{4.51}$$

$$F_{2A}(\overline{G},\overline{V}) = 263.8346 + 13.1592\overline{G} - 6.1827\overline{V} + 0.0672\overline{V}^2, \tag{4.52}$$

$$F_{3S}(\overline{G},\overline{V}) = 509.7646 + 29.8595\overline{G} - 9.9014\overline{V} + 0.0892\overline{V}^2, \tag{4.53}$$

where F_{MC} , F_{2A} , and F_{3S} are fuel consumption (gallons/1000 miles) for medium car, 2-axle single unit truck, and 3-S2 diesel truck, respectively
$\overline{G}$ = grade of road section (%)
$\overline{V}$ = vehicle running speed (mph).

Table 4.2: Summary of statistical analysis of fuel consumption: medium car.

Regression Statistics			
Multiple R	0.944181	Adjusted R Square	0.890086
R Square	0.891477	Standard Error	7.860416
Observations	238		

ANOVA					
	df	*SS*	*MS*	*F*	*Significance F*
Regression	3	118767.10	39589.03	640.743	1.7E-112
Residual	234	14457.96	61.78613		
Total	237	133225.10			

	Coeff.	*Standard Error*	*t Statistic*	*P-value*	*Lower 95%*	*Upper 95%*
γ_0	72.877	1.776	41.028	7.40E-109	69.377	76.376
$\gamma_1(\overline{G})$	4.191	0.104	40.296	3.00E-107	3.986	4.396
$\gamma_2(\overline{V})$	-1.732	0.109	-15.898	4.39E-39	-1.947	-1.518
$\gamma_3(\overline{V}^2)$	0.024	0.001	17.033	7.38E-43	0.021	0.027

Table 4.3: Summary of statistical analysis of fuel consumption: 2A-SU truck.

Regression Statistics			
Multiple R	0.940582	Adjusted R Square	0.883217
R Square	0.884695	Standard Error	26.44099
Observations	238		

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	3	1255215.0	418404.9	598.4688	2E-109
Residual	234	163595.4	699.1257		
Total	237	1418810.0			

	Coeff.	*Standard Error*	*t Statistic*	*P-value*	*Lower 95%*	*Upper 95%*
γ_0	263.835	5.975	44.156	1.80E-115	252.063	275.607
$\gamma_1(\overline{G})$	13.159	0.350	37.614	3.50E-101	12.470	13.848
$\gamma_2(\overline{V})$	-6.183	0.367	-16.869	2.59E-42	-6.905	-5.461
$\gamma_3(\overline{V}^2)$	0.067	0.005	14.135	3.31E-33	0.058	0.077

Table 4.4: Summary of statistical analysis of fuel consumption: 3-S2 diesel truck.

Regression Statistics			
Multiple R	0.922545	Adjusted R Square	0.847427
R Square	0.851089	Standard Error	44.91455
Observations	126		

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	3	1406639.0	468879.7	232.4274	2.9E-50
Residual	122	246112.6	2017.316		
Total	125	1652752.0			

	Coeff.	*Standard Error*	*t Statistic*	*P-value*	*Lower 95%*	*Upper 95%*
γ_0	509.765	15.265	33.395	3.23E-63	479.546	539.983
$\gamma_1(\overline{G})$	29.860	1.550	19.268	8.11E-39	26.792	32.927
$\gamma_2(\overline{V})$	-9.901	0.856	-11.572	2.35E-21	-11.595	-8.208
$\gamma_3(\overline{V}^2)$	0.089	0.011	8.040	6.58E-13	0.067	0.111

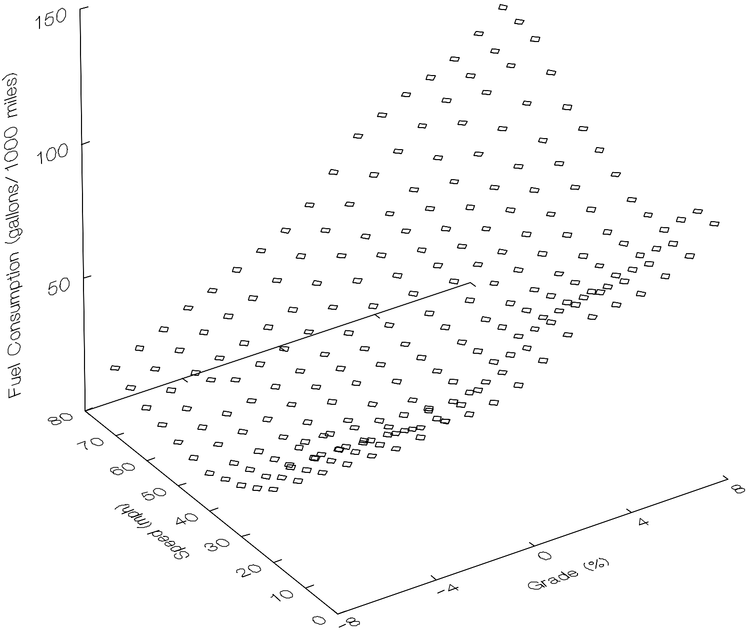


Figure 4.16: Fuel consumption data plots: medium car.

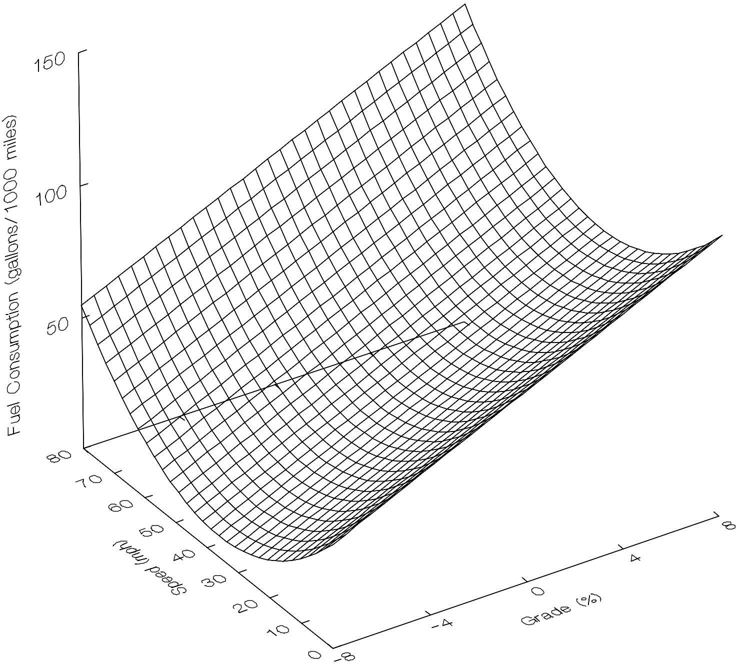


Figure 4.17: Regression function of fuel consumption: medium car.

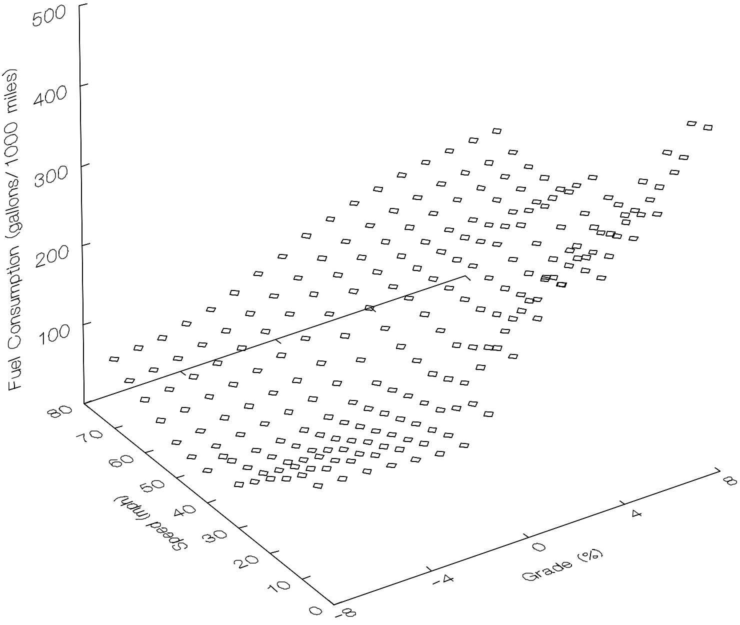


Figure 4.18: Fuel consumption data plots: 2A-SU truck.

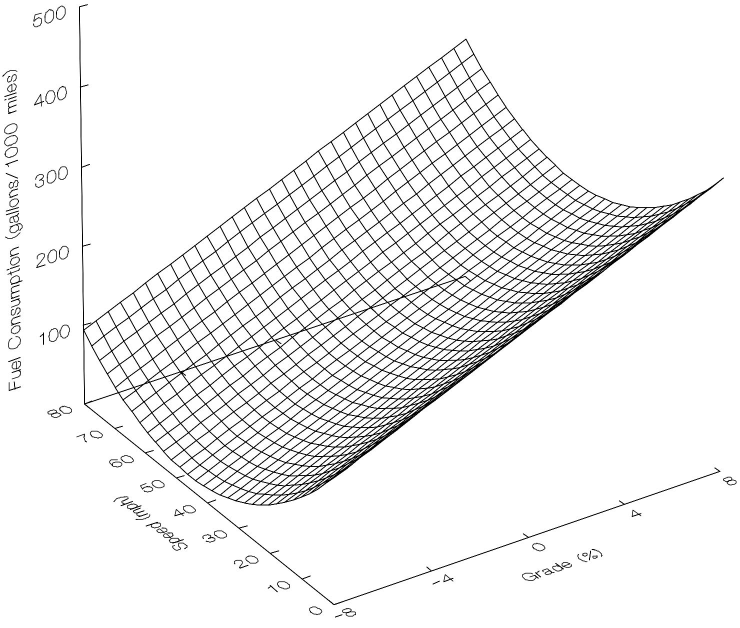


Figure 4.19: Regression function of fuel consumption: 2A-SU truck.

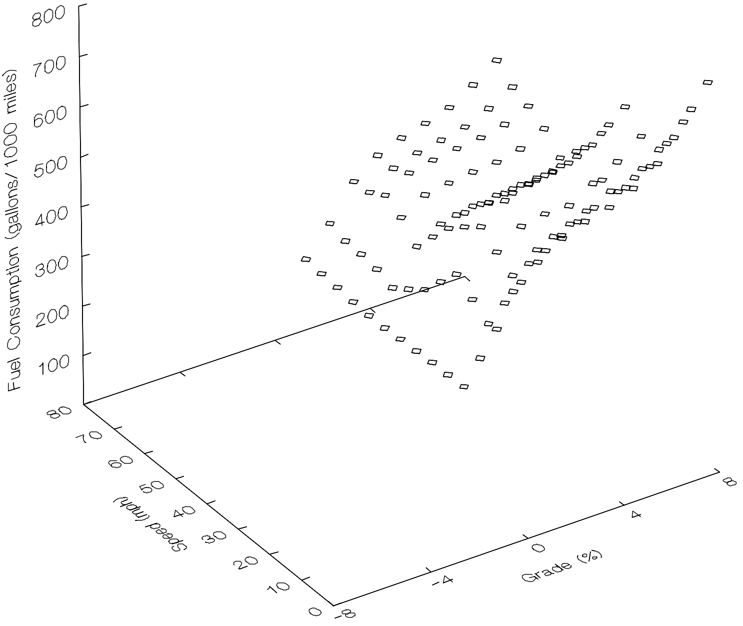


Figure 4.20: Fuel consumption data plots: 3-S2 diesel truck.

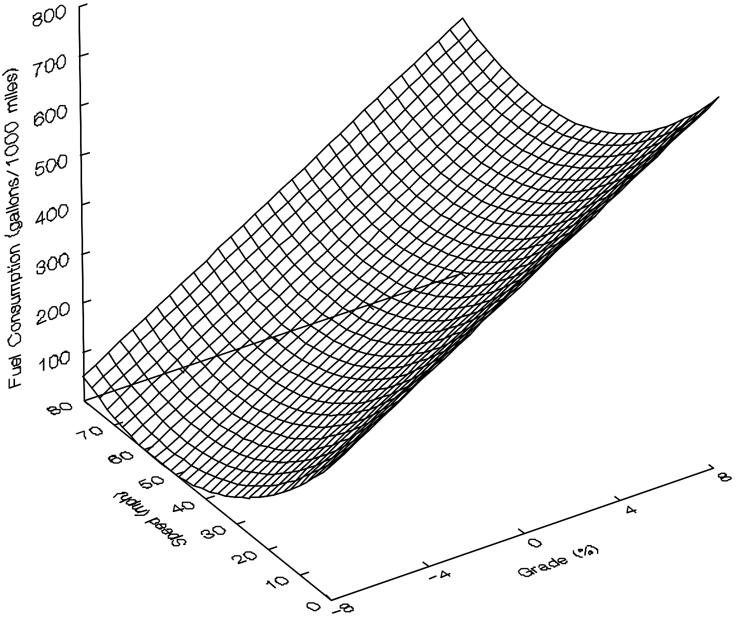


Figure 4.21: Regression function of fuel consumption: 3-S2 diesel truck.

For a macroscopic model it is undesirable to analyze the fuel consumption for each road section and at different constant speeds. Accordingly, average road grade and average running speed will be used for $\overline{G}$ and $\overline{V}$. It can be easily verified that the average road grade over different road sections for a given alignment is equal to $(z_E - z_S)/L_n$, where z_S and z_E are the elevations of the start (i.e., S) and end (i.e., E) points; L_n is the total length of the alignment, given in eqn (4.31). $\overline{V}$ can be obtained with eqns (4.46) to (4.48), depending on what time period and which traffic direction are analyzed. Note that the functional form in eqns (4.51) to (4.53) is slightly different from that in eqn (4.49). However, the data show that the structure of eqns (4.51) to (4.53) has the best statistical performance. That is probably because road grade is also treated as an independent variable in the regression function.

Let U_g and U_d be the unit prices for gasoline and diesel fuel, respectively ($/gallon). Then from eqns (4.51) to (4.53), we can obtain the fuel consumption costs for each type of vehicle. For easy notation, a vector representation is employed.

$$\mathbf{F}(\overline{G},\overline{V}) = \begin{bmatrix} U_g F_{MC}(\overline{G},\overline{V}) \\ U_g F_{2A}(\overline{G},\overline{V}) \\ U_d F_{3S}(\overline{G},\overline{V}) \end{bmatrix}, \tag{4.54}$$

where $F_{MC}(\overline{G},\overline{V})$, $F_{2A}(\overline{G},\overline{V})$, and $F_{3S}(\overline{G},\overline{V})$ are given in eqns (4.51) to (4.53).

In highway project evaluation, the traffic composition is usually indicated by a T factor for the proportion of heavy vehicles. Since we use 2A single-unit truck and 3-S2 diesel truck to represent the whole population of heavy vehicles, the relative percentage of 2A and 3-S3 trucks must be estimated. This can be done by using nationwide or statewide data. Let P_{2A} and P_{3S} denote the percentages of 2A and 3-S2 trucks in the heavy truck stream, respectively, where $P_{2A} + P_{3S} = 1$. Then the traffic composition vector $\mathbf{T}$ is as follows:

$$\mathbf{T} = \begin{bmatrix} (1-T) \\ P_{2A}T \\ P_{3S}T \end{bmatrix}. \tag{4.55}$$

To estimate total fuel consumption cost for an alignment, the traffic must be analyzed separately in each direction. If the road grade is $\overline{G}$ in one direction, then for the opposite direction, the road grade would be $-\overline{G}$. Assume that for each day the morning and afternoon peaks have the same duration $H_p/2$, where H_p is the number of peak hours per day. Then, taking different vehicle classes and time periods into account, the fuel consumption cost for one direction traffic in the base year can be obtained with the following equation:

$$\begin{aligned} c_F^B\left(\overline{G},L_n,\overline{V}_{pp},\overline{V}_{pn},\overline{V}_o\right) &= Q_{pp}(154.5H_p)\frac{L_n}{5280\times 1000}\left[\mathbf{T}\cdot\mathbf{F}(\overline{G},\overline{V}_{pp})\right] \\ &+Q_{pn}(154.5H_p)\frac{L_n}{5280\times 1000}\left[\mathbf{T}\cdot\mathbf{F}(\overline{G},\overline{V}_{pn})\right] \\ &+Q_o(6570-309H_p)\frac{L_n}{5280\times 1000}\left[\mathbf{T}\cdot\mathbf{F}(\overline{G},\overline{V}_o)\right], \end{aligned} \tag{4.56}$$

where c_F^B = fuel consumption cost for traffic in one direction in the base year ($/year)
Q_{pp} = peak hourly traffic in the prevalent direction (vph), given in eqn (4.43)
Q_{pn} = peak hourly traffic in the non-prevalent direction (vph), given in eqn (4.44)
Q_o = one-way hourly traffic in the off-peak period (vph), given in eqn (4.45)
L_n = the total length of the alignment (ft), obtained with eqn (4.31).

Note that in eqn (4.56), the first and second terms denote the fuel consumption costs for prevalent and non-prevalent directions in the peak period. The third term represents the fuel consumption cost in the off-peak period.

Similarly to eqn (4.56), the fuel consumption cost for the traffic in the opposite direction is derived by taking the road grade equal to $-\overline{G}$:

$$\begin{aligned} c_F^B\left(-\overline{G},L_n,\overline{V}_{pp},\overline{V}_{pn},\overline{V}_o\right) &= Q_{pp}(154.5H_p)\frac{L_n}{5280\times 1000}\left[\mathbf{T}\cdot\mathbf{F}(-\overline{G},\overline{V}_{pp})\right] \\ &+Q_{pn}(154.5H_p)\frac{L_n}{5280\times 1000}\left[\mathbf{T}\cdot\mathbf{F}(-\overline{G},\overline{V}_{pn})\right] \\ &+Q_o(6570-309H_p)\frac{L_n}{5280\times 1000}\left[\mathbf{T}\cdot\mathbf{F}(-\overline{G},\overline{V}_o)\right]. \end{aligned} \tag{4.57}$$

Again, we assume that the annual traffic will increase at a constant rate r_t. Then for the entire analysis period n_y, the net present value of fuel consumption cost will be

$$c_F=\left[c_F^B(\overline{G},L_n,\overline{V}_{pp},\overline{V}_{pn},\overline{V}_o)+c_F^B(-\overline{G},L_n,\overline{V}_{pp},\overline{V}_{pn},\overline{V}_o)\right]\times\left[\frac{e^{(r_t-\rho)n_y}-1}{r_t-\rho}\right], \tag{4.58}$$

where c_F = the net present value of total fuel consumption cost ($)
ρ = assumed interest rate (decimal fraction).

The above analysis implies that the geometric design of an alignment affects the total fuel consumption cost in two ways. First, geometric design influences the average running speed through eqns (4.46) to (4.48), which further affects fuel consumption via eqns (4.51) to (4.53). Next, the fuel consumption model itself is a function of road grade and road length, which are the design features of the alignment. The relations allow us to assess the fuel consumption costs for various designs of an alignment.

4.6.2 Travel time costs

AASHTO (1977) indicates that the travel time costs per hour for passenger cars, and for 2A and 3-S2 trucks are \$3, \$7, and \$8 respectively. The values were estimated based on the Consumer Price Index (CPI) in 1975. Thus, they must be updated to current price level. According to 1994 CPI values provided in Wright (1996), we update the unit values of travel time to about \$8.5, \$20, and \$23. Let $\mathbf{v}$ be the vector of unit time values defined as

$$\mathbf{v} = \begin{bmatrix} v_{MC} \\ v_{2A} \\ v_{3S} \end{bmatrix} = \begin{bmatrix} 8.5 \\ 20 \\ 23 \end{bmatrix}, \qquad (4.59)$$

where v_{MC}, v_{2A}, and v_{3S} are unit values of travel time for medium car, 2A trucks and 3-S2 trucks, respectively. Then considering different time periods, the total travel time costs for two-way traffic in the base year can be derived as

$$\begin{aligned} c_T^B\left(L_n, \overline{V}_{pp}, \overline{V}_{pn}, \overline{V}_o\right) &= Q_{pp}(309H_p)\frac{\left(L_n/5280\right)}{\overline{V}_{pp}}(\mathbf{T}\cdot\mathbf{v}) \\ &+ Q_{pn}(309H_p)\frac{\left(L_n/5280\right)}{\overline{V}_{pn}}(\mathbf{T}\cdot\mathbf{v}) \\ &+ Q_o\left[2(6570-309H_p)\right]\frac{\left(L_n/5280\right)}{\overline{V}_o}(\mathbf{T}\cdot\mathbf{v}), \end{aligned} \qquad (4.60)$$

where Q_{pp}, Q_{pn}, and Q_o are hourly volumes (vph) in the peak prevalent direction, peak non-prevalent direction, and either direction in off-peak periods, as given in eqns (4.43), (4.44), and (4.45)

L_n = the total length of the alignment (ft), obtained with eqn (4.31),

$\mathbf{T}$ = traffic composition vector defined in eqn (4.55).

Following the same assumption on traffic demand, then the net present value of total travel time costs for the entire analysis period n_y is

$$c_T = c_B^T\left(L_n, \overline{V}_{pp}, \overline{V}_{pn}, \overline{V}_o\right) \times \left[\frac{e^{(r_t-\rho)n_y} - 1}{r_t - \rho}\right], \qquad (4.61)$$

where c_T = the net present value of total travel time costs ($)
r_t = the annual growth rate of $AADT$ (decimal fraction)
ρ = assumed interest rate (decimal fraction).

The above analysis relates the travel time cost to geometric designs by two approaches. First, the travel time cost is a function of the total length of the alignment. Secondly, the travel time cost depends on the average running speed, which is a function of geometric features of an alignment as shown in eqns (4.46) to (4.48).

4.6.3 Accident costs

The estimation of accident cost is relatively difficult due to the complexity of accident causes and the lack of empirical data for a non-existent highway. Note that accidents may happen anywhere along the alignment. However, it is likely that accident costs and rates are higher on sharper curves due to changes in speed and inconsistency in geometric design standards. Therefore, the analysis will focus on curves. Ben-Akiva *et al* (1985) indicated that when the available sight distance is inadequate, the costs of accidents due to speed changes increase more than linearly with increasing discrepancies between braking and sight distances. Let $R_{\min}$ be minimum radius for a given design speed. Then the accident costs on a curve may be defined in the following form:

$$c_c(i) = \begin{cases} b(R_i - R_{\min})^2 & \forall R_i < R_{\min} \\ 0 & \forall R_i \geq R_{\min} \end{cases}, \qquad (4.62)$$

where $c_c(i)$ = accident costs on curve i
R_i = radius of curve i
b = coefficient.

Note that our model does not yet consider the curvature constraint. A function form like eqn (4.62) might be used to reflect the accident costs on curves and eventually force the alignment to satisfy the maximum curvature (or minimum radius) constraints anywhere along the alignment. From this point of view, eqn (4.62) can be also regarded as the penalty cost for violating the minimum radius constraint at the i^{th} curve. The magnitude of coefficient b determines the weight of the penalty. If b is extremely large, then the penalty cost will be very high when the minimum radius constraint is violated. Accordingly, it is very likely that the resulting alignment will satisfy the constraint. On the other hand, if b is small, then it is possible that the final alignment may slightly violate the minimum radius constraints. In such a case, a speed limit must be enforced.

Eqn (4.62) is very easy to use and plays an important role in penalizing the violations of minimum radius constrains. However, it cannot reflect the accident costs under different traffic situations and geometric designs, even the minimum radius constraints are satisfied. A more detailed estimation of accident cost on a curve can be obtained by employing the model developed by Zegeer *et al* (1992). The authors established a linear regression model for estimating the number of accidents on the curve of a two-lane rural highway in the following form:

$$A = (1.55 L_C Q_T + 0.14 D_C Q_T - 0.12 S_C Q_T)(0.978)^{W-30}, \tag{4.63}$$

where A = total number of accidents on a curve in a 5-year period
L_C = length of curve (miles)
D_C = degree of curvature (degrees)
Q_T = traffic volume in a 5-year period (in millions of vehicles) passing through the curve (both directions)
S_C = existence of spiral transition curve, where $S_C = 0$, if no spiral curve is used; $S_C = 1$, otherwise
W = width of roadway on the curve.

Note that the alignment generated by Algorithm 4.1 is only composed of tangent segments and circular curves. However, it does not mean that spiral curves would not be used. In fact, omitting spiral transition in the alignment generating procedure is intended to ease computation. In estimating accident cost on a curve, we assume that the spiral does exist between tangent segment and circular curve. Recall that an alignment may consists of n curves, where n is total number of intersection points as defined in section 4.2. Moreover, the analysis period n_y is supposed to be much longer than 5 years (usually 25 or 30 years). Assume that n_y is n_k times as much as 5 (i.e., $n_y = 5n_k$), and the traffic volume differs from year to year. Then the number of accidents must be estimated for each 5-year period, and thus totally n_k different traffic volumes will be analyzed. Taking all these into account, we prefer to rewrite the above equation as

$$A_{kj} = \left(1.55 L_C(j) Q_T(k) + 0.14 D_C(j) Q_T(k) - 0.12 Q_T(k)\right)(0.978)^{W-30}, \tag{4.64}$$

where A_{kj} = total number of accidents on the j^{th} curve in the k^{th} 5-year period
$L_C(j)$ = length of the j^{th} curve (miles)
$D_C(j)$ = degree of curvature at the j^{th} curve (degrees)
$Q_T(k)$ = traffic volume in the k^{th} 5-year period (in millions of vehicles).

To apply the above equation, we must determine the appropriate values of its predictors (independent variables) based on the geometric designs of the alignment and traffic demand. The length at the j^{th} curve can be computed with

$$L_C(j) = \frac{R_j \Delta_j}{5280}, \tag{4.65}$$

where R_j = the radius of the j^{th} curve (ft), which is obtained with Algorithm 4.1
Δ_i = the intersection angle at the j^{th} curve (radians), obtained with eqn (4.10).

The degree of curvature at the j^{th} curve is

$$D_C(j) = \frac{5730}{R_j}\,. \tag{4.66}$$

The calculation of traffic volume is relatively complex. Assume annual traffic will increase at a constant rate r_t. Then it can be verified that in the k^{th} 5-year period, the total traffic volume is

$$\begin{aligned} Q_T(k) &= \frac{365\times AADT}{10000000}\sum_{i=5(k-1)+1}^{5k}(1+r_t)^i \\ &= 3.65\times10^{-5}\times AADT\times\left[\frac{(1+r_t)^6-(1+r_t)}{r_t}\right](1+r_t)^{5(k-1)} \end{aligned} \tag{4.67}$$

where $AADT$ = Annual Average Daily Traffic (two-way) (vehicles/day).

Assume that annual accidents are equal and concentrated distributed at the end of each year in a 5-year period. Then for the entire analysis period, the net present value of accident cost is

$$c_c = \frac{1}{5}\left[\frac{(1+\rho)^5-1}{\rho(1+\rho)^5}\right]\sum_{k=1}^{n_k}\left[\frac{1}{(1+\rho)^{5(k-1)}}\sum_{j=1}^{n}A_{kj}\right]\cdot U_c\,, \tag{4.68}$$

where c_c = accident costs on all curves of the alignment ($)
ρ = assumed interest rate (decimal fraction)
n_k = number of 5-year periods
n = number of curves (intersection points) along the alignment
A_{kj} = total number of accidents on the j^{th} curve in the k^{th} 5-year period, given in eqn (4.64)
U_c = cost per accident ($).

The cost per accident U_c in the above equation can be obtained by averaging the costs associated with different types of accidents. Usually accident costs are classified as death (fatal accidents), injury, and property damage. For a newly built highway, nationwide or statewide data can be used. Note that if the traffic remains unchanged throughout the entire analysis period, then A_{kj} will be equal to A_{1j} for all k, and thus the above equation can be reduced to

$$c_c = \left[\frac{1}{5}\sum_{j=1}^{n} A_{1j} U_c\right]\left[\frac{(1+\rho)^{n_y}-1}{\rho(1+\rho)^{n_y}}\right], \tag{4.69}$$

which is equivalent to the net present value of a series of uniform annual end-of-period costs

$$\frac{1}{5}\sum_{j=1}^{n} A_{1j} U_c . \tag{4.70}$$

It should be noted that eqn (4.68) does not yield a significant difference between the costs for $R_i < R_{\min}$ versus $R_i \geq R_{\min}$, and accordingly has poor control over the minimum radius constraints. A solution to this problem is to combine eqns (4.62) and (4.68) so that a penalty cost is incurred for a curve whose radius is less than the minimum radius, i.e.,

$$c_c = \sum_{i=1}^{n} c_c(i) + \frac{1}{5}\left[\frac{(1+\rho)^5 - 1}{\rho(1+\rho)^5}\right]\sum_{k=1}^{n_k}\left[\frac{1}{(1+\rho)^{5(k-1)}}\sum_{j=1}^{n} A_{kj}\right]\cdot U_c , \tag{4.71}$$

where $c_c(i)$ = penalty cost imposed on the i^{th} curve, given in eqn (4.62).

In eqn (4.71), the first tem represents the penalty costs for violating minimum radius constraints. The main purpose of the penalty term is to force the final alignment to satisfy the minimum radius constraints. The second term is used to reflect the accident costs for different geometric designs under different traffic conditions.

With the above analysis, we are now able to compute the user costs for a given alignment by summing up fuel consumption cost, travel time cost, and accident cost on curves:

$$C_U = c_F + c_T + c_c , \tag{4.72}$$

where C_U = user costs of a given highway ($)
c_F = fuel consumption cost ($), given in eqn (4.58)
c_T = travel time cost, given in eqn (4.61)
c_c = accident cost on curves ($), from eqns (4.68).

4.7 Final model and its properties

Recall that the decision variables in the proposed model are the coordinates d_i's along vertical cutting lines. The objective function is the summation of various cost items. Hence the final model can be formulated as follows:

Model 1 – Model for optimizing non-backtracking horizontal alignments

$$\underset{d_1, d_2, \ldots\ldots, d_n}{\text{Minimize}} \quad C_T = C_N + C_L + C_U \,, \tag{4.73}$$

$$\text{subject to} \quad d_{iL} \le d_i \le d_{iU} \text{, for all } i = 1, \ldots\ldots, n \,, \tag{4.74}$$

where the alignment is generated by Algorithm 4.1

C_T = total cost ($)
C_N = location-dependent cost ($), computed by Algorithms 4.2 and 4.3
C_L = length-dependent cost ($), given in eqn (4.35)
C_U = user cost ($), given in eqn (4.72)
d_{iL} and d_{iU} are lower and upper bound of the i^{th} decision variable.

The above model form differs from that formulated in Model 0. Model 0 is a rigorous definition of the problem for optimizing highway alignments, but is not a solvable form. On the other hand, Model 1 is a computable form because we have already shown the relations between different cost items and the decision variables d_i 's. For any given set of decision variables, we will be able to compute the objective value. Regardless of gradient and vertical curvature constraints, and of the ability to deal with a backtracking alignment, we want to show the equivalence of Model 1 to Model 0 because Model 0 provides a more rigorous mathematical definition of alignment optimization problems.

The objective function defined in eqn (3.10) includes various cost items. The area-dependent cost C_A and volume-dependent cost C_V appearing in eqn (3.10) have been incorporated into length-dependent cost C_L and location-dependent cost C_N in eqn (4.73) respectively. Regardless of spiral transition curves, the alignment generated by Algorithms 4.1 will definitely satisfy the boundary conditions (eqns (3.11) and (3.12)), and alignment necessary conditions (eqns (3.13) and (3.14)). Moreover, both the continuity condition (defined in eqn (3.2)) and the first continuously differentiable condition (defined in eqn (3.3)) are also satisfied. The horizontal curvature constraint defined in eqn (3.15) is combined into user cost by introducing a big penalty into accident cost, and thus will be satisfied if the solution algorithm successfully locates the optimal alignment. Finally, the inaccessibility constraint (eqn (3.18)) is incorporated into the location-dependent cost and the alignment is bounded by eqn (4.74). Therefore, it will also be satisfied.

Note that the objective function C_T in eqn (4.73) is the summation of different cost components and thus, possible tradeoffs among various cost items may be made. For example, the increase in accident cost for a sharper curve may be compensated by the savings in location-dependent cost. The optimized alignment should have the lowest total cost by making the best tradeoffs among different cost components.

Next, we want to examine whether the proposed model satisfies the necessary conditions of a good model for optimizing highway alignment as described in section 2.7. The results are summarized in Table 4.5.

Table 4.5: Checklist of necessary conditions for a good alignment optimization. model

No.	Description of Conditions	Check Box
(1)	Consider all dominating and sensitive costs	yes
(2)	Formulate all important constraints	yes
(3)	Yield a realistic alignment	yes
(4)	Be able to handle alignments with backward bends	no
(5)	Simultaneously optimize 3-dimensional alignments	no
(6)	Find globally or near globally optimal solution	not investigated yet
(7)	Have an efficient solution algorithm	not investigated yet
(8)	Have low storage requirements	possibly
(9)	Have a continuous search space	yes
(10)	Automatically avoid inaccessible regions	yes
(11)	Be compatible with GIS	yes

The above table shows that conditions (4) and (5) are not satisfied in the proposed model. We are not yet able to investigate conditions (6) and (7) at this point because the solution algorithm has not been presented. For condition (8), the proposed model employs an efficient mechanism compatible with GIS data format for storing the information of the study region at a desired precision level. However, the memory requirement also depends on the solution algorithm.

The main task remaining in developing the proposed model is to design an efficient solution algorithm. Note that the search space is an n-dimensional hyperspace rather than a 2-dimensioanl plane because the number of decision variables is n even though we are only optimizing a 2-dimensional alignment. By carefully investigating the detailed structure of the model, we find that the objective function is an implicit function of the decision variables set. The computation of each cost component requires the coordinates of points of curvature, points of tangency, and points of intersection, as well as the intersection angle and radius at each circular curve. However, the information is not available until the decision variable set is given and Algorithms 4.1 has been applied to generate the alignment. Consequently, the problem turns out to be an implicit, constrained optimization model with a non-differentiable objective function, which is probably very noisy and has many local optima.

Due to the properties of the model, no gradient-based search algorithms are applicable to the problem. As a result, only direct search methods can be used. This seems to be an unpromising approach to the problem because most existing direct search methods are unable to locate the global optimal solution unless a very exhaustive search is employed. Then how shall we solve the problem? Is there any method other than the conventional search algorithms with the potential to find the global optimal solution or at least a relatively good local optimum? Fortunately, with the development of artificial intelligence (AI) methods, it becomes possible to efficiently search for a relatively good solution without any gradient information. The proposed approach is known as Genetic Algorithms. In Appendix A, a brief introduction to the method is provided. A comparison of this method to other optimization techniques is also presented.

Chapter 5

Solution algorithms for optimizing non-backtracking horizontal alignments

In chapter 4, we show that the proposed Model 1 is a non-convex, constrained optimization problem, where the objective is an implicit function of decision variables. Therefore, only direct search methods can be applied in solving the problem. From experience, we also know that the objective function is very noisy, i.e., non-smooth with many peaks. In Appendix A, which provides an overview of genetic algorithms and defines their terminology, we compare several optimization techniques. Among those methods not requiring gradient information, GAs may have the potential to efficiently find a relatively good solution to problems. However, we also realize that the applications of GAs require problem-specific genetic encoding and operators, which will definitely affect the performance of a GA in solving the problem. Therefore, the purpose of this chapter is to develop a solution algorithm evolved from GAs for optimizing non-backtracking horizontal alignment.

Note that the solution algorithm is probabilistic and involves the process of generating random numbers. For notational convenience, we denote $r_c[b_L, b_U]$ ($r_d[b_L, b_U]$) as a random number generated from a continuous (discrete) uniform distribution whose domain is within the interval $[b_L, b_U]$. Such notation will be employed repeatedly throughout this book.

5.1 Genetic encoding

For notational convenience, in this book we refer to a chromosome as Λ and individual gene as λ subscripted by its location. For example, a seven-gene chromosome may be represented by $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_7]$. For Model 1 developed in chapter 4, instead of using binary digits to represent a solution, a floating point encoding scheme is employed. The chromosome can be easily defined as the set of decision variables, each of which is a continuous real number confined within its associated boundaries. That is

$$\Lambda = [\lambda_1, \lambda_2, ..., \lambda_n] = [d_1, d_2, ..., d_n], \tag{5.1}$$

where d_i = the coordinate along the i^{th} vertical cutting line as defined in section 4.2

n = total number of vertical cuts (or intersection points).

In the above equation, the alleles of the i^{th} gene will be bounded within the interval $[d_{iL}, d_{iU}]$, where d_{iL} and d_{iU} are the corresponding lower and upper bounds defined in eqns (4.4) to (4.7). The boundaries play an important role in generating the initial population and developing genetic operators, which will be discussed later in this chapter.

5.2 Initial population

In order to keep the gene pool as large as possible so that the entire search space can be explored, the initial population should be randomly generated. For optimizing highway alignment, however, it is suggested that certain chromosomes that represent a straight alignment to the problem be included in the initial population. If somehow a model user has some initial guesses about the solutions, they might be included as well. Without loss of generality, we assume that no prior knowledge about the solution is available at this moment. Then the population can be generated as follows:

(1) Intersection points lie on the straight line connecting the start and end points

In this case, the set of intersection points forms a straight alignment. This may not be a good solution, but to some extent, it may contain some useful information about the problem. In fact, except for location-dependent costs, all other cost items may be reduced by straighter and shorter alignments. Recall that in this case, the intersection points lie exactly at the origins of their associated vertical cuts. The chromosome is thus represented by

$$\Lambda = [\lambda_1, \lambda_2, ..., \lambda_n] = [0,0,..........,0]. \tag{5.2}$$

(2) Intersection points lie randomly on the vertical cuts

In this case, each gene of the chromosomes is randomly generated from a continuous uniform distribution within the corresponding boundaries. That is

$$\lambda_i = r_c[d_{iL}, d_{iU}], \quad \forall i = 1, ..., n . \tag{5.3}$$

5.3 Fitness function

For Model 1, the fitness function can be easily defined as the objective function C_T as defined in eqn (4.72). Recall that given a chromosome, the computation of the fitness function involves a complicated procedure. First, the corresponding intersection points of the chromosome must be decoded by eqn (4.8). Next, given the set of intersection points, we generate the corresponding alignment with Algorithm 4.1. Then based on the resulting alignment, we can compute its associated cost items, which are discussed in detail in chapter 4.

5.4 Selection/replacement

In our proposed solution algorithm, the selection/replacement (or referred as sampling mechanism) procedure has several important aspects. It is characterized as an ordinal-based selection, regular sampling space, generational replacement, and elitism model. The scheme is modified from the two-step selection algorithm introduced by Michalewicz (1996). The structure of the procedure is outlined as follows:

Algorithm 5.1 Selection/replacement procedure

(1) Step 1: Select n_r (not necessarily distinct) parents from the current population (population size is n_p) to reproduce offspring. Each of the selected chromosomes is marked as applicable to exactly one genetic operation. Let n_r parent chromosomes breed to produce exactly n_r offspring.

(2) Step 2: Select n_r distinct chromosomes from the current population to die. Each of the selected chromosomes will be replaced by new offspring at the next generation.

(3) Step 3: Clone the rest $n_p - n_r$ chromosomes to the next generation.

(4) Step 4: Insert these n_r new offspring into the next generation.

Note that the selected n_r parents for reproducing offspring are not necessarily distinct. A better chromosome may be selected several times. The selection is ordinal-based. In the proposed algorithm, the probability that an individual is selected for reproducing offspring is specified as a nonlinear function of a user-defined parameter q:

$$p_k = cq(1-q)^{k-1}, \tag{5.4}$$

where p_k = the selection probability for the k^{th} chromosome in the ranking of the population

c = coefficient

q = user defined parameter, $q \in (0,1)$.

The above function is proposed by Michalewicz (1996). Note that the summation of all probabilities must be equal to 1. That is

$$\begin{aligned}\sum_{k=1}^{n_p} p_k &= c[q+q(1-q)+q(1-q)^2+\cdots\cdots\cdots+q(1-q)^{n_p-1}] \\ &= c[1-(1-q)^{n_p}]=1\end{aligned} . \tag{5.5}$$

The above equation implies

$$c=\frac{1}{1-(1-q)^{n_p}}, \tag{5.6}$$

and thus

$$p_k=\frac{q(1-q)^{k-1}}{1-(1-q)^{n_p}}. \tag{5.7}$$

There is an alternative approach to derive eqn (5.7). Suppose that we conduct a sequence of independent Bernoulli trials in which q denotes the probability that a chromosome is selected in a single trial. Assume that the trial is started from the first to the last chromosome in the ranking of the population. Then the probability that the k^{th} chromosome is selected should be $q(1-q)^{k-1}$. If after n_p trials, none of the chromosomes in the population is selected, then the selection process is repeated until we successfully select a chromosome. This procedure is somewhat like a geometric distribution with finite states (i.e., n_p) of the random variable. Note that the probability that none of the chromosomes is selected in the first cycle is

$$1-[q+q(1-q)+q(1-q)^2+\cdots\cdots\cdots+q(1-q)^{n_p-1}]=(1-q)^{n_p}. \tag{5.8}$$

Therefore, the overall probability that the k^{th} chromosome is selected will be

$$q(1-q)^{k-1}\sum_{i=0}^{\infty}[(1-q)^{n_p}]^i=\frac{q(1-q)^{k-1}}{1-(1-q)^{n_p}}. \tag{5.9}$$

The above relation is identical to eqn (5.7). It shows that a better chromosome has a higher probability of being selected. Note that the user-defined parameter q is the indication of selective pressure. Larger values of q imply stronger selective pressure, which may result in premature convergence. On the other hand, with lower values of q, GAs tend to converge slowly but may explore the search space more thoroughly.

The sampling process is now performed by generating a random number $r_c[0,1]$. If r_c falls in between the cumulative probabilities of the $(k-1)^{th}$ and k^{th} chromosomes, then the k^{th} chromosome is selected to reproduce offspring.

The procedure for selecting chromosomes to die is similar. The sampling process is performed on a regular sampling space, and eqn (5.7) is used again. However, for replacement, a worse chromosome must have a higher selection probability than a better chromosome. Therefore, if we generate a random number, which indicates the k^{th} chromosome is selected, then the actual chromosome selected to die is the $(n_p - k + 1)^{th}$ chromosome. Also, if k is equal to n_p, then we must regenerate a random number because the best chromosome is never selected to die. This elitism model guarantees the search is non-deteriorating. That means that after the GA starts to run, the best chromosome in the population will evolve to become more adapted to the environment of the problem, or at least will remain no worse than in the previous generation.

5.5 Genetic operators

It is important to mention that the relation between the genes in a chromosome and the corresponding alignment is indirect. Each gene in the chromosome is an intersection point in a 2 dimensional space, where the alignment is obtained by fitting circular curves at each intersection point. Moreover, the genes are not independent of each other because whenever the location of an intersection point is changed, the configuration of the alignment at other intersection points may change as well due to the curve fitting process.

The above properties make it more difficult to develop appropriate genetic operators. Moreover, the genes in a chromosome are encoded as floating-point (i.e., real) numbers. For these reasons, conventional operators do not work well in this situation. To facilitate the search, we must devise problem-specific genetic operators. Eight kinds of operators are proposed in the proposed solution algorithm. We will discuss them in turn.

5.5.1 Uniform mutation

For most applications of GAs, where the genes are encoded as real numbers, the uniform mutation is performed by randomly selecting a gene and replacing its value with a randomly selected real number. Let the chromosome to be mutated be $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_n]$ and the k^{th} gene be selected to apply the operator (i.e., $k = r_d[1, n]$). Then λ_k will be replaced by $\lambda'_k = r_c[d_{kL}, d_{kU}]$.

Recall that mutation operators are supposed to help in exploring the entire search space and are often seen as background operators to maintain genetic diversity in the population. Unfortunately, the simple mutation operator does not work as expected in our proposed model. To illustrate this, let us consider the situation shown in Figure 5.1.

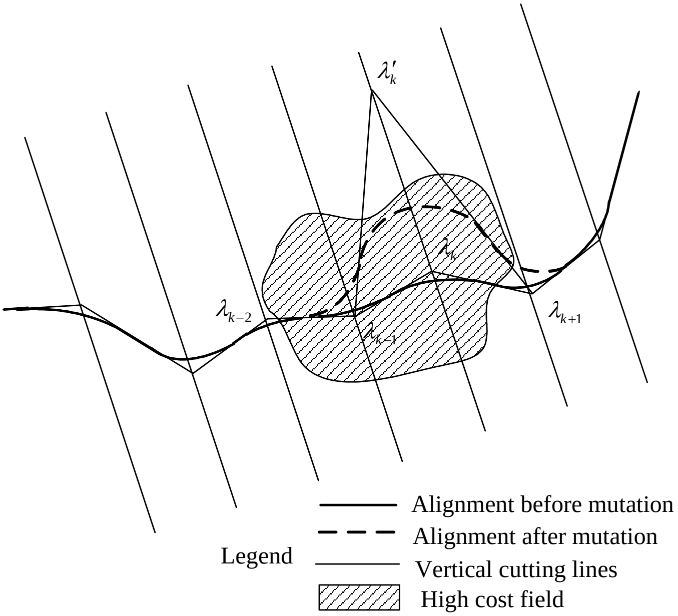


Figure 5.1: An example of a failure mutation.

In the above figure, the corresponding alignment of the chromosome passes through a high cost field. Suppose that λ_k is selected to mutate and the resulting new allele of the gene is λ'_k. The figure shows that the new alignment is even worse than the original one. We may expect next time λ_{k-2}, λ_{k-1}, and λ_{k+1} will be selected to undergo mutation operators so that the corresponding alignment can jump out of the high cost field. However, before these adjacent genes are selected to mutate, the chromosome might die off because the new chromosome is worse and more likely to be selected to die. In fact, as the GA evolves, the final solution will be the one passing through the high cost field at minimal distance. Obviously it is not the global optimum, but just a local optimum.

A solution to this problem is to design the operator in such a way that the corresponding alignment of the chromosome has the potential to jump out of the high cost field at once. The operator is then modified by generating another two independent loci *i* and *j* in addition to *k*, where $i = r_d[0, k-1]$ and $j = r_d[k+1, n+1]$, such that the corresponding intersection points of λ_i and λ_k are connected by straight-line segments, and so are the corresponding points of λ_k and λ_j. Note that here locus = 0 represents the starting point *S* of the alignment, and locus = $n+1$ denotes the end point *E*. We will name this procedure "elimination" because it is analogous to eliminating the curves between the i^{th} and k^{th} intersection points as well as those between the k^{th} and j^{th} intersection points. We now summarize the elimination procedure as follows:

Algorithm 5.2 Curve elimination procedure for Model 1

(1) Step 1: Encode S and E into the chromosome
Expand the chromosome $\Lambda = [\lambda_1, ..., \lambda'_k ....., \lambda_n]$ to $[0, \lambda_1, ..., \lambda'_k, ..., \lambda_n, 0]$.

(2) Step 2: Generate two independent random loci i and j

$$i = r_d[0, k-1], \text{ and } \quad j = r_d[k+1, n+1]$$

(3) Step 3: Change the values of the intermediate genes between the i^{th} and k^{th} genes

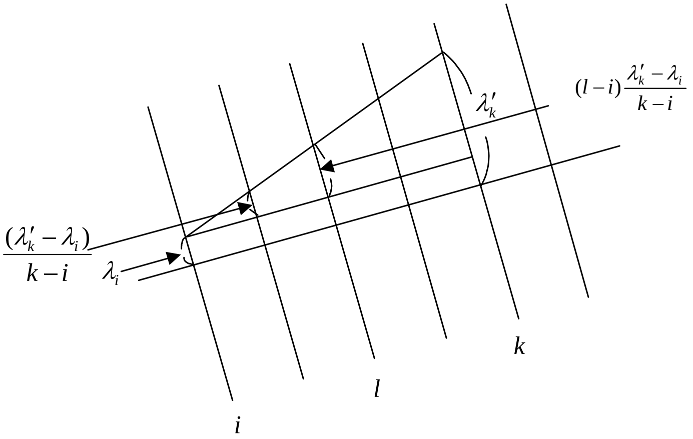
$$\lambda'_l = \lambda_i + (l-i)\frac{(\lambda'_k - \lambda_i)}{k-i}\text{, for all } l = i+1, ..., k-1 \tag{5.10}$$

(4) Step 4: Change the values of the intermediate genes between the k^{th} and j^{th} genes

$$\lambda'_l = \lambda_k + (l-k)\frac{(\lambda_j - \lambda'_k)}{j-k}\text{, for all } l = k+1, ..., j-1 \tag{5.11}$$

(5) Step 5: Remove the encoded genes of S and E from the chromosome
Truncate the chromosome $\Lambda' = [0, \lambda_1, ..., \lambda_i, \lambda'_{i+1} ....., \lambda'_k, ..., \lambda'_{j-1}, \lambda_j, ..., \lambda_n, 0]$ to $[\lambda_1, ..., \lambda_i, \lambda'_{i+1} ....., \lambda'_k, ..., \lambda'_{j-1}, \lambda_j, ..., \lambda_n]$.

Recall that the vertical cutting lines are parallel to each other. Hence, we can derive eqns (5.10) and (5.11) by elementary trigonometry. The following figure illustrates the concept of eqn (5.10):



$(l-i)\frac{\lambda'_k - \lambda_i}{k-i}$
λ'_k
$\frac{(\lambda'_k - \lambda_i)}{k-i}$
λ_i
k
l
i

Figure 5.2: Determination of the values of intermediate genes in the curve elimination procedure.

The above curve elimination procedure endows mutation operators with the capabilities of jumping away from the high cost field at once. The following figure is a successful example for the same case shown in Figure 5.1.

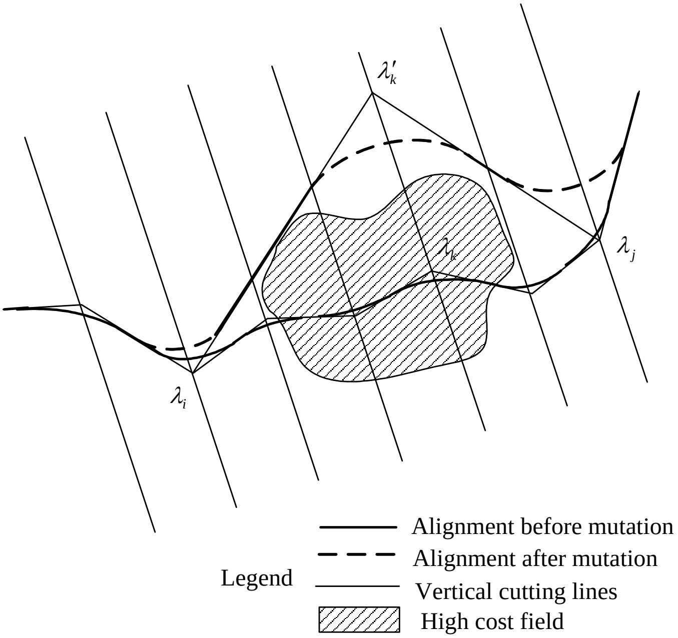


Figure 5.3: A successful example of the curve elimination procedure.

5.5.2 Straight mutation

This operator is proposed to straighten the alignment between two randomly selected intersection points. The idea behind this operator is that a straight alignment may result in lower total cost because most cost components are minimized by a more direct and shorter alignment. Let the chromosome to be mutated be $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_n]$. We randomly generate two independent loci i and j, where $i = r_d[0, n+1]$, $j = r_d[0, n+1]$, $i \neq j$, and $i < j$. Then the values of the intermediate genes between the i^{th} and j^{th} genes will be replaced by the following equation (the derivation is the same as for uniform mutation):

$$\lambda_l' = \lambda_i + (l-i)\frac{(\lambda_j - \lambda_i)}{j-i}, \text{ for all } l = i+1, ..., j-1. \tag{5.12}$$

In eqn (5.12), if $i = 0$ (which represents the start point S), then we set $\lambda_i = 0$. Similarly if $j = n+1$ (which denotes the end point E), then λ_j is also set to 0. The example in Figure 5.4 illustrates the straight mutation operator.

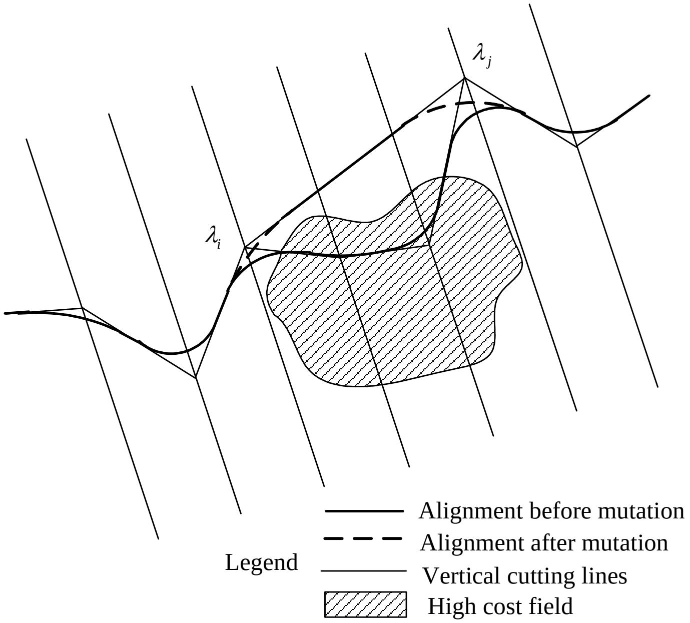


Figure 5.4: An example of a straight mutation operator.

5.5.3 Non-uniform mutation

Non-uniform mutation was introduced by Michalewicz (1996). Sometimes it is called dynamic mutation (Gen and Cheng, 1997) because the mutation range decreases through successive generations. The operator is designed for fine-tuning the solution. At early generations the mutation range is relatively large, while at later generations, the mutation is limited to a small range for local refinement. For a given parent $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_n]$, in which the k^{th} gene ($k = r_d[1, n]$) is selected for mutation. We first generate a binary random digit $r_d[0,1]$. Then λ_k will be replaced according the following rules:

$$\begin{aligned} &\text{If } r_d[0,1] = 0 \text{, then } \lambda'_k = \lambda_k - f(t, \lambda_k - d_{kL}) . \\ &\text{If } r_d[0,1] = 1 \text{, then } \lambda'_k = \lambda_k + f(t, d_{kU} - \lambda_k) . \end{aligned} \qquad (5.13)$$

In the above equations, t is the current generation number, while d_{kL} and d_{kU} are the corresponding lower and upper bounds of the k^{th} gene as defined in eqns (4.4) to (4.7).

The function $f(t, y)$ in the eqn (5.13) returns a random value in the range $[0, y]$ such that the probability of $f(t, y)$ approaching 0 increases as t increases. This property causes the operator to search the space uniformly at initial generations and very locally at later stages. The function $f(t, y)$ is

$$f(t,y) = y \times r_c[0,1] \times \left(1 - \frac{t}{n_T}\right)^{\xi}, \tag{5.14}$$

where n_T = the maxim number of generations
ξ = a user defined parameter which determines the degree of non-uniformity.

Note that in order to prevent the solutions from sticking at a local optimum, after a gene is mutated, the curve elimination procedure (Algorithm 5.2) is repeated.

5.5.4 Whole non-uniform mutation

This operator is introduced in Michalewicz’s GENOCOP system (GEnetic algorithm for Numerical Optimization for COnstrained Problem). It applies the non-uniform mutation operator to each of the genes in a given chromosome in a randomly generated sequence. The resulting offspring will be totally different from its parent. In other words, this operator forces a chromosome jump to another location of the search space to maintain the diversity of genes in the population. In the early stages of the evolution, the shift in the search space is significant. However, in the later generations, the operation perturbs all genes of a chromosome only around the vicinity of the corresponding solution for local refinement. The spirit of the operator is similar to some conventional optimization techniques, such as hill climbing methods, in which the step size is larger in earlier iterations and smaller in later ones.

5.5.5 Simple crossover

This operator is analogous to the one-point crossover in binary implementations of GAs. Let two parents $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots\ldots, \lambda_{in}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots\ldots, \lambda_{jn}]$ be crossed after a randomly generated position k, where $k = r_d[1,n]$. Then the resulting offspring are

$$\Lambda'_i = \left[\lambda_{i1}, \lambda_{i2}, \ldots\ldots, \lambda_{ik}, \lambda_{j(k+1)}, \ldots\ldots, \lambda_{jn}\right], \tag{5.15}$$

$$\Lambda'_j = \left[\lambda_{j1}, \lambda_{j2}, \ldots\ldots, \lambda_{jk}, \lambda_{i(k+1)}, \ldots\ldots, \lambda_{in}\right]. \tag{5.16}$$

Through this operator, it is expected that the new offspring inherit the good genes from their parents. Figure 5.5 is an example of simple crossover operator, where the first child receives all good information from its parents.

5.5.6 Two-point crossover

The idea of this operator is to exchange the genes between two randomly generated positions k and l for two given parents $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots\ldots, \lambda_{in}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots\ldots, \lambda_{jn}]$, where $k = r_d[1,n]$, $l = r_d[1,n]$, $k \neq l$, and $k < l$. The resulting offspring are

$$\Lambda'_i = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots, \lambda_{ik}, \lambda_{j(k+1)}, \ldots\ldots, \lambda_{jl}, \lambda_{i(l+1)}, \ldots\ldots\ldots, \lambda_{in}] \text{ and} \qquad (5.17)$$

$$\Lambda'_j = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots, \lambda_{jk}, \lambda_{i(k+1)}, \ldots\ldots, \lambda_{il}, \lambda_{j(l+1)}, \ldots\ldots\ldots, \lambda_{jn}]\,. \qquad (5.18)$$

The offspring are expected to obtain a combination of good information from their parents through this operator. Figure 5.6 is a successful example, in which child 1 inherits good genes from both its parents.

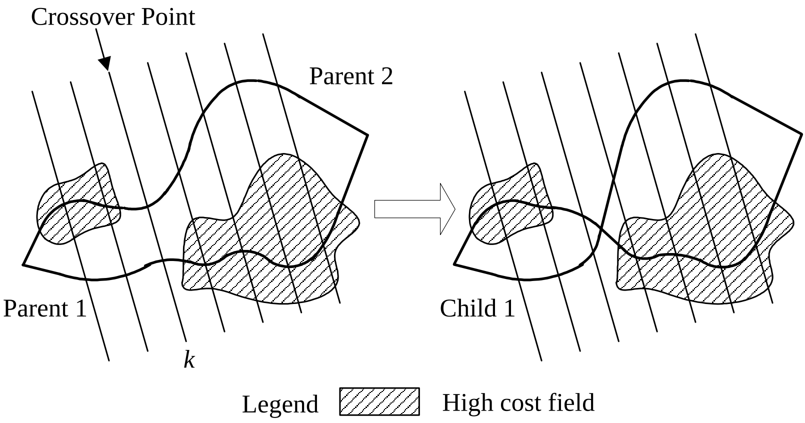


Figure 5.5: An example of a simple crossover operator.

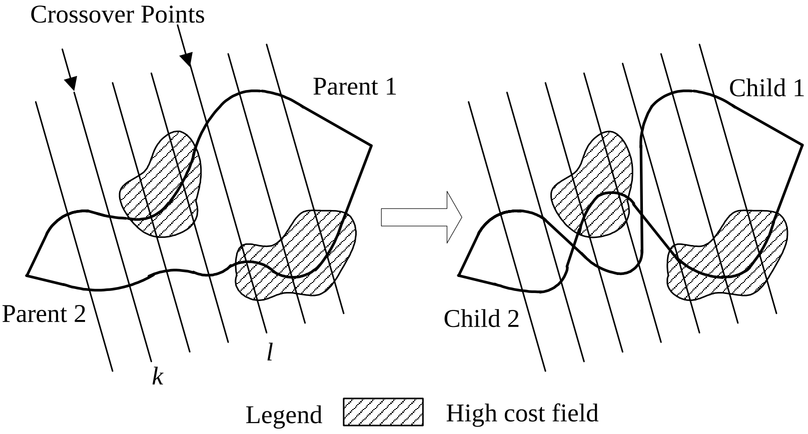


Figure 5.6: An example of a two-point crossover operator.

5.5.7 Arithmetic crossover

The basic concept of this kind of operator is borrowed from the definition of a convex set: any linear combination of two points in a convex set will also fall into the set (see e.g., Nash and Sofer, 1996). Therefore, if the feasible region of a constrained optimization problem is a convex set, then any linear combination of two feasible points will be feasible as well. Let $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, ..., \lambda_{in}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, ..., \lambda_{jn}]$ be two parents to be crossed. Then, based on the same concept, the resulting offspring are defined as a linear combination of two parent chromosomes, which guarantees the offspring is always feasible:

$$\Lambda'_i = \omega\Lambda_i + (1-\omega)\Lambda_j\,, \tag{5.19}$$

$$\Lambda'_j = \omega\Lambda_j + (1-\omega)\Lambda_i\,, \tag{5.20}$$

where $\omega = r_c[0,1]$.

The operator is also called convex crossover, linear crossover, and intermediate crossover (see Gen and Chang 1997, Michalewicz 1996). When $\omega = 0.5$, it yields a special case, which is usually called average crossover. An example of an Arithmetic Crossover operator is shown in Figure 5.7.

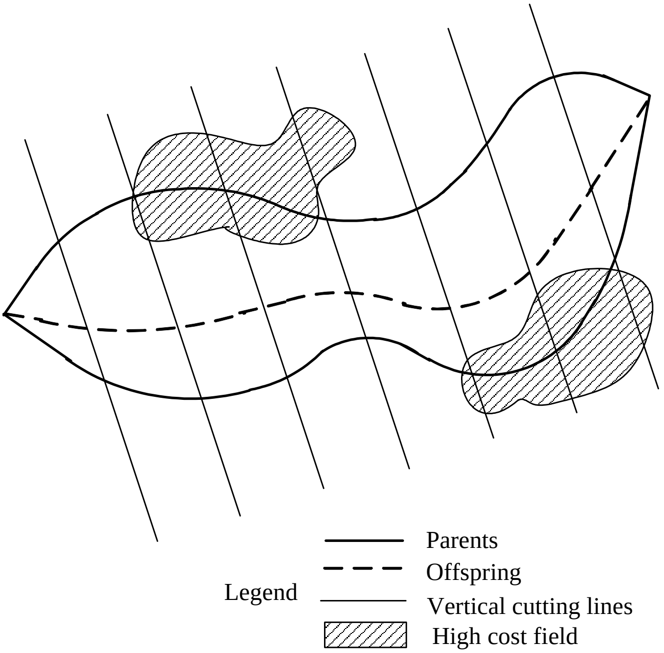


Figure 5.7: An example of an arithmetic crossover operator.

5.5.8 Heuristic crossover

The operator is a unique crossover for the following reasons: (1) it uses values of the fitness value in determining the direction of the search, (2) it produces only one offspring, and (3) the resulting offspring may not be feasible. Let $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, ..., \lambda_{in}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, ..., \lambda_{jn}]$ be two parents subjected to this operator, where we assume $C_T(\Lambda_i) \le C_T(\Lambda_j)$ (i.e., Λ_i is better or at least as good as Λ_j). Then the operator generates a single offspring Λ' according to the following rule:

$$\Lambda' = \omega\left(\Lambda_i - \Lambda_j\right) + \Lambda_i \,, \tag{5.21}$$

where $\omega = r_c\left[0,1\right]$.

It is possible for this operator to generate an offspring that is not feasible. In such a case, another random number is generated and another offspring is created. If after a certain number of user-defined attempts no new offspring can meet the boundary constraints defined in eqn (4.73), the operator gives up and returns Λ_i as the offspring. Figure 5.8 provides a successful example to illustrate the heuristic crossover operator.

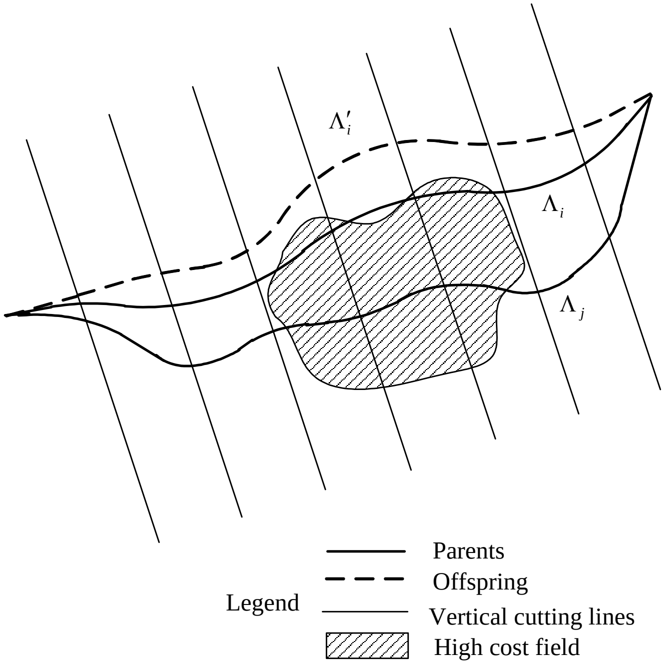


Figure 5.8: An example of a heuristic crossover operator.

5.6 Convergence

The termination rule of the proposed algorithm is to let the population evolve through a user-specified number of generations. This is the simplest way to stop the GA. Since the non-uniform mutation operator requires the maximum number of generations as an input parameter, our experience shows that in most situations, the objective keeps improving until the specified number of generations is reached. This occurs because at the final stages, the non-uniform operators start to locally refine the solutions.

Another stopping rule can be implemented by continuing the iterations for as long as the improvements are noticeable. However, our experience shows that sometimes the objective function remains unchanged through many generations before dropping sharply. This might occur because the objective function is very noisy. Since the computational cost is relatively small when compared to the total highway cost, this computation cost may be neglected. Therefore, if the objective function is noisy, it is suggested that the program should run until the pre-specified maximum number of generations is reached.

The best way to stop the program may be developed by combining some conventional optimization techniques into the GA. The experiences in some research show that such hybrid systems may outperform a pure GA (see Gen and Cheng, 1997; Goldberg, 1989; Michalewicz, 1996). However, such hybrid algorithms are not developed here.

5.7 Other issues

Several issues arise in implementing GAs. One of these is the population size. The larger the population, the more likely the GA is to explore the search space thoroughly. Goldberg has also shown that the efficiency of a GA in reaching a global optimum instead of local ones largely depends on the population size. However, a larger population also requires more memory and computational effort. In this study, we set the population size proportionally to the number of decision variables. For Model 1, a population size $n_p = 5n$ is adopted. The idea is based on the observation that for irregular terrain, the behavior of the objective function tends to be noisy. Then the optimal alignment is more circuitous and, thus, more intersection points are employed in the model. Since the population size is proportional to the number of decision variables (intersection points), a model with more decision variables should have a larger population, which enhances the GA's ability to find a near globally optimal solution.

The next issue is how to maintain a wider diversity of alleles in the gene pool as the population evolves over time. Mutation operators are devised for this purpose. However, there is a tendency that in latter generations, the chromosomes in the population resemble each other due to the mechanism of "survival of the fittest". If the objective function is very noisy, the chromosomes may stick in local optima. A solution to this problem is to regenerate the population while still preserving the best chromosome. In this way, we introduce new individuals into the population and keep a larger variety of gene pool to help in exploring the search space. In practice, the population can be regenerated after every fixed number of generations (e.g., 100 generations), or when the improvement in the objective function is not significant. Both options are included in our proposed algorithms. The idea is borrowed from the GENOCOP system developed by Michalewicz (1996).

Another issue is determining how many operators of each type to apply to the population at each generation. Instead of using fixed numbers, we leave these as user-defined parameters. A procedure is then developed to check the validity of the parameters. We avoid fixed numbers because the proper number of each operator type is largely determined by the problem itself. If the terrain and the location-dependent costs are very irregular, then additional mutation operators may help explore the search space. On the other hand, if the terrain is flat, more crossover operators might facilitate exploiting the merits of the better chromosomes.

It is noted that the proposed solution algorithm is probabilistic since it involves random numbers. For pseudo random number generators, once the random seed is determined, the resulting random stream is fixed. To make use of the stochastic features, a time-dependent random seed is utilized. In other words, each time we run the GA to solve the same problem, the solution will be different. If the time budget allows, we may run the program several times. Then the result with the lowest total cost will be selected as the final solution.

The last issue is the programming of the solution algorithms. Unlike in some other studies, where the problems are formulated in some typical forms readily solvable with existing software, this work relies on programming the solution algorithms because they are problem-specific and no existing software is applicable in solving the proposed model. Therefore, programming becomes one of the major tasks of this approach. The selection of programming language and program structures are quite essential to the efficiency of the proposed algorithms. The selected language for this research is C because some CAD software compilers (for example Microstation) recognize the C language. It allows the algorithms to be incorporated into some commercial software. Moreover, due to the complexities of the model, a large number of variables are defined in the algorithms. In order to increase the efficiency of the algorithms and save memory, the dynamic memory allocation feature provided by the C language is exploited in the programs. This helps the proposed algorithms to meet the necessary conditions (7) and (8) presented in section 2.7.

This page intentionally left blank

Chapter 6

Model and solution algorithms for optimizing backtracking horizontal alignments

In chapter 4, a model (Model 1) has been developed for optimizing non-backtracking horizontal alignments. However, if the terrain over which the alignment is optimized is extremely irregular, then it is very likely that the optimal alignment is backtracking. In such cases, Model 1 is unable to represent the optimal alignment. This problem may be solved by using a two-phase search. In the first phase, a non-backtracking alignment is optimized. In the second phase, some links resulting from the first search, which are more likely to have backward bends, are selected for further search. In this way, we are able to optimize backtracking alignments. Figure 6.1 illustrates the idea of the two-phase search method.

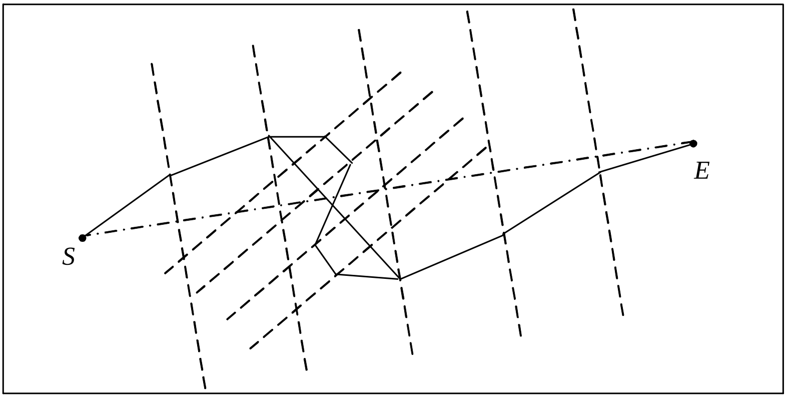


Figure 6.1: Two-phase search in irregular terrain.

Besides the two-phase search method, in this chapter we will develop another model that can optimize backtracking horizontal alignments in a one-phase search. In section 6.1 we will present the modeling approach for optimizing backtracking horizontal alignments. The cost function is then discussed in section 6.2, followed by the final model formulations. Since this model differs in structure from Model 1, the genetic encoding and operators are redefined in sections 6.4 and 6.5 so that they can be fitted into this model. Finally, an example is designed to demonstrate the performance of the model and solution algorithm.

6.1 Representation of alignment

In optimizing non-backtracking alignments, we know from Corollary 3.1 that each vertical cutting line will intersect the alignment at exactly one point. Unfortunately, in optimizing backtracking alignments, this property does not hold because some vertical cutting lines may intersect the alignment at more than one point, as shown in Figure 3.8b. In fact, we don’t know in advance whether the vertical cutting lines will intersect the alignment at one point or several points until we know the overall behavior of the alignment. This makes the modeling process more difficult.

To formulate the problem for backtracking alignments, we must think in another way. In this problem, the intersection points of the alignment are not restricted anymore to the vertical cutting planes. Figure 6.2 illustrates how the intersection points may be specified in a search process that allows backtracking alignments.

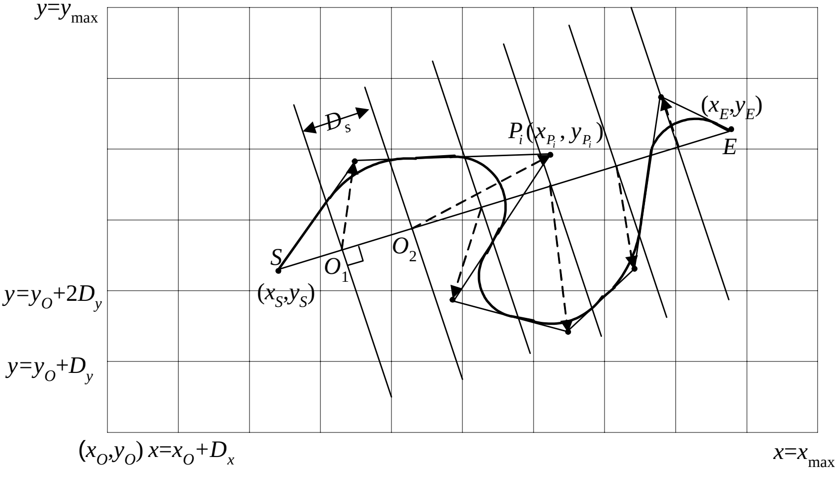


Figure 6.2: An example for modeling backtracking horizontal alignments.

Assume that we use n intersection points to describe the alignment and let $P_i(x_{P_i}, y_{P_i})$ be the i^{th} intersection point. Then the set of points P_i, $i = 0,.......,n+1$, where $P_0 = S$ and $P_{n+1} = E$ will be able to generate an alignment if Algorithm 4.1 is applied. Unlike Model 1, where each of the intersection points is determined by one decision variable, the intersection points for optimizing backtracking alignments require two decision variables (i.e., X and Y coordinates) to precisely locate their positions. Since we have no prior knowledge about the locations of the intersection points, they can be anywhere in the region of interest. That is

$$x_O \le x_{P_i} \le x_{\max}, \quad \forall\, i = 1,.....,n\,, \tag{6.1}$$

$$y_O \le y_{P_i} \le y_{\max}, \quad \forall\, i = 1,.....,n\,, \tag{6.2}$$

where (x_O, y_O) = the X,Y coordinates of the bottom-left corner of the study region

$(x_{\max}, y_{\max})$ = the X,Y coordinates of the top-right corner of the study region.

The modeling approach enables us to formulate both backtracking and non-backtracking horizontal alignments. However, we also have less control over the configurations of the alignments. For any given set of intersection points within the study region, the resulting alignments may become too circuitous (as for example in Figure 6.3). To ensure that the final alignment is reasonably shaped,

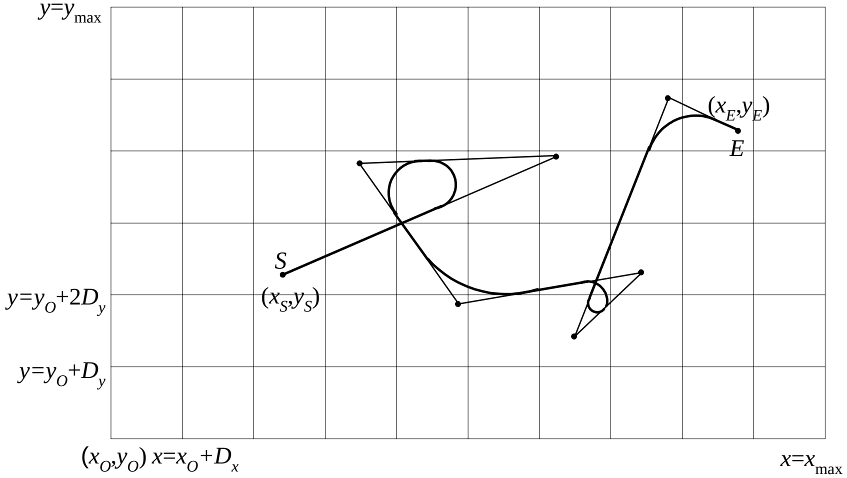


Figure 6.3: An example of undesirable alignments.

the solution algorithm must perform well enough to avoid unreasonable alignments. This relies much on the genetic operators through which we expect that the alignments will evolve to reasonable configurations adapted to the problem. We will discuss these in more detail in section 6.5.

6.2 Cost function

The cost computation for optimizing backtracking horizontal alignments is the same as for optimizing non-backtracking horizontal alignments. For detailed computation procedures and the logic behind the calculations, please refer to sections 4.3 to 4.6.

6.2.1 Location-dependent cost

Given a set of intersection points, an alignment alternative is obtained by applying Algorithm 4.1. The resulting alignment consists of tangent sections and circular curves. The computation procedures of location-dependent costs for tangent sections and circular curves are separated because their behaviors are quite different. The location-dependent cost is computed with Algorithm 4.2 for tangent sections, and with Algorithm 4.3 for circular curves. Note that the earthwork cost for a 2-dimensional alignment is imposed on location-dependent cost. Other location-dependent costs include land acquisition, soil stabilization, environmental impact, etc.

6.2.2 Length-dependent cost

Length-dependent cost consists of those costs proportional to the length of an alignment. If road width is fixed and the projected traffic demand is given, then area-dependent cost and VMT-dependent cost can be converted to length-dependent cost as well. Length-dependent cost is computed by multiplying total alignment length by unit-length-dependent cost.

6.2.3 User costs

User costs consist of fuel-consumption cost, travel time cost, and accident cost. The user costs of an alignment depend on future traffic demand and various features of the alignment, such as curvature, length of curve, gradient, etc. User costs are computed over the whole analysis period and discounted to net present values. The detailed calculations are presented in section 4.6.

6.3 Final model and its properties

The total cost of an alignment alternative is obtained by the summation of different types of costs. Recall that the decision variables in the proposed model are the coordinates of the set of intersection points whose domains are defined in eqns (6.1) and (6.2). Therefore, the final model can be formulated as follows:

Model 2 – Model for optimizing backtracking horizontal alignments

$$\underset{x_{P_1}, y_{P_1}, x_{P_2}, y_{P_2}, \ldots\ldots\ldots, x_{P_n}, y_{P_n}}{\text{Minimize}} \qquad C_T = C_N + C_L + C_U \,, \tag{6.3}$$

$$\text{subject to } \; x_O \le x_{P_i} \le x_{\max}, \qquad \forall\, i = 1, \ldots.., n \,, \tag{6.4}$$

$$y_O \le y_{P_i} \le y_{\max}, \qquad \forall\, i = 1, \ldots.., n \,, \tag{6.5}$$

where the alignment is generated with Algorithm 4.1

C_T = total cost ($)
C_N = location-dependent cost ($), computed with Algorithms 4.2 and 4.3
C_L = length-dependent cost ($), given in eqn (4.35)
C_U = user cost ($), given in eqn (4.71)
(x_O, y_O) = the X, Y coordinates of the lower-left corner of the study region
(x_{P_i}, y_{P_i}) = the X, Y coordinates of intersection point P_i
$(x_{\max}, y_{\max})$ = the X, Y coordinates of the top-right corner of the study region.

The structure of Model 2 is similar to Model 1 presented in section 4.7. The only differences are the decision variables and the domain constraints. It should be noted that Model 2 shares the properties of Model 1, as mentioned in section 4.7. In addition, unlike Model 1, Model 2 can optimize a non-backtracking alignment.

The decision variables in Model 1 are the coordinates of intersection points on vertical cutting lines, whereas the decision variables in Model 2 are the coordinates of intersection points in Cartesian coordinate system. For calculating the cost associated with an alignment in Model 1, the coordinates of intersection points must be first transformed into Cartesian coordinate system by applying eqn (4.8). In Model 2, the coordinate transformation is not necessary. However, the total number of decision variables is double that in Model 1 if the same number of intersection points is adopted. From the standpoint of their search spaces, these two models are quite different despite their many similarities. Accordingly, the genetic encoding and genetic operators must be redefined to fit into the model. These are discussed in the following two sections.

6.4 Genetic encoding and initial population

In Model 2, each intersection point is determined by two variables. For an alignment with *n* intersection points, the encoded solution will consist of 2*n* genes. Therefore, the chromosome is defined as

$$\Lambda = [\lambda_1, \lambda_2, \lambda_3, \lambda_4, \ldots\ldots, \lambda_{2n-1}, \lambda_{2n}] = [x_{P_1}, y_{P_1}, x_{P_2}, y_{P_2}, \ldots\ldots, x_{P_n}, y_{P_n}] \,, \tag{6.6}$$

where Λ = chromosome

λ_i = the i^{th} gene, for all $i = 1,.......,2n$
(x_{P_i}, y_{P_i}) = the coordinates of the i^{th} intersection point, for all $i = 1,.......,n$.

It can be seen that the mappings between the genes in a chromosome and the coordinates of the intersection points are

$$\lambda_{2i-1} = x_{P_i}\,,\ \forall i = 1,.......,n\,, \tag{6.7}$$

$$\lambda_{2i} = y_{P_i}\,,\ \forall i = 1,.......,n\,, \tag{6.8}$$

and thus

$$x_O \le \lambda_{2i-1} \le x_{\max}, \qquad \forall\, i = 1,.....,n\,, \tag{6.9}$$

$$y_O \le \lambda_{2i} \le y_{\max}, \qquad \forall\, i = 1,.....,n\,. \tag{6.10}$$

Eqns (6.9) and (6.10) will play an important role in generating initial population and developing genetic operators.

Three types of potential solutions are produced in the initial population:

(1) Intersection points lie on the straight line connecting the start and end points at equal distances

In this case, the set of intersection points represents a straight alignment, which reduces length-dependent cost to a minimum. The intersection points are equivalent to the origins of vertical cuts. Therefore, the chromosome is defined as

$$\Lambda = [\lambda_1, \lambda_2, ..., \lambda_{2n-1}, \lambda_{2n}] = [x_{O_1}, y_{O_1}, ..., x_{O_n}, y_{O_n}]\,, \tag{6.11}$$

where (x_{O_i}, y_{O_i}) = the coordinates of the origin of the i^{th} vertical cut, as given in eqn (4.2).

(2) Intersection points lie randomly on the vertical cuts

In this case, each gene of the chromosomes is generated by the coordinate transformation of a continuous random number within the boundaries of the corresponding vertical cut. Following the same notation used in the previous chapters, the genes of the chromosomes are defined as

$$\lambda_{2i-1} = x_{O_i} + r_c[d_{iL}, d_{iU}]\cos\theta, \forall i = 1,...., n\,, \tag{6.12}$$

$$\lambda_{2i} = y_{O_i} + r_c[d_{iL}, d_{iU}]\sin\theta, \forall i = 1,...., n\,, \tag{6.13}$$

where $r_c[d_{iL}, d_{iU}]$ = random number from a continuous uniform distribution whose domain is within the interval $[d_{iL}, d_{iU}]$

d_{iL} and d_{iU} are lower bound and upper bound of the i^{th} vertical cut, as defined in eqns (4.4) to (4.7)

θ = the angle of vertical cuts, given in eqn (4.3).

(3) Intersection points scattered randomly within the study region

This type of population maintains a large variety in its gene pool and carries useful information about backtracking alignments. The genes of the chromosomes are defined as follows:

$$\lambda_{2i-1} = r_c[x_O, x_{\max}], \forall i = 1,....,n \, , \tag{6.14}$$

$$\lambda_{2i} = r_c[y_O, y_{\max}], \forall i = 1,....,n \, . \tag{6.15}$$

According to the discussions in section 5.7, the population size is set to be proportional to the number of decision variables to reflect the dimension of the search space and the complexity of the objective function. If n intersection points are used in the model, the total number of genes in a chromosome will be $2n$. Therefore, a population size $n_p = 10n$ is suggested.

6.5 Genetic operators

The genetic operators for Model 2 and the logic behind each operator are the same as Model 1. The only differences are the definitions of genes. Each gene in Model 1 represents an intersection point, whereas in Model 2, an intersection point is encoded with 2 genes. Note that all operators are developed to work on intersection points rather than individual genes to increase the variety of the corresponding alignment. In this section, each operator is briefly discussed using the new definitions of genes. For detailed ideas behind each operator, please refer to section 5.5.

6.5.1 Uniform mutation

Let $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_{2n-1}, \lambda_{2n}]$ be the chromosome to be mutated. If the k^{th} intersection point is selected to apply the operator, where $k = r_d[1, n]$, a random number generated from a discrete uniform distribution whose domain is defined within the interval $[1, n]$, then the corresponding genes λ_{2k-1} and λ_{2k} will be replaced by

$$\lambda'_{2k-1} = r_c[x_O, x_{\max}] \, , \tag{6.16}$$

$$\lambda'_{2k} = r_c[y_O, y_{\max}] \, . \tag{6.17}$$

As mentioned in section 5.5, a curve elimination procedure is required to prevent the resulting alignment from getting trapped at a local optimum. The procedure is summarized as follows:

Algorithm 6.1 Curve elimination procedure for Model 2

(1) Step 1: Encode S and E into the chromosome
Expand the chromosome $\Lambda = [\lambda_1, \lambda_2, ..., \lambda'_{2k-1}, \lambda'_{2k}, ..., \lambda_{2n-1}, \lambda_{2n}]$ to

$$\begin{aligned}\Lambda &= [\lambda_{-1}, \lambda_0, \lambda_1, ..., \lambda'_{2k-1}, \lambda'_{2k}, ...\lambda_{2n}, \lambda_{2n+1}, \lambda_{2n+2}] \\ &= [x_S, y_S, \lambda_1, ..., \lambda'_{2k-1}, \lambda'_{2k}, ..., \lambda_{2n}, x_E, y_E]\end{aligned}$$

(2) Step 2: Generate two independent random loci i and j

$$i = r_d[0, k-1] \text{ and } j = r_d[k+1, n+1]$$

(3) Step 3: Change the values of the intermediate genes between the i^{th} and k^{th} genes

$$\lambda'_{2l-1} = \lambda_{2i-1} + (l-i)\frac{(\lambda'_{2k-1} - \lambda_{2i-1})}{k-i}, \text{ for all } l = i+1, ..., k-1 \tag{6.18}$$

$$\lambda'_{2l} = \lambda_{2i} + (l-i)\frac{(\lambda'_{2k} - \lambda_{2i})}{k-i}, \text{ for all } l = i+1, ..., k-1 \tag{6.19}$$

(4) Step 4: Change the values of the intermediate genes between the k^{th} and j^{th} genes

$$\lambda'_{2l-1} = \lambda_{2k-1} + (l-k)\frac{(\lambda_{2j-1} - \lambda'_{2k-1})}{j-k}, \text{ for all } l = k+1, ..., j-1 \tag{6.20}$$

$$\lambda'_{2l} = \lambda_{2k} + (l-k)\frac{(\lambda_{2j} - \lambda'_{2k})}{j-k}, \text{ for all } l = k+1, ..., j-1 \tag{6.21}$$

(5) Step 5: Remove the encoded genes of S and E from the chromosome
Truncate the resulting chromosome

$$\Lambda' = [x_S, y_S, \lambda_1, ..., \lambda'_{2i+1}, \lambda'_{2i+2}, ..., \lambda'_{2k-1}, \lambda'_{2k}, ..., \lambda'_{2j-3}, \lambda'_{2j-2}, ..., \lambda_{2n}, x_E, y_E] \text{ to}$$
$$\Lambda' = [\lambda_1, ..., \lambda'_{2i+1}, \lambda'_{2i+2}, ..., \lambda'_{2k-1}, \lambda'_{2k}, ..., \lambda'_{2j-3}, \lambda'_{2j-2}, ..., \lambda_{2n}]$$

6.5.2 Straight mutation

Let $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_{2n-1}, \lambda_{2n}]$ be the chromosome to be mutated. We randomly generate two independent discrete random numbers i and j, where $i = r_d[0, n+1]$, $j = r_d[0, n+1]$, $i \neq j$, and $i < j$. The alleles of the intermediate genes between the $(2i)^{th}$ and $(2j-1)^{th}$ genes will be replaced by

$$\lambda'_{2l-1} = \lambda_{2i-1} + (l-i)\frac{(\lambda_{2j-1} - \lambda_{2i-1})}{j-i}, \quad \text{for all } l = i+1, ..., j-1, \tag{6.22}$$

$$\lambda'_{2l} = \lambda_{2i} + (l-i)\frac{(\lambda_{2j} - \lambda_{2i})}{j-i}\text{ , for all } l = i+1,......, j-1 . \tag{6.23}$$

In the above equations, if $i = 0$ (which represents the start point S), we then set $\lambda_{2i-1} = \lambda_{-1} = x_S$ and $\lambda_{2i} = \lambda_0 = y_S$. Similarly if $j = n+1$ (which denotes the end point *E*), then $\lambda_{2j-1} = \lambda_{2n+1} = x_E$ and $\lambda_{2j} = \lambda_{2n+2} = y_E$. Note that the operator places all intermediate intersection points between the i^{th} and the j^{th} intersection points on the line segment connecting these two points at equal spacing.

6.5.3 Non-uniform mutation

For a given parent $\Lambda = [\lambda_1, \lambda_2 ,......, \lambda_{2n-1}, \lambda_{2n}]$, where the k^{th} $(k = r_d[1,n])$ intersection point is selected for mutation, we first generate two random binary digits $r_d[0,1]$. Then the alleles of λ'_{2k} and λ'_{2k-1} in the resulting offspring $\Lambda' = [\lambda_1, \lambda_2 ,..., \lambda'_{2k-1}, \lambda'_{2k} ,..., \lambda_{2n-1}, \lambda_{2n}]$ are determined by the following rules:

(1) If the first random digit $r_d[0,1] = 0$, then

$$\lambda'_{2k-1} = \lambda_{2k-1} - f(t, \lambda_{2k-1} - x_O) , \tag{6.24}$$

$$\lambda'_{2k-1} = \lambda_{2k-1} + f(t, x_{\max} - \lambda_{2k-1}) . \tag{6.25}$$

(2) If the second random digit $r_d[0,1] = 0$, then

$$\lambda'_{2k} = \lambda_{2k} + f(t, \lambda_{2k} - y_O) , \tag{6.26}$$

$$\lambda'_{2k} = \lambda_{2k} + f(t, y_{\max} - \lambda_{2k-1}) . \tag{6.27}$$

The definitions of the function f and its argument are presented in section 5.5, and are not repeated here. Note that in order to prevent the resulting alignment from sticking at a local optimum, the non-uniform mutation is immediately followed by the curve elimination procedure, as shown in Algorithm 6.1.

6.5.4 Whole non-uniform mutation

This operator applies the non-uniform operator to each pair of genes for a given chromosome in a randomly generated sequence. The resulting offspring will be totally different from its parent. The mutation range is relatively large in early generations and reduced in later generations.

6.5.5 Simple crossover

Let two parents $\Lambda_i = [\lambda_{i1}, \lambda_{i2} ,........., \lambda_{i(2n)}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2} ,........., \lambda_{j(2n)}]$ be crossed after a randomly generated position $2k$, where $k = r_d[1,n]$. Then the resulting offspring are

$$\Lambda_i' = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots, \lambda_{i(2k)}, \lambda_{j(2k+1)}, \ldots\ldots, \lambda_{j(2n)}], \tag{6.28}$$

$$\Lambda_j' = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots, \lambda_{j(2k)}, \lambda_{i(2k+1)}, \ldots\ldots, \lambda_{i(2n)}]. \tag{6.29}$$

6.5.6 Two-point crossover

Let $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots\ldots, \lambda_{i(2n)}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots\ldots, \lambda_{j(2n)}]$ denote the two parents to be crossed between two randomly generated positions $2k$ and $2l$, where $k = r_d[1, n]$, $l = r_d[1, n]$, $k \neq l$, and $k < l$. The resulting offspring are

$$\Lambda_i' = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots, \lambda_{i(2k)}, \lambda_{j(2k+1)}, \ldots\ldots, \lambda_{j(2l)}, \lambda_{i(2l+1)}, \ldots\ldots\ldots, \lambda_{in}], \tag{6.30}$$

$$\Lambda_j' = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots, \lambda_{j(2k)}, \lambda_{i(2k+1)}, \ldots\ldots, \lambda_{i(2l)}, \lambda_{j(2l+1)}, \ldots\ldots\ldots, \lambda_{jn}]. \tag{6.31}$$

6.5.7 Arithmetic crossover

Given two parents $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots\ldots, \lambda_{i(2n)}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots\ldots, \lambda_{j(2n)}]$, the arithmetic crossover generates two offspring as follows

$$\Lambda_i' = \omega\Lambda_i + (1-\omega)\Lambda_j, \tag{6.32}$$

$$\Lambda_j' = \omega\Lambda_j + (1-\omega)\Lambda_i, \tag{6.33}$$

where $\omega = r_c[0,1]$.

6.5.8 Heuristic crossover

Let the two parents to undergo this operator be denoted by $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots\ldots, \lambda_{i(2n)}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots\ldots, \lambda_{j(2n)}]$, where we assume $C_T(\Lambda_i) \leq C_T(\Lambda_j)$ (i.e., Λ_i is at least as good as Λ_j). Then the operator generates a single offspring Λ' according to the following rule:

$$\Lambda' = \omega\left(\Lambda_i - \Lambda_j\right) + \Lambda_i, \tag{6.34}$$

where $\omega = r_c[0,1]$.

It is possible for this operator to generate an offspring that is not feasible. In such a case, another random number is generated and another offspring is created. If after a certain number of user-defined attempts no new offspring can meet the boundary constraints defined in eqns (6.4) and (6.5), the operator gives up and returns Λ_i as the offspring.

6.6 An example

In this section, we investigate the performance of the proposed solution algorithm. Here, the main concerns are whether the algorithm can find a relatively good solution and whether it is efficient. Therefore, the example is designed in such a way that we know in advance the globally or near globally optimal solution. The test case map is shown in Figure 6.4.

In Figure 6.4, the shade of a cell represents its location-dependent cost. The darker the shade, the higher the location-dependent cost of the cell. The cross-patterned areas on the right side of the map represent the inaccessible region, from which the alignment is excluded. The map shows that the alignment must wind along the low cost cells, and skirt the high cost cells to minimize total cost.

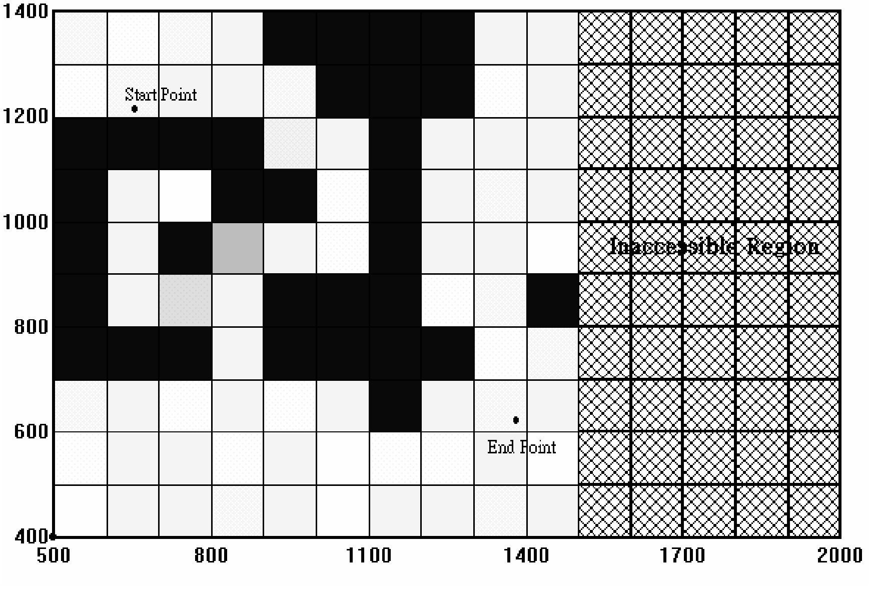


Figure 6.4: The testing map.

We now run the program to see whether it can find the globally or near globally optimal solution. In order to provide visual inspections, a Graphical User Interface (GUI) is designed to display the evolution of the alignments. The program's specifications for this test example are summarized as follows:

Parameter	Value
Population size n_p	50
Number of intersection points n	10
Maximal number of generations n_T	500

To visualize the evolution of the program, we print the best alignment found at the 100th, 300th, and 500th generations. The results are shown in
Figure 6.5 to Figure 6.7. The figures indicate how the best alignment in the population evolves. As we can see, at the 100th generation, the alignment is not very good since it is too long and passes through two very expensive black cells. However, at the 300th generation, the alignment seems nearly optimal.

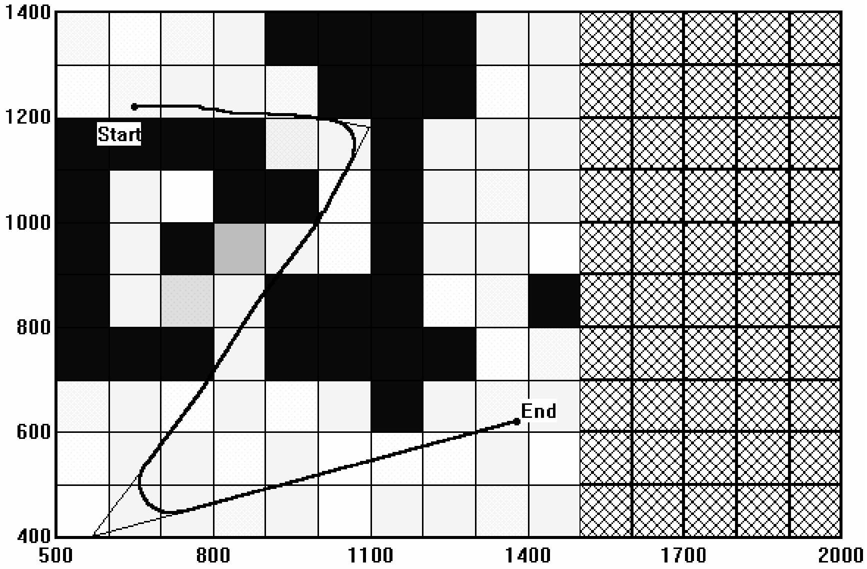


Figure 6.5: The best alignment at the 100th generation.

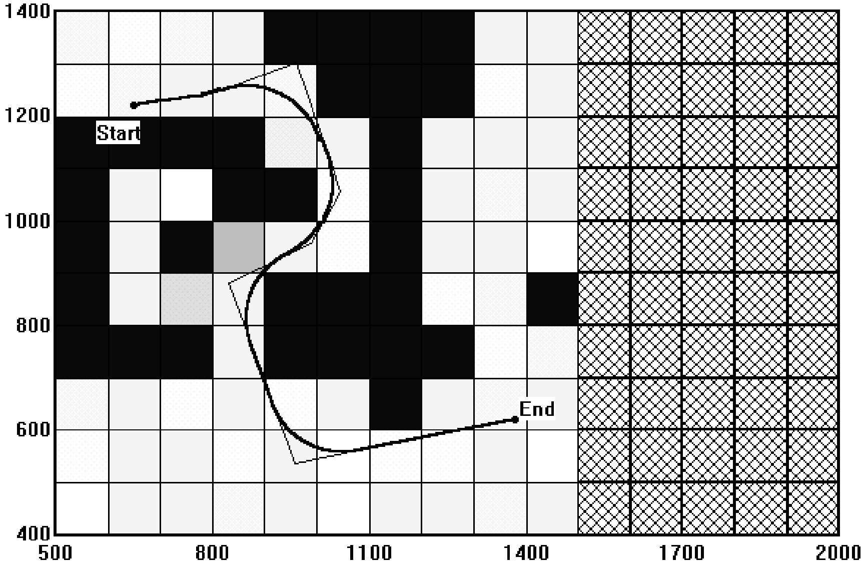


Figure 6.6: The best alignment at the 300th generation.

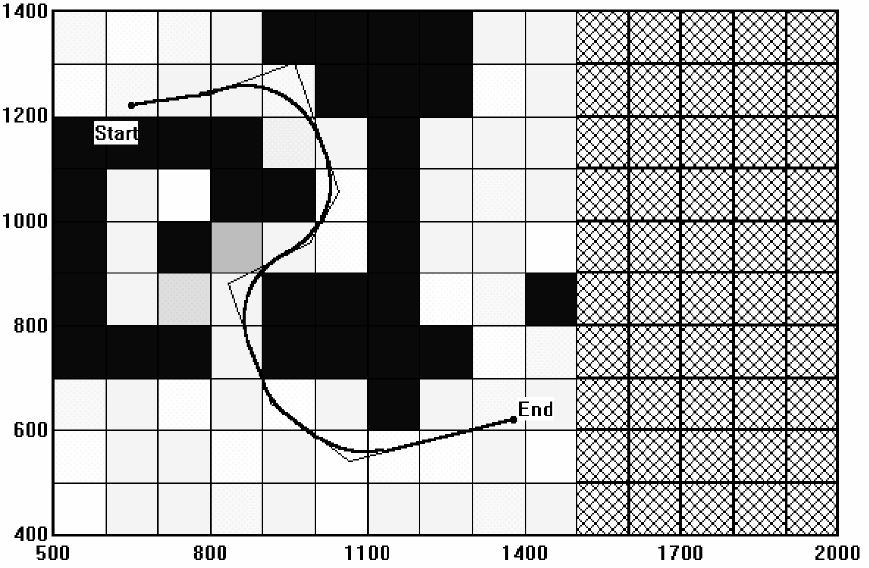


Figure 6.7: The best alignment at the 500th generation.

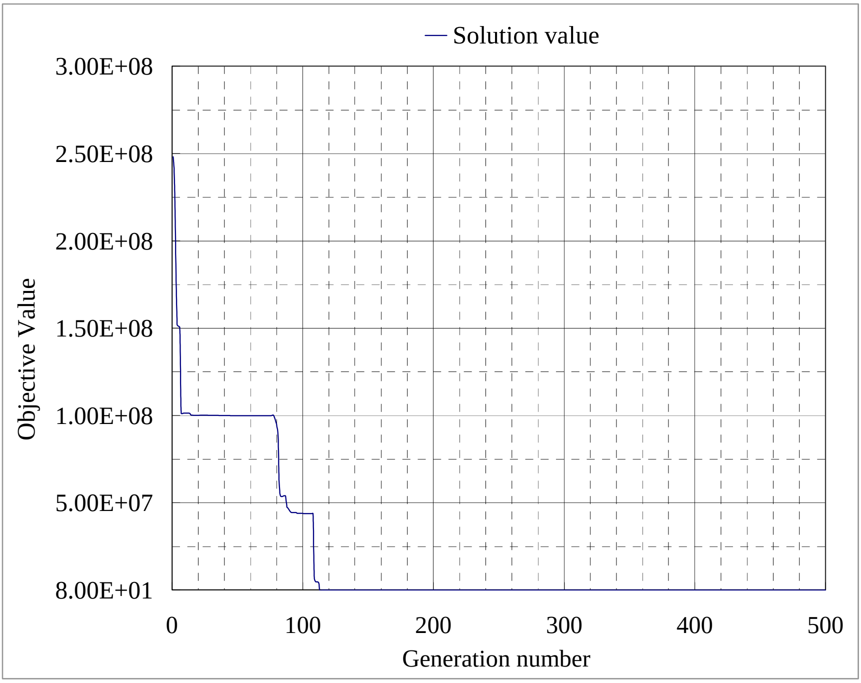


Figure 6.8: Objective values over successive generations for the test example.

To indicate how solutions improve over successive generations, the objective function values in each generation are plotted in Figure 6.8. It shows that in the initial stage of the search, the objective value is extremely high. That is because the example is designed in such a way that any randomly generated alignment will very probably pass through the high cost cells. After ten generations, the objective value drops sharply from 3×10^8 to 1×10^8, and it further drops to 5×10^7 around the 85th generation. After 120 generations, the objective value is very close to the optimal value found at the 500th generation (about 89).

This example shows that the GA algorithm works very well even in finding a path through a maze. This approach seems to be promising for optimizing highway alignments because it does not require gradient information about the objective function, which is unavailable in our model, and has the potential to find a near-globally optimal solution. These features give us confidence in developing models for simultaneously optimizing 3-dimensional alignments.

Chapter 7

Model and solution algorithms for optimizing non-backtracking 3-dimensional alignments

This chapter concentrates on developing a model for simultaneously optimizing non-backtracking 3-dimensional alignments. In the proposed model, the earthwork cost for a particular alignment is computed directly rather than imposed on the location-dependent costs. In addition, the estimation of user costs will now also depend on vertical profiles. Unlike 2-dimensional models, in which the gradient and vertical curvature constraints are not explicitly taken into account, the 3-dimensional model will consider these constraints directly.

This model can be deemed as an extension of Model 1. However, its genetic representations of the chromosomes and genetic operators are more complex. At the end of this chapter, we will develop problem-specific operators for the search algorithm.

7.1 Representation of alignment

The basic concept for representing non-backtracking 3-dimensional alignments applies vertical cutting planes to the line segment $\overline{SE}$, as shown in Figure 7.1. Then from Corollary 3.1, we know that the optimal alignment will intersect each vertical cutting plane at exactly one point in 3-dimensional space. Based on the set of intersection points, we will be able to generate a 3-dimensional alignment.

Note that in this model, each intersection point in the 3-dimensional space can be determined by only two decision variables (the abscissa and ordinate on the vertical cutting plane). This helps reduce the dimensions of the search space. Then by trigonometry, we can further transform these intersection points into the Cartesian coordinate system.

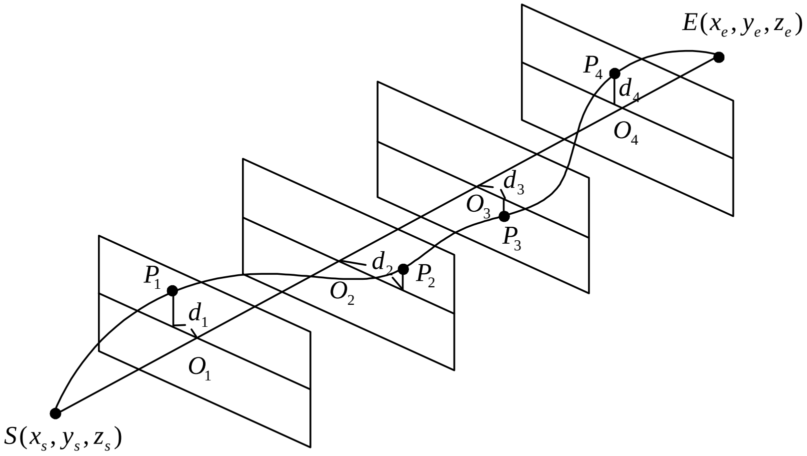


Figure 7.1: The modeling structure for non-backtracking 3-dimensional alignments.

Let O_i be the point at which the line segment $\overline{SE}$ intersects the i^{th} cutting plane, where S and E are the start and end points of the alignment. Then the X and Y coordinates of O_i can be obtained with eqn (4.2). On each vertical cutting plane, the abscissa, denoted by d_i, is defined as the axis that passes through O_i and parallels the XY plane, with O_i as its origin. The direction of the abscissa d_i and its boundaries are defined as in Model 1. The ordinate on the i^{th} cutting plane is simply defined as the Z coordinate in the Cartesian coordinate system to reduce coordinate transformation requirements. Let P_i be the intersection point on the i^{th} cutting plane, whose coordinates are (d_i, z_i). Then the Cartesian coordinates of P_i, denoted by $(x_{P_i}, y_{P_i}, z_{P_i})$ can be obtained by

$$\begin{bmatrix} x_{P_i} \\ y_{P_i} \\ z_{P_i} \end{bmatrix} = \begin{bmatrix} x_{O_i} \\ y_{O_i} \\ 0 \end{bmatrix} + \begin{bmatrix} d_i \cos\theta \\ d_i \sin\theta \\ z_i \end{bmatrix}, \qquad (7.1)$$

where (x_{O_i}, y_{O_i}) = the coordinates of the origin of the abscissa on the i^{th} cutting plane, obtained with eqn (4.2)

θ = the angle of cutting planes on the XY plane.

It can be seen from eqn (7.1) that d_i alone determines the X and Y coordinates of intersection point P_i, while z_i is equal to the Z coordinate of P_i. Now given a set of intersection points $P_i, i = 1, \ldots\ldots, n$, the corresponding horizontal alignment can be obtained after applying Algorithm 4.1. For the

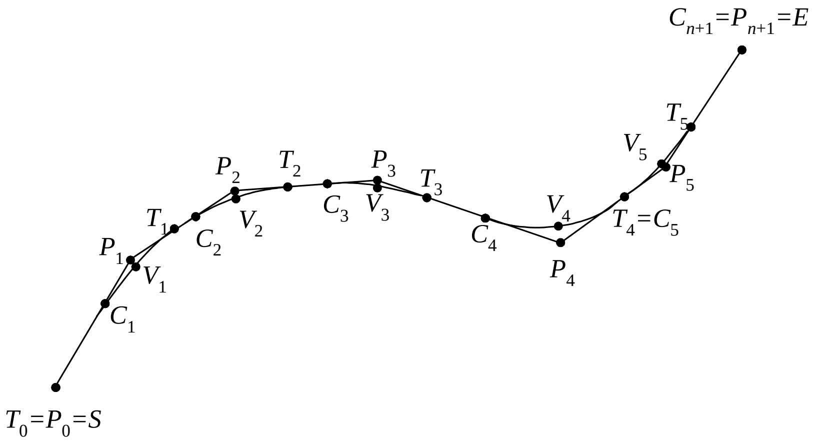


Figure 7.2: Geometric relations of reference points for an alignment on the XY plane.

vertical alignment, we assume that the elevations of the middle points of circular curves are controlled by the Z coordinates of the set of intersection points P_i 's. Figure 7.2 shows the geometric relations of the reference points of an alignment, in which $V_i(x_{V_i}, y_{V_i}, z_{V_i})$ denotes the middle point of the i^{th} circular curve, where $z_{V_i} = z_{P_i}$. The X,Y coordinates of V_i can be obtained by vector analysis. Let $\delta_i(x_{\delta_i}, y_{\delta_i})$ be the coordinates of the center of the i^{th} circular curve as shown in Figure 7.3. Then (x_{V_i}, y_{V_i}) are the coordinates of the point that lies on the vector from δ_i to P_i at the distance R_i , where R_i is the radius of the i^{th} circular curve. In mathematical form, we get

$$\begin{bmatrix} x_{V_i} \\ y_{V_i} \end{bmatrix} = \begin{bmatrix} x_{\delta_i} \\ y_{\delta_i} \end{bmatrix} + R_i \frac{\mathbf{P}_i - \boldsymbol{\delta}_i}{\|\mathbf{P}_i - \boldsymbol{\delta}_i\|}, \tag{7.2}$$

where $(x_{\delta_i}, y_{\delta_i})$ are given in eqn (4.22).

If we imagine a surface orthogonal to the XY plane along the horizontal alignment and stretch it to be flat (following the definition given in chapter 3, it is called the HZ plane, where H is the distance measured from S along the horizontal alignment), then the set of points $S,V_1,......,V_n,E$ on the HZ plane are the control points for the corresponding vertical alignment. Figure 7.4 illustrates the set of control points on the HZ plane for the example shown in Figure 7.2, in which each pair of successive control points is connected by a tangent segment. The piecewise linear segments in Figure 7.4 provide a rough appearance of the vertical alignment.

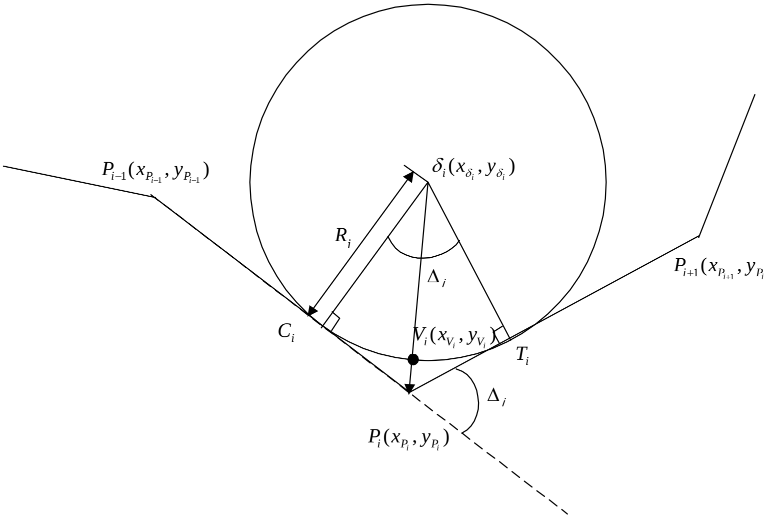

$P_{i-1}(x_{P_{i-1}}, y_{P_{i-1}})$
$\delta_i(x_{\delta_i}, y_{\delta_i})$
R_i
Δ_i
$P_{i+1}(x_{P_{i+1}}, y_{P_{i+1}})$
C_i
$V_i(x_{V_i}, y_{V_i})$
T_i
Δ_i
$P_i(x_{P_i}, y_{P_i})$

Figure 7.3: The geometric relations among V_i, δ_i, and P_i.

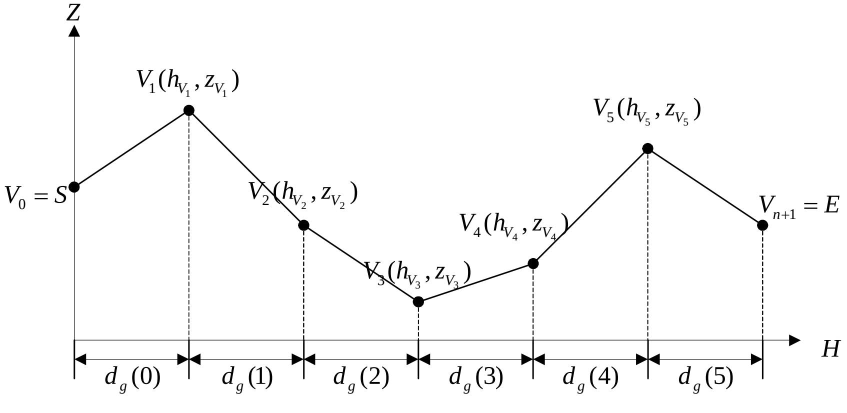


Figure 7.4: The control points on the HZ plane.

To fit the vertical alignment to the set of the control points $S, V_1, ..., V_n, E$, their coordinates on the HZ plane must be determined. For notational convenience, let V_0 and V_{n+1} denote S and E on the HZ plane respectively. Then the H coordinate of the point V_i, denoted by h_{V_i}, $i = 0, ..., n+1$, can be calculated with

$$h_{V_i} = 0, \text{ if } i = 0, \tag{7.3}$$

$$h_{V_i} = \sum_{k=0}^{i-1} \sqrt{(x_{T_k} - x_{C_{k+1}})^2 + (y_{T_k} - y_{C_{k+1}})^2} + \sum_{k=1}^{i-1} R_k \Delta_k + \frac{1}{2} R_i \Delta_i, \text{ if } 1 \le i \le n, \qquad (7.4)$$

$$h_{V_i} = \sum_{k=0}^{n} \sqrt{(x_{T_k} - x_{C_{k+1}})^2 + (y_{T_k} - y_{C_{k+1}})^2} + \sum_{k=1}^{n} R_k \Delta_k, \text{if } i = n+1, \qquad (7.5)$$

where (x_{C_k}, y_{C_k}) = the coordinates of C_k, which are given in eqn (4.15)
(x_{T_k}, y_{T_k}) = the coordinates of T_k, which are given in eqn (4.16)
R_k = the radius of the circular curve from C_k to T_k, obtained with Algorithm 4.1
Δ_k = the intersection angle at intersection point P_k, obtained with eqn (4.10).

Note that in the above equation, T_0 denotes S and C_{n+1} represents E for notational convenience.

The horizontal distance between any two consecutive control points V_i and V_{i+1} on the HZ plane, denoted by $d_g(i)$, $i = 0,.......,n$, can be obtained by

$$d_g(i) = h_{V_{i+1}} - h_{V_i}. \qquad (7.6)$$

Let g_i be the grade (percent) at the tangent segment $\overline{V_iV_{i+1}}$, $i = 0,.......,n$. Then

$$g_i = 100 \frac{z_{V_{i+1}} - z_{V_1}}{d_g(i)}. \qquad (7.7)$$

At each control point V_i, $i = 1,.......,n$, on the HZ plane, a parabolic vertical curve is used to smoothly connect the tangent segments $\overline{V_{i-1}V_i}$ and $\overline{V_iV_{i+1}}$. In a "crest", gradients decrease along the curve, while in a "sag", gradients increase along the curve. For example, the resulting parabolic curves at V_1 and V_5 in Figure 7.4 are crest curves, while those at $V_2, V_3,$ and V_4 are sag curves.

The minimum length of a parabolic curve is dominated by the stopping sight distance corresponding to the design speed, and the change of road grade. The vertical curves must be designed in such a way that the minimum sight distance is satisfied. The stopping sight distance consists of (1) the distance traveled from the time the object is sighted to the instant the brakes are applied, and (2) the distance required for stopping the vehicle after the brakes are applied. According to AASHTO (2001), the total stopping sight distance may be expressed as:

$$S_d = 1.467 V_d t_r + \frac{V_d^2}{30(f_r \pm g)}, \qquad (7.8)$$

where S_d = stopping sight distance (ft)

V_d = design speed (mph)
t_r = perception-reaction time (sec)
f_r = coefficient of forward rolling friction (decimal)
g = road grade (decimal).

In eqn (7.8), the perception-reaction time $t_r = 2.5$ sec, as is recommended by AASHTO (2001). The coefficient of rolling friction f_r depends on running speed, vehicle tires, and road surface type (dry or wet). The recommended value of f_r for highway design at different design speeds can also be found in AASHTO (2001). Since grade keeps changing along a parabolic vertical curve, an appropriate value of road grade g must be specified for calculating stopping sight distance. In this book, the road grade g for two different road sections may be taken as their average value. Suppose that we want to find the sight distance on the vertical curve at V_i. Then the road grade can be determined as follows:

$$\overline{g}_i = \frac{1}{200}\left(g_{i-1} + g_i\right) = \frac{1}{2}\left(\frac{z_{V_i} - z_{V_{i-1}}}{d_g(i-1)} + \frac{z_{V_{i+1}} - z_{V_i}}{d_g(i)}\right), \tag{7.9}$$

where $\overline{g}_i$ = average road grade (decimal) used for calculating the sight distance on the vertical curve at V_i.

Let $S_d(i)$ be the sight distance on the vertical curve at V_i. Taking the above information into account, we obtain

$$S_d(i) = 3.67V_d + \frac{V_d^2}{30(f_r + \overline{g}_i)}. \tag{7.10}$$

The relations among minimum length of parabolic curve, sight distance, and change of road grade depend on the configuration of vertical curve (crest or sag) and whether the length of the parabolic curve is greater than the required sight distance. The relations can be found in AASHTO (2001), Wright (1996), and Underwood (1991). The derivations of the relations are presented more detail in Underwood (1991). The results are given as follows:

(1) Crest Curve

$$L_m = \frac{A_\Delta S_d^2}{100\left(\sqrt{2h_d} + \sqrt{2h_o}\right)^2}, \text{ if } L_m > S_d, \tag{7.11}$$

$$L_m = 2S_d - \frac{100\left(\sqrt{2h_d} + \sqrt{2h_o}\right)^2}{A_\Delta}, \text{ if } L_m < S_d, \tag{7.12}$$

where L_m = minimum length of vertical curve (ft)
A_Δ = algebraic difference in grades (percent)
S_d = sight distance (ft)
h_d = height of driver's eye above roadway surface (ft)
h_o = height of object above roadway surface (ft).

For the design purposes, $h_d = 3.5'$ and $h_o = 2.0'$ are suggested. Inserting these numbers into eqns (7.11) and (7.12), we get

$$L_m = \frac{A_\Delta S_d^2}{2158}\text{, if } L_m > S_d \text{ (or equivalently } A_\Delta > 2158/S_d\text{),} \tag{7.13}$$

$$L_m = 2S_d - \frac{2158}{A_\Delta}\text{, if } L_m < S_d \text{ (or equivalently } A_\Delta < 2158/S_d\text{).} \tag{7.14}$$

(2) Sag Curve

$$L_m = \frac{A_\Delta S_d^2}{400 + 3.5S_d}\text{, if } L_m > S_d \text{ (or } A_\Delta > \frac{400 + 3.5S_d}{S_d}\text{)} \tag{7.15}$$

$$L_m = 2S_d - \frac{400 + 3.5S_d}{A_\Delta}\text{, if } L_m < S_d \text{ (or } A_\Delta < \frac{400 + 3.5S_d}{S_d}\text{)} \tag{7.16}$$

Let $L_m(i)$ be the minimum required length of the vertical curve at V_i , $i = 1,.....,n$. Then $L_m(i)$ can be calculated with either eqns (7.13) and (7.14) or eqns (7.15) and (7.16) with $A_\Delta = |g_i - g_{i-1}|$ and $S_d = S_d(i)$. The values of $L_m(i)$'s are treated as constraints in fitting parabolic curves to the set control points V_i 's. Let $L_v(i)$ denote the actual length of vertical curve at V_i . Then

$$L_v(i) \geq L_m(i), \forall i = 1,....,n\ . \tag{7.17}$$

The above equation restricts the length of the vertical curve at each control point V_i to be greater than or equal to the minimum required length. However, it is not the most critical constraint. The most important constraint is to keep the alignment continuous (the second condition in Definition 3.1).

In some situations, the tangent length between any two consecutive control points (say V_i and V_{i+1}) may not be long enough to accommodate the minimum length of vertical curve at V_i and V_{i+1} (see Figure 7.5 for an example). Then a discontinuity occurs, which violates the alignment definition. To avoid such a discontinuity, additional constraints are required.

Given the property of symmetric parabolic curves, half of the parabolic curve at control point V_i will be accommodated in the tangent segment $\overline{V_{i-1}V_i}$, while the other half-length will be in $\overline{V_iV_{i+1}}$, as shown in Figure 7.5.

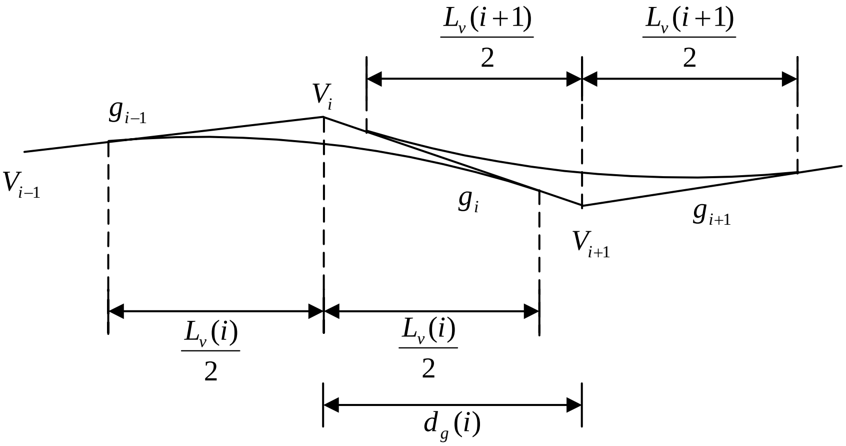


Figure 7.5: An example of discontinuity in vertical curve.

For notational convenience, let $L_v(0) = L_v(n+1) = 0$. Then the following constraint must be satisfied to achieve a continuous vertical alignment:

$$\frac{1}{2}\left(L_v(i) + L_v(i+1)\right) \le d_g(i), \forall i = 0,......, n\,. \tag{7.18}$$

To fit vertical curves while satisfying eqns (7.17) and (7.18), an iterative procedure is required. Without violating the design standard, we may first use minimum length to fit the vertical curves at each control point V_i . If the continuity condition holds, then the resulting vertical alignment will satisfy both eqns (7.17) and (7.18). If a discontinuity occurs (say between V_i and V_{i+1}), then the vertical curves at V_i and V_{i+1} must be shortened so that the continuity condition can hold. For consistency, assume that the deficiencies in the lengths of vertical curves at V_i and V_{i+1} are the same. Then the resulting lengths of vertical curves will be

$$\begin{aligned} L_v(i) &= L_m(i) - \left[\frac{\left(L_m(i) + L_m(i+1)\right)}{2} - d_g(i)\right], \\ &= \frac{L_m(i) - L_m(i+1)}{2} + d_g(i) \end{aligned} \tag{7.19}$$

$$\begin{aligned} L_v(i+1) &= L_m(i+1) - \left[\frac{\left(L_m(i) + L_m(i+1)\right)}{2} - d_g(i)\right]. \\ &= \frac{L_m(i+1) - L_m(i)}{2} + d_g(i) \end{aligned} \tag{7.20}$$

If discontinuities exist in a series of successive tangents, then it is best to first reduce the lengths of vertical curves at the end points of the most critical tangent where the vertical curve lengths are not yet finalized. Based on the updated lengths of vertical curves, the next critical tangent is identified and the lengths of the vertical curve at its two (or one) end point are reduced. This process is continued until all tangents satisfy the continuity condition. We now summarize below the algorithm for determining the lengths of vertical curves:

Algorithm 7.1 Procedure for determining the lengths of vertical curves

(0) Step 0: Variable definition

Variables	Descriptions
D_i	The deficiency in the length of the i^{th} tangent, $i = 0,....,n$
D_c	The deficiency in the length of the most critical tangent
D_t	Temporary deficiency
I_c	The index indicating the most critical tangent
f_i	Boolean variable indicating whether the length of the vertical curve at V_i must be fixed ($f_i = 1$) or not ($f_i = 0$)

```
(1) Step 1: Initialization
    Set $L_v(0) = L_v(n+1) = 0$
    Set $L_v(i) = L_m(i), \forall i = 1,....,n$
    Set $D_i = 0.5\left(L_v(i) + L_v(i+1)\right) - d_g(i)$
    Set $f_0 = f_{n+1} = 1$
    If $L_v(i) = 0$, then set $f_i = 1$, $\forall i = 1,....,n$
    If $L_v(i) > 0$, then set $f_i = 0$, $\forall i = 1,....,n$
    Set $I_c = 0$
(2) Step 2: Identify the most critical tangent
    Set $D_c = 0$
    Set $i = 0$
    2-1  If $i \le n$, then continue 2-2; otherwise go to step 3
    2-2  If $D_i > 0$, then continue; otherwise go to step 2-4
         2-2-1 If $f_i = f_{i+1} = 0$, then
               Set $D_t = D_i / 2$
               If $L_v(i) < 2D_t$, then set $D_t = L_v(i)/2$
               If $L_v(i+1) < 2D_t$, then set $D_t = L_v(i+1)/2$
               Go to step 2-3
         2-2-2 If $f_i = 0$ and $f_{i+1} = 1$, then
                  Set $D_t = D_i$
                  Go to step 2-3
         2-2-3 If $f_i = 1$ and $f_{i+1} = 0$, then
                  Set $D_t = D_i$
                  Go to step 2-3
    2-3  If $D_t > D_c$, then
         Set $D_c = D_t$
         Set $I_c = i$
```

```
      2-4  Set $i = i + 1$; go to step 2-1
(3)   Step 3: Termination rule
      If $D_c = 0$, then stop. Otherwise continue step 4.
(4)   Step 4: Update the lengths of vertical curves and other variables
      4-1  If $f_i = f_{i+1} = 0$, then
           Set $L_v(I_c) = L_v(I_c) - 2D_c$
           Set $L_v(I_c + 1) = L_v(I_c + 1) - 2D_c$
           Set $D_{I_c} = D_{I_c} - 2D_c$
           If $I_c \neq 0$, then set $D_{I_c - 1} = D_{I_c - 1} - D_c$
           If $I_c \neq n$, then set $D_{I_c + 1} = D_{I_c + 1} - D_c$
      If $L_v(I_c) = 0$, then set $f_{I_c} = 1$
      If $L_v(I_c + 1) = 0$, then set $f_{I_c + 1} = 1$
      Go to step 2
      4-2  If $f_i = 0$ and $f_{i+1} = 1$, then
           Set $L_v(I_c) = L_v(I_c) - 2D_c$
           Set $D_{I_c} = 0$
           If $I_c \neq 0$, then set $D_{I_c - 1} = D_{I_c - 1} - D_c$
           Go to step 2
      4-3  If $f_i = 1$ and $f_{i+1} = 0$, then
           Set $L_v(I_c + 1) = L_v(I_c + 1) - 2D_c$
           Set $D_{I_c} = 0$
           If $I_c \neq n$, then set $D_{I_c + 1} = D_{I_c + 1} - D_c$
           Go to step 2
```

The above algorithm endeavors to mimic an engineer’s design process. For a given set of control points in the HZ plane, the algorithm can generate a vertical alignment, consisting of tangents and parabolic curves, which satisfies the definition of vertical alignments. In addition, the resulting alignment will always satisfy the continuity constraint as stated in eqn (7.18). The sight distance constraint (eqn (7.17)) may be violated. To ensure that eqn (7.17) holds for the final alignment, a penalty cost of violation will be added to the objective function, which will be investigated in the next section.

7.2 Cost function

The formulations for the cost components in optimizing non-backtracking 3-dimensional alignment are similar to those developed for Model 1. The only difference is that some estimates needed in calculating alignment costs can now be obtained from the ground profile and vertical alignments. These estimates include earthwork cost and user costs.

7.2.1 Earthwork cost

In optimizing horizontal alignments, earthwork cost was imposed on location-dependent costs due to the lack of information about actual vertical profiles. In optimizing 3-dimensional alignments, however, earthwork cost can be directly estimated.

The method adopted in this study to estimate earthwork volume of a highway project is called "average-end-area" method (Wright, 1996; Garber and Hoel, 1996) and has been used by a majority of state highway agencies in the U.S. In order to compute earth excavation and embankment of a highway project, the cut or fill cross-section areas at some station points along the horizontal alignment are determined. Then the earthwork volume between any two successive station points is computed. Summing up the quantities from each pair of successive station points will give estimates of total earth excavation and embankment volumes for a proposed highway alignment.

The precision of the earth volume estimates depends on the density of station points. The more station points, the better the estimates of earthwork volumes will be. Assume that the station points are equally spaced along the horizontal alignment and the distance between any two consecutive station points is L_s feet measured along the horizontal alignment. Let n_s be the total number of stations, excluding start point S and end point *E*. Then

$$n_s = \begin{cases} \left\lceil \dfrac{L_n}{L_s} \right\rceil^{-} - 1, & \text{if the remainder of } L_n / L_s \text{ is } 0 \\ \left\lceil \dfrac{L_n}{L_s} \right\rceil^{-}, & \text{otherwise} \end{cases}, \qquad (7.21)$$

where L_n = the total length of horizontal alignment, given in (4.26)

$\lceil \cdot \rceil^{-}$ denotes the truncated integer value.

Assume that the i^{th} station point is denoted by S_i, $i = 0,.......,n_s + 1$, where $S_0 = S$ and $S_{n_s+1} = E$ represent the start and end points of the alignment, respectively. Then the distance between S_{n_s} and S_{n_s+1} may be less than L_s because L_n is not necessarily a multiple of L_s. Moreover, n_s is not a fixed number in the proposed solution algorithm. Rather, it is dependent on L_n. For different alignment alternatives, L_n will be different and so is n_s.

The road cross-section at each station S_i, $i = 0,.......,n_s + 1$ can be fill, cut, or neither, depending on the difference between ground and road elevations. Figure 7.6 shows the typical fill and cut sections considered in this study.

In Figure 7.6, W denotes road width, while W_g represents construction width. θ_f and θ_c are corresponding filling and cutting angles. The road and ground elevations are denoted by z_r and z_g respectively. By simple geometry, the cross-section area A_r is then determined by

$$A_r = \frac{1}{2}\left(W + W_g\right)\left|z_g - z_r\right| \qquad (7.22)$$

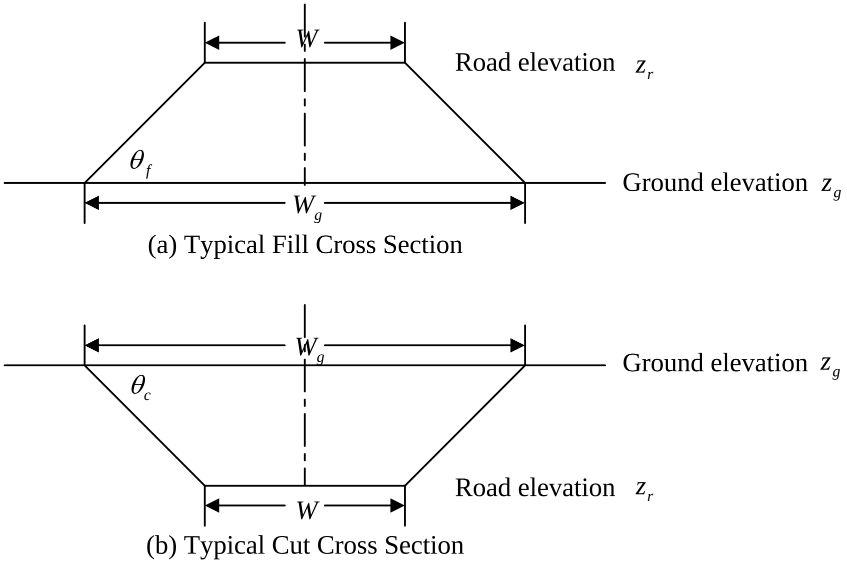


Figure 7.6: Typical fill and cut cross sections.

For fill and cut sections, construction width W_g can be computed with the following two equations:

$$W_g = W + 2(z_r - z_g)\cot\theta_f \text{, if } z_r > z_g \text{ (fill),} \tag{7.23}$$

$$W_g = W + 2(z_g - z_r)\cot\theta_c \text{, if } z_g > z_r \text{ (cut).} \tag{7.24}$$

Substituting them into eqn (7.22), we arrive at

$$\text{If } z_r > z_g \text{ (fill), } A_r = \left[W + (z_r - z_g)\cot\theta_f\right](z_r - z_g)\,; \tag{7.25}$$

$$\text{If } z_g > z_r \text{ (cut), } A_r = \left[W + (z_g - z_r)\cot\theta_c\right](z_g - z_r)\,. \tag{7.26}$$

To apply eqns (7.25) and (7.26), the road and ground elevations at each station point S_i, $i = 0,......., n_s + 1$ must be given. Let $z_g(i)$ and $z_r(i)$ denote the road and ground elevations at S_i. Then $Z_g(i)$ and $z_r(i)$ are determined by the position of S_i on the horizontal alignment. The distance between S_i and $S_0 = S$ (i.e., the alignment start point) along the horizontal alignment, denoted by h_{S_i}, can be obtained with

$$h_{S_i} = iL_s \text{, if } i = 0,......., n_s, \tag{7.27}$$

$$h_{S_i} = L_n \text{, if } i = n_s + 1. \tag{7.28}$$

The determinations of $z_g(i)$ and $z_r(i)$ are based on different logic and will be discussed separately in the following two subsections.

7.2.1.1 Determination of ground elevation

The ground elevation $z_g(i)$ is determined by the location of S_i in the *XY* plane. $z_g(i)$ is then taken as the elevation of the cell where S_i is located. The rules for positioning the *XY* coordinates of S_i, denoted by (x_{S_i}, y_{S_i}), depend on whether S_i is located on a circular curve or a tangent section of the corresponding horizontal alignment. Let h_{C_i} and h_{T_i} denote the distances of C_i and T_i measured along the horizontal alignment, where C_i and T_i are the i^{th} point of curvature and the i^{th} point of tangent whose geometric meanings are presented in Figure 7.2. Then

$$h_{T_i} = 0, \text{ if } i = 0\,, \tag{7.29}$$

$$h_{T_i} = \sum_{k=0}^{i-1} \sqrt{(x_{T_k} - x_{C_{k+1}})^2 + (y_{T_k} - y_{C_{k+1}})^2} + \sum_{k=1}^{i} R_k \Delta_k, \quad \text{if } 1 \le i \le n\,, \tag{7.30}$$

and

$$h_{C_i} = \sum_{k=0}^{i-1} \sqrt{(x_{T_k} - x_{C_{k+1}})^2 + (y_{T_k} - y_{C_{k+1}})^2} + \sum_{k=1}^{i-1} R_k \Delta_k, \quad \text{if } 1 \le i \le n+1\,. \tag{7.31}$$

The circular curve or tangent section where S_i is located can be determined by the following rules:

If $h_{C_k} \le h_{S_i} \le h_{T_k}$, then

S_i is located on the k^{th} circular curve between C_k and T_k. (7.32)

If $h_{T_k} \le h_{S_i} \le h_{C_{k+1}}$, then

S_i is located on the tangent section between T_k and C_{k+1}. (7.33)

The determinations of (x_{S_i}, y_{S_i}) for both situations are given below:

(1) Station point is located on a circular curve

Assume that S_i is located on the k^{th} circular curve (i.e., between C_k and T_k) as shown in Figure 7.7. Let $\theta_{i \cdot k}$ be the angle between $\overrightarrow{\delta_k C_k}$ and $\overrightarrow{\delta_k S_i}$, and $\theta'_{i \cdot k}$ be the angle between $\overrightarrow{\delta_k S_i}$ and $\overrightarrow{\delta_k T_k}$, where δ_k is the center of the k^{th} circular curve. Then

$$\theta_{i \cdot k} = \frac{h_{S_i} - h_{C_k}}{R_k}\,, \tag{7.34}$$

where R_k is the radius of the k^{th} circular curve (ft), obtained with Algorithm 4.1, and

$$\theta'_{i.k} = \Delta_k - \theta_{i\cdot k}\ , \qquad (7.35)$$

where Δ_k is the intersection angle at P_k , given in eqn (4.10).

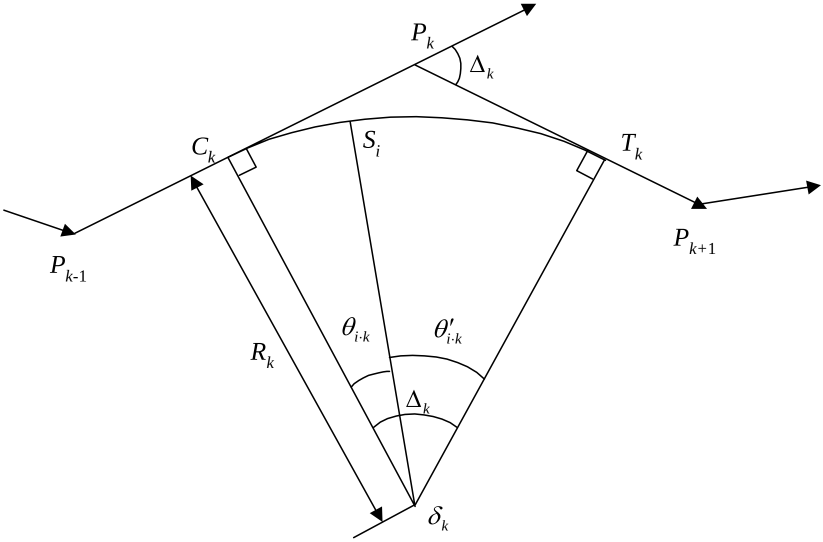


Figure 7.7: Station point on a circular curve of the horizontal alignment.

We also know that

$$\begin{aligned}\cos\theta_{i\cdot k} &= \frac{\overrightarrow{\delta_k C_k}\cdot\overrightarrow{\delta_k S_i}}{\left\|\overrightarrow{\delta_k C_k}\right\|\left\|\overrightarrow{\delta_k S_i}\right\|} = \frac{(\mathbf{C}_k - \boldsymbol{\delta}_k)\cdot(\mathbf{S}_i - \boldsymbol{\delta}_k)}{R_k^2} \\ &= \frac{(x_{C_k} - x_{\delta_k})(x_{S_i} - x_{\delta_k}) + (y_{C_k} - y_{\delta_k})(y_{S_i} - y_{\delta_k})}{R_k^2}\end{aligned}\ , \qquad (7.36)$$

$$\begin{aligned}\cos\theta'_{i\cdot k} &= \frac{\overrightarrow{\delta_k T_k}\cdot\overrightarrow{\delta_k S_i}}{\left\|\overrightarrow{\delta_k T_k}\right\|\left\|\overrightarrow{\delta_k S_i}\right\|} = \frac{(\mathbf{T}_k - \boldsymbol{\delta}_k)\cdot(\mathbf{S}_i - \boldsymbol{\delta}_k)}{R_k^2} \\ &= \frac{(x_{T_k} - x_{\delta_k})(x_{S_i} - x_{\delta_k}) + (y_{T_k} - y_{\delta_k})(y_{S_i} - y_{\delta_k})}{R_k^2}\end{aligned}\ . \qquad (7.37)$$

Rearranging eqns (7.36) and (7.37), we obtain the following linear system:

$$\begin{aligned}(x_{C_k}-x_{\delta_k})(x_{S_i}-x_{\delta_k})+(y_{C_k}-y_{\delta_k})(y_{S_i}-y_{\delta_k})&=R_k^2\cos\theta_{i\cdot k}\\(x_{T_k}-x_{\delta_k})(x_{S_i}-x_{\delta_k})+(y_{T_k}-y_{\delta_k})(y_{S_i}-y_{\delta_k})&=R_k^2\cos\theta'\end{aligned}\;. \tag{7.38}$$

In the above linear system, x_{δ_k} and y_{δ_k} can be obtained from eqn (4.22) and thus the only unknown variables are x_{S_i} and y_{S_i}. Since this is only a 2×2 linear system, we can solve it analytically. The solutions are given as follows:

$$\begin{aligned}x_{S_i}&=x_{\delta_k}+\frac{(y_{T_k}-y_{\delta_k})R_k^2\cos\theta_{i\cdot k}-(y_{C_k}-y_{\delta_k})R_k^2\cos\theta'_{i\cdot k}}{(x_{C_k}-x_{\delta_k})(y_{T_k}-y_{\delta_k})-(x_{T_k}-x_{\delta_k})(y_{C_k}-y_{\delta_k})},\\y_{S_i}&=y_{\delta_k}+\frac{(x_{C_k}-x_{\delta_k})R_k^2\cos\theta'_{i\cdot k}-(x_{T_k}-x_{\delta_k})R_k^2\cos\theta_{i\cdot k}}{(x_{C_k}-x_{\delta_k})(y_{T_k}-y_{\delta_k})-(x_{T_k}-x_{\delta_k})(y_{C_k}-y_{\delta_k})}.\end{aligned} \tag{7.39}$$

(2) Station point is located on a tangent section

The determination of (x_{S_i}, y_{S_i}) is relatively simple if S_i is located on a tangent section rather than a circular curve. Assume that S_i is located on the tangent section between T_k and C_{k+1} as shown in Figure 7.8. Then (x_{S_i}, y_{S_i}) is the point that lies on the vector from T_k to C_{k+1} at the distance $h_{S_i}-h_{T_k}$, i.e.,

$$\begin{bmatrix}x_{S_i}\\ y_{S_i}\end{bmatrix}=\begin{bmatrix}x_{T_k}\\ y_{T_k}\end{bmatrix}+(h_{S_i}-h_{T_k})\frac{\mathbf{C}_{k+1}-\mathbf{T}_k}{\left\|\mathbf{C}_{k+1}-\mathbf{T}_k\right\|}. \tag{7.40}$$

With eqns (7.39) and (7.40), we can easily compute the coordinates of any station points. Once (x_{S_i}, y_{S_i}) are given, the ground elevation $Z_g(i)$ is taken as the elevation of the cell where S_i is located. The indexes of the cell are determined with the following equations:

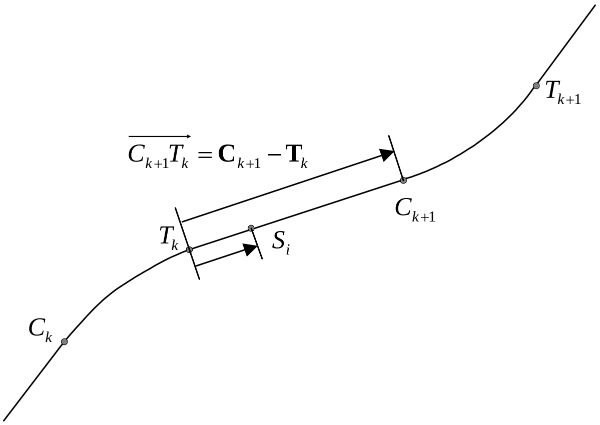


Figure 7.8: Station point on a tangent section of the horizontal alignment.

$$1^{\text{st}} \text{ index: } \left\lceil \frac{x_{S_i} - x_O}{D_x} \right\rceil^{-}, \tag{7.41}$$

$$2^{\text{nd}} \text{ index: } \left\lceil \frac{y_{S_i} - y_O}{D_y} \right\rceil^{-}, \tag{7.42}$$

where $\lceil \cdot \rceil^{-}$ denotes the truncated integer of its argument
(x_O, y_O) are the coordinates of the origin of the study area.

7.2.1.2 Determination of road elevation

The road elevation $z_r(i)$ is determined by the horizontal position of S_i on the vertical profile. A typical vertical profile generated by Algorithm 7.1 is displayed in Figure 7.9. The vertical alignment consists of tangent sections and parabolic curves. Let C_i^V be the i^{th} point of curvature (beginning of the i^{th} parabolic curve) and T_i^V be the i^{th} point of tangency (beginning of the i^{th} tangent section) on a vertical alignment. For notational convenience, we further denote $T_0^V = V_0 = S$ and $C_{n+1}^V = V_{n+1} = E$ as the start and end points of the alignment. Then T_i^V and C_{i+1}^V are linked by a tangent section for all $i = 0,.....,n$, whereas C_i^V and T_i^V are connected by a parabolic curve whose length is $L_v(i)$, for all $i = 1,.....,n.$ In some extreme cases, where the tangent between two control points (say V_i and V_{i+1}) is completely eliminated by the associated parabolic curves, then the point of tangency T_i^V will coincide with the next point of curvature C_{i+1}^V, as, for example, for T_3^V and C_4^V in Figure 7.9.

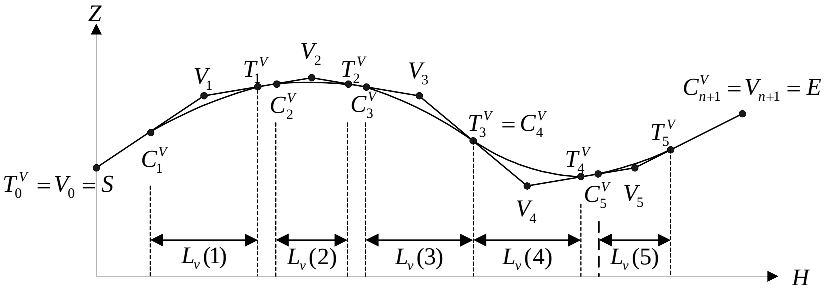


Figure 7.9: An example of vertical alignments.

The rules for determining the road elevation at station point S_i depend on whether S_i is located on a parabolic curve or a tangent section of the corresponding vertical alignment. Let $(h_{C_i^V}, z_{C_i^V})$ and $(h_{T_i^V}, z_{T_i^V})$ denote the HZ coordinates of C_i^V and T_i^V on the HZ plane. Then from the property of symmetric parabolic curves, we obtain

$$h_{C_i^V} = h_{V_i} - 0.5L_v(i), \text{ if } 1 \le i \le n+1, \tag{7.43}$$

$$h_{T_i^V} = h_{V_i} + 0.5L_v(i), \ \text{ if } 0 \le i \le n, \tag{7.44}$$

where $L_v(i)$ is the length of vertical curve at V_i (ft), obtained with Algorithm 7.1, and

$$z_{C_i^V} = z_{V_i} - 0.5L_v(i)\frac{g_{i-1}}{100}, \quad \text{if } 1 \le i \le n+1, \tag{7.45}$$

$$z_{T_i^V} = z_{V_i} + 0.5L_v(i)\frac{g_i}{100}, \quad \text{if } 0 \le i \le n, \tag{7.46}$$

where g_i is the grade (%), given in eqn (7.7).

With eqns (7.45) and (7.46), the parabolic curve or tangent section where S_i is located can be determined by the following rules:

If $h_{C_k^V} \le h_{S_i} \le h_{T_k^V}$, then

S_i is located on the k^{th} parabolic curve between C_k^V and T_k^V. (7.47)

If $h_{T_k^V} \le h_{S_i} \le h_{C_{k+1}^V}$, then

S_i is located on the tangent section between T_k^V and C_{k+1}^V. (7.48)

The road elevations $z_r(i)$ at station point S_i for both situations are given below:

(1) Station point located on a parabolic curve

Assume that S_i is located on the k^{th} parabolic curve (i.e., between C_k^V and T_k^V) as displayed in Figure 7.10. Then from the properties of parabolic curves (see Wright, 1996 and Underwood, 1991), the road elevation at S_i can be obtained with the following equation:

$$z_r(i) = z_{C_k^V} + \frac{g_{k-1}}{100}(h_{S_i} - h_{C_k^V}) + \frac{1}{2L_v(k)}\frac{(g_k - g_{k-1})}{100}(h_{S_i} - h_{C_k^V})^2. \tag{7.49}$$

(2) Station point located on a tangent section

If a station point S_i is located on a tangent section, then the road elevation $z_r(i)$ can be easily obtained by simple trigonometry analysis. Assume that S_i is located on the tangent section between T_k^V and C_{k+1}^V as shown in Figure 7.11. Then $z_r(i)$ can be computed with

$$z_r(i) = z_{T_k^V} + (h_{S_i} - h_{T_k^V})\frac{z_{C_{k+1}^V} - z_{T_k^V}}{h_{C_{k+1}^V} - h_{T_k^V}}. \tag{7.50}$$

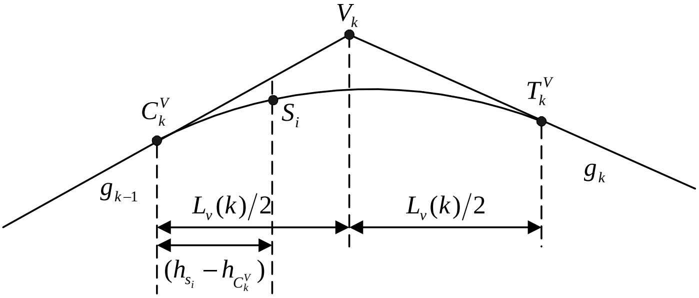


Figure 7.10: Station point on a parabolic curve of the vertical alignment.

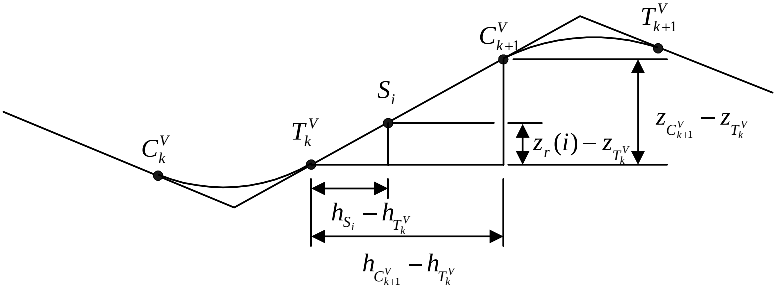


Figure 7.11: Station point on a tangent section of the vertical alignment.

7.2.1.3 Computation of earthwork volume

With the above analysis, we are able to compute $z_g(i)$ (ground elevation) and $z_r(i)$ (road elevation) at all station points S_i, $i = 0,.....,n_s + 1$. Substituting $z_g(i)$ and $z_r(i)$ for z_g and z_r into eqns (7.25) and (7.26), we can obtain the cross-section area at each control point S_i, $i = 0,.....,n_s + 1$. Let $A_r(i)$ be the resulting cross-section area at S_i. Further assume that the ground profile is smooth between any two successive station points S_i and S_{i+1}. Then the earthwork volume (excavation and embankment) between S_i and S_{i+1}, $i = 0,....,n_s$ can be computed according to the following situations:

(1) If $z_g(i) \geq z_r(i)$ and $z_g(i+1) \geq z_r(i+1)$ (i.e., for excavation only), then

$$E_c(i) = \frac{1}{2}\big(A_r(i) + A_r(i+1)\big)L_s\,, \tag{7.51}$$

$$E_f(i) = 0\,, \tag{7.52}$$

where $E_c(i)$ and $E_f(i)$ denote the volumes of earth excavation and embankment between V_i and V_{i+1} (cu ft).

L_s = the horizontal distance between station points S_i and S_{i+1}. If $i = n_s$, then L_s must be replaced by $L_n - h_{S_{n_s}}$.

(2) If $z_g(i) \le z_r(i)$ and $z_g(i+1) \le z_r(i+1)$ (i.e., for embankment only), then

$$E_c(i) = 0\,, \tag{7.53}$$

$$E_f(i) = \frac{1}{2}\big(A_r(i) + A_r(i+1)\big)L_s\,. \tag{7.54}$$

(3) If $z_g(i) > z_r(i)$ and $z_g(i+1) < z_r(i+1)$ (i.e., for excavation and then embankment)

$$E_c(i) = \frac{1}{2}\frac{z_g(i) - z_r(i)}{z_g(i) - z_r(i) + z_r(i+1) - z_g(i+1)}A_r(i)L_s\,, \tag{7.55}$$

$$E_f(i) = \frac{1}{2}\frac{z_r(i+1) - z_g(i+1)}{z_g(i) - z_r(i) + z_r(i+1) - z_g(i+1)}A_r(i+1)L_s\,. \tag{7.56}$$

(4) If $z_g(i) < z_r(i)$ and $z_g(i+1) > z_r(i+1)$ (i.e., for embankment and then excavation)

$$E_c(i) = \frac{1}{2}\frac{z_r(i) - z_g(i)}{z_r(i) - z_g(i) + z_g(i+1) - z_r(i+1)}A_r(i)L_s\,, \tag{7.57}$$

$$E_f(i) = \frac{1}{2}\frac{z_g(i+1) - z_r(i+1)}{z_r(i) - z_g(i) + z_g(i+1) - z_r(i+1)}A_r(i+1)L_s\,. \tag{7.58}$$

Eqns (7.51) to (7.54) can be easily understood, but eqns (7.55) to (7.58) are more complex and thus should be clarified. Both equations are derived similarly. Here, only the derivation of eqns (7.55) and (7.56) is presented. Figure 7.12 graphically illustrates eqns (7.55) and (7.56). From the similarity of triangles, we know that

$$\frac{l_c}{z_g(i) - z_r(i)} = \frac{l_f}{z_r(i+1) - z_g(i+1)}\,, \tag{7.59}$$

and thus

$$l_c = \frac{z_g(i) - z_r(i)}{z_r(i+1) - z_g(i+1)}l_f\,. \tag{7.60}$$

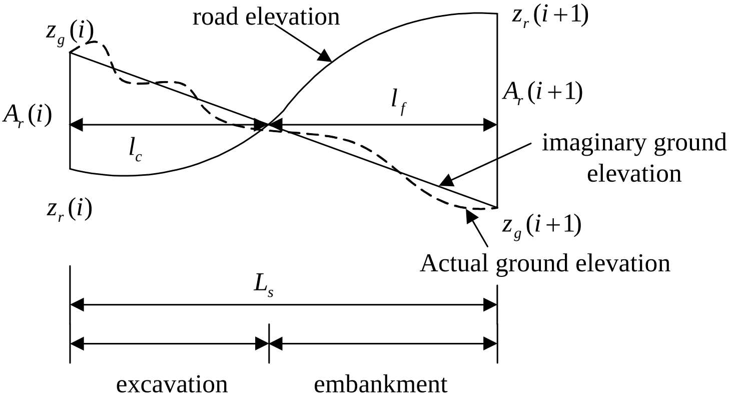


Figure 7.12: Graphical illustration of a change from cut to fill sections.

Since $l_c + l_f = L_s$, we obtain

$$\frac{z_g(i) - z_r(i) l_f}{z_r(i+1) - z_g(i+1)} + l_f = L_s \,. \tag{7.61}$$

After rearranging, we arrive at

$$l_f = \frac{z_r(i+1) - z_g(i+1)}{z_g(i) - z_r(i) + z_r(i+1) - z_g(i+1)} L_s \,. \tag{7.62}$$

Hence,

$$l_c = \frac{z_g(i) - z_r(i)}{z_g(i) - z_r(i) + z_r(i+1) - z_g(i+1)} L_s \,. \tag{7.63}$$

With eqns (7.62) and (7.63), we now can easily derive the earth excavation and embankment volumes, as shown in eqns (7.55) and (7.56).

The total earth excavation and embankment quantities for a proposed alignment are then computed by summing up individual volume from each pair of successive station points. After converting the units from cubic feet to cubic yards, we obtain

$$E_C = \frac{1}{27} \sum_{i=0}^{n} E_c(i) \,, \tag{7.64}$$

$$E_F = \frac{1}{27}\sum_{i=0}^{n} E_f(i)\,, \tag{7.65}$$

where E_C and E_F are total earth excavation and embankment (cu yd).

7.2.1.4 Estimation of earthwork cost

In highway design, one major way to minimize the cost of earthwork is to balance (i.e., equalize) the volumes of cuts and fills. However, when earth is excavated and hauled to form an embankment, the material may be compacted and the final volume may be less than its original quantity. This phenomenon is called "shrinkage". The amount of shrinkage varies with the soil type and the depth of the fill. Let s_e denote the earth shrinkage factor. Then the net earthwork volume will be

$$E_N = E_C s_e - E_F\,. \tag{7.66}$$

If the net earthwork E_N is positive, then the extra earth must be shipped to a landfill. If E_N is negative, then the deficiency must be supplemented from a borrow pit. Let K_l be the transportation cost for moving one cubic yard of earth to a landfill, and K_b represent the transportation cost for moving one cubic yard of earth from a borrow pit. Further denote K_C and K_F as cutting and filling cost per cubic yard. Then the total earthwork cost for an alignment alternative can be computed as

$$C_E = K_C E_C + K_F E_F + K_l \max\{E_N, 0\} - K_b \min\{E_N, 0\}, \tag{7.67}$$

where C_E is total earthwork cost (\$), including excavation, embankment, and transshipment cost.

7.2.2 User costs

One of the key factors in estimating user costs is average running speed, through which we can compute vehicle operating costs and travel time costs. In section 4.6, we have already presented a formula to estimate average running speed based on various geometric design features and traffic conditions (see eqns (4.46) to (4.48)). In optimizing horizontal alignments, the average hilliness $\overline{H}$ used in eqns (4.46) to (4.48) is ignored due to lack of information about vertical profiles. In optimizing 3-dimensional alignments, however, $\overline{H}$ can be determined from the corresponding vertical alignments.

The calculation of $\overline{H}$ is given in eqn (4.39). To compute $\overline{H}$, the vertical distance between adjacent sag and crest must be determined. Using the same notation, we find that

(1) If $g_i > g_{i-1}$ and $g_{i+1} < g_i$ (sag and then crest), then

$$h_i = z_r(V_{i+1}) - z_r(V_i)\,, \tag{7.68}$$

where h_i = vertical distance between the i^{th} sag and the following $(i+1)^{th}$ crest $z_r(V_i)$ and $z_r(V_{i+1})$ are road elevations at control points V_i and V_{i+1}, and can be computed following the procedures in section 7.2.1.2.

(2) If $g_i < g_{i-1}$ and $g_{i+1} > g_i$ (crest and then sag), then

$$h_i = z_r(V_i) - z_r(V_{i+1})\,. \tag{7.69}$$

With eqns (7.68), (7.69), (4.38) to (4.40), and (4.46) to (4.48), we are able to estimate average running speed for a 3-dimensional alignment, which will be further used for computing vehicle operating costs and travel time costs. For detailed discussion and formulas, please refer to sections 4.6.1 and 4.6.2.

7.3 Final model and its properties

In this section, the final model formulations for optimizing non-backtracking 3-dimensional alignments are presented. We will first discuss the constraints that must be considered in this model and then covert them into penalty functions. Finally the penalty costs will be incorporated into the objective function.

Recall that the decision variables in this model include the abscissa d_i and the ordinate z_i of each intersection point P_i on its associated vertical cutting plane. The domain constraints of d_i's are defined in such a way that P_i's are restricted within the study region, as shown in eqns (4.4) to (4.7). However, the ordinates z_i's do not have any domain constraints. Instead, z_i's are restricted by the gradient limitations required by highway designs. The maximum grade depends on the design speed and the surrounding topography, and is usually treated as a design control parameter. Let $G_{\max}$ denote the maximum allowable gradient (%). Then the gradient constraint can be expressed as

$$|g_i| \le G_{\max}, \forall i = 0,.........,n\,, \tag{7.70}$$

where g_i = the road grade at tangent segment $\overline{V_iV_{i+1}}$, obtained with eqn (7.7).

As shown in section 4.6.3, the minimum radius constraints are incorporated into the objective function by adding penalties to accident costs. In addition, the horizontal alignment generated by Algorithm 4.1 will satisfy the continuity constraint. For vertical alignment, the continuity constraint is also satisfied. However, the minimum required sight distance may be violated. Taking all these into account, the problem can be formulated as

Model 3 – Model for optimizing non-backtracking 3-dimensional alignments

$$\underset{d_1,z_1,d_2,z_2,\ldots\ldots,d_n,z_n}{\text{Minimize}} \quad C_T = C_N + C_L + C_U + C_E \,, \tag{7.71}$$

$$\text{subject to} \quad d_{iL} \le d_i \le d_{iU} \,, \forall i = 1,\ldots\ldots\ldots,n \,, \tag{7.72}$$

$$|g_i| \le G_{\max}, \forall i = 0,\ldots\ldots\ldots,n \,, \tag{7.73}$$

$$L_v(i) \ge L_m(i), \forall i = 1,\ldots\ldots,n \,, \tag{7.74}$$

where the horizontal alignment is produced with Algorithm 4.1 while the vertical alignment is generated with Algorithm 7.1

C_T = total cost ($)

C_N = location-dependent cost ($), computed with Algorithms 4.2 and 4.3

C_L = length-dependent cost ($), given in eqn (4.35)

C_U = user cost ($), given in eqn (4.71) with updated average running speed

C_E = earthwork cost ($), obtained with eqn (7.67)

d_{iL} and d_{iU} are lower and upper bounds of the i^{th} abscissa, given in eqns (4.4) to (4.7)

$L_v(i)$ = actual length of vertical curve at the i^{th} control point V_i

$L_m(i)$ = minimum required length of vertical curve at V_i

(d_i, z_i) = the coordinates of the intersection points on the i^{th} vertical cutting plane.

In the optimization process, penalty functions representing constraints (7.73) and (7.74) will be added to the objective function. Let $C_g(i)$ denote the penalty cost for violating gradient constraint at the i^{th} tangent. Then $C_g(i)$ is given the following function form:

$$C_g(i) = \begin{cases} \alpha_0 + \alpha_1 \left(|g_i| - G_{\max} \right)^{\alpha_2} & \text{if } |g_i| > G_{\max} \\ 0 & \text{if } |g_i| \le G_{\max} \end{cases}, \qquad \forall i = 0,\ldots\ldots,n \,, \tag{7.75}$$

where α_0 = penalty cost when gradient constraint is violated

α_1 and α_2 ($\alpha_2 > 1$) are user-specific coefficients.

The total penalty cost for violating gradient constraints, denoted by C_G, is then obtained by the summation of individual penalty cost at each tangent. That is

$$C_G = \sum_{i=0}^{n} C_g(i) \,. \tag{7.76}$$

Similarly, the penalty cost for violating the minimum length of vertical curve is computed as

$$C_m(i) = \begin{cases} \beta_0 + \beta_1 \left(L_m(i) - L_v(i)\right)^{\beta_2} & \text{if } L_v(i) < L_m(i) \\ 0 & \text{if } L_v(i) \geq L_m(i) \end{cases}, \quad \forall i = 1, \ldots\ldots, n\,, \tag{7.77}$$

where $C_m(i)$ = penalty cost for violating the minimum length of vertical curve at V_i

β_0 = fixed penalty cost coefficient for violating the minimum length of vertical curve

β_1 and β_2 ($\beta_2 > 1$) are user-specified coefficients.

Hence,

$$C_M = \sum_{i=1}^{n} C_m(i)\,, \tag{7.78}$$

where C_M = total penalty cost for violating minimum length of vertical curve.

Adding the above penalty costs into the objective function, we may rewrite Model 3 as:

$$\underset{d_1, z_1, d_2, z_2, \ldots\ldots, d_n, z_n}{\text{Minimize}} \quad C_T = C_N + C_L + C_U + C_E + C_G + C_M\,, \tag{7.79}$$

$$\text{subject to} \quad d_{iL} \leq d_i \leq d_{iU}\,,\ \forall i = 1, \ldots\ldots\ldots, n\,. \tag{7.80}$$

The penalty function approach allows the constraints to be violated slightly during the search. The magnitudes of penalty coefficients α_i and β_i determine the tradeoff between constraint violations and other costs. The selection of α_i and β_i depends on the terrain and the functional categories of the highway. Larger coefficients tend to force the final solutions to satisfy the constraints as closely as the users wish.

The objective function defined in the above model includes most of the important costs considered in highway design. The design constraints, including minimum radius, maximum gradient, and minimum length of vertical curve are all taken into account by penalizing the violations of constraints. Moreover, the horizontal alignment generated by Algorithm 4.1 and the vertical alignment produced by Algorithm 7.1 will hold the continuity condition (defined in eqn (3.2)) and the first continuously differentiable condition (defined in eqn (3.3)). In addition, the necessary conditions of alignments defined in eqns (3.13) and (3.14) are also satisfied. To complete the model, we now need an efficient search algorithm to solve it. In the next section, we will define an appropriate genetic encoding scheme and operators to perform the search.

7.4 Genetic encoding and initial population

In Model 3, each intersection point is determined by two variables, the abscissa and ordinate on the associated vertical cutting plane. For an alignment with n intersection points, the encoded solution will consist of $2n$ genes. Therefore, the chromosome is defined as

$$\Lambda = [\lambda_1, \lambda_2, \lambda_3, \lambda_4, ..., \lambda_{2n-1}, \lambda_{2n}] = [d_1, z_1, d_2, z_2, ..., d_n, z_n], \tag{7.81}$$

where Λ = chromosome

λ_i = the i^{th} gene, for all $i = 1,.......,2n$

(d_i, z_{i_i}) = the coordinates of intersection point on the i^{th} vertical cutting plane, for all $i = 1,.......,n$.

It can be seen that the mappings between the genes in a chromosome and the coordinates of the intersection points are

$$\lambda_{2i-1} = d_i, \ \forall i = 1,.......,n, \tag{7.82}$$

$$\lambda_{2i} = z_i, \ \forall i = 1,.......,n. \tag{7.83}$$

The alleles of odd genes in a chromosome will be limited to the range of the corresponding abscissa. That is

$$d_{iL} \le \lambda_{2i-1} \le d_{iU}, \forall\, i = 1,.....,n. \tag{7.84}$$

Eqn (7.84) is the domain constraint of Model 3 as shown in eqn (7.80). The constraint will be always satisfied throughout the solution algorithm by restricting the mutation range of the genes.

To maintain a large variety in the gene pool, the initial population of intersection points includes the following 3 categories:

(1) Intersection points lying on the straight line connecting the start and end points

In this case, the set of intersection points represents a straight alignment, which reduces length-dependent cost to a minimum. The chromosome is defined as

$$\Lambda = [\lambda_1, \lambda_2, ..., \lambda_{2n-1}, \lambda_{2n}] = [0, z_{O_1}, 0, z_{O_2}, ..., 0, z_{O_n}], \tag{7.85}$$

where z_{O_i} = the Z coordinates of the origin of the abscissa on the i^{th} vertical cut.

z_{O_i} in the above equation is simply determined by

$$z_{O_i} = z_S + i\frac{(z_E - z_S)}{n+1}, \tag{7.86}$$

where z_S and z_E are the Z coordinates of the start and end point of the alignment.

(2) Intersection points lying randomly on the vertical cuts with random elevations

In this case, the odd genes of the chromosomes (i.e., d_i 's) are randomly generated from continuous uniform distributions defined by the boundaries of the corresponding vertical cuts:

$$\lambda_{2i-1} = r_c[d_{iL}, d_{iU}], \ \forall i = 1,......., n. \tag{7.87}$$

The even genes (i.e., z_i 's) are randomly generated from the ranges determined by the gradient constraints. Note that given the odd genes, the X and Y coordinates of the decoded intersection points can be determined by eqn (7.1), and thus the corresponding horizontal alignment can be obtained with Algorithm 4.1. Consequently, the distance along the horizontal alignment between any two successive control points V_i and V_{i+1} can be computed with eqn (7.6). Recall that the i^{th} even gene represents the elevation at V_i, namely $\lambda_{2i} = z_i = z_{V_i}$. Assume that the even genes are generated from the first one (i.e., λ_2) to the last one (i.e., λ_{2n}). Then the range of the i^{th} ($i = 1,....,n$) even gene, denoted by the interval $[z_{iL}, z_{iU}]$, is determined according to the following equations:

$$z_{iL} = \max\left\{\left(z_{V_{i-1}} - d_g(i-1)\frac{G_{\max}}{100}\right), \left(z_{V_{n+1}} - \sum_{k=i}^{n} d_g(k)\frac{G_{\max}}{100}\right)\right\}, \tag{7.88}$$

$$z_{iU} = \min\left\{\left(z_{V_{i-1}} + d_g(i-1)\frac{G_{\max}}{100}\right), \left(z_{V_{n+1}} + \sum_{k=i}^{n} d_g(k)\frac{G_{\max}}{100}\right)\right\}, \tag{7.89}$$

where z_{V_i} = the elevation at V_i

$z_{V_{n+1}}$ = the elevation at V_{n+1}

$d_g(i-1)$ = the distance between control points V_{i-1} and V_i along the horizontal alignment, computed from eqn (7.6)

$G_{\max}$ = maximum allowable gradient (%).

Hence, we get

$$\lambda_{2i} = r_c\left[z_{iL}, z_{iU}\right], \forall i = 1,...., n. \tag{7.90}$$

Note that the first term on the right-hand side of eqn (7.88) is the lower bound of the elevation at V_i determined by the gradient constraint from V_{i-1} to V_i, while the second term is the lower bound dominated by the gradient constraint from V_{n+1} to V_i. The final lower bound z_{iL} is then taken as the one that is larger. Similarly, the final upper bound z_{iU} is determined by the allowable upper bounds from V_{i-1} to V_i or from V_{n+1} to V_i, whichever is smaller.

(3) Intersection points lying randomly on the vertical cuts with elevations possibly close to the existing ground elevations

This population type is similar to the previous category. The odd genes of the chromosomes are generated by eqn (7.87). The even genes, which represent the elevations of intersection points, are set as close as possible to the existing ground elevations at the corresponding control points V_i 's, but should be within the allowable range calculated by eqns (7.88) and (7.89). The ground elevation at the i^{th} control point V_i, denoted by $z_g(i)$, is determined by the XY coordinates of V_i. In the format of study region, $z_g(i)$ is the elevation of the cell whose indexes are computed by eqns (7.41) and (7.42). If $z_g(i) < z_{iL}$, then λ_{2i} is set to z_{iL}. If $z_g(i) > z_{iU}$, then λ_{2i} is set to z_{iU}. If $z_{iL} < z_g(i) < z_{iU}$, then λ_{2i} is taken as $z_g(i)$.

According to the discussions in section 5.7, the population size is set to be proportional to the number of decision variables. In Model 3, if n intersection points are used to represent the alignment, the total number of genes in a chromosome will be $2n$. Thus, a population size $n_p = 10n$ is recommended.

7.5 Genetic operators

Eight different types of genetic operators are employed in solving Model 3. The first four are mutation-based operators, while the last four operators are crossover-based. The operators discussed in this section are similar to those developed for Models 1 and 2. However, in order to take the effects of 3-dimensional alignments into account, several modifications are made to facilitate the search. To fit the nature of the problem, all operators are intentionally designed to work on the decoded intersection points rather than a single encoded gene. We will now briefly discuss each operator in turn.

7.5.1 Uniform mutation

For a given chromosome $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_{2n-1}, \lambda_{2n}]$, if the k^{th} intersection point is selected for uniform mutation, where $k = r_d[1, n]$, then λ_{2k-1} will be replaced by

$$\lambda'_{2k-1} = r_c[d_{kL}, d_{kU}]. \qquad (7.91)$$

Since the allowable ranges of elevations at intersection points depend on the horizontal alignment, the other encoded gene of the k^{th} intersection point (i.e., λ_{2k}), which represents the elevation, will not be changed until the new horizontal alignment is determined. As mentioned in section 5.5, a curve elimination procedure is required to prevent a horizontal alignment from getting trapped at a local optimum. Hence, λ_{2k} must be changed after applying this procedure. The procedure for Model 3 is shown below:

Algorithm 7.2 Curve elimination procedure for Model 3

(1) Step 1: Encode S and E into the chromosome

Expand the chromosome $\Lambda = [\lambda_1, \lambda_2, ..., \lambda'_{2k-1}, \lambda_{2k}, ..., \lambda_{2n-1}, \lambda_{2n}]$ to

$$\Lambda = [\lambda_{-1}, \lambda_0, \lambda_1, ..., \lambda'_{2k-1}, \lambda_{2k}, ..., \lambda_{2n}, \lambda_{2n+1}, \lambda_{2n+2}]$$
$$= [0, z_S, \lambda_1, ..., \lambda'_{2k-1}, \lambda_{2k}, ..., \lambda_{2n}, 0, z_E]$$

(2) Step 2: Generate two independent random loci i and j

$$i = r_d[0, k-1] \text{ and } j = r_d[k+1, n+1]$$

(3) Step 3: Change the values of the first encoded genes of the intermediate intersection points between P_i and P_k

$$\lambda'_{2l-1} = \lambda_{2i-1} + (l-i)\frac{(\lambda'_{2k-1} - \lambda_{2i-1})}{k-i}, \text{ for all } l = i+1, ..., k-1 \tag{7.92}$$

(4) Step 4: Change the values of the intermediate genes between the k^{th} and j^{th} genes

$$\lambda'_{2l-1} = \lambda_{2k-1} + (l-k)\frac{(\lambda_{2j-1} - \lambda'_{2k-1})}{j-k}, \text{ for all } l = k+1, ..., j-1 \tag{7.93}$$

(5) Step 5: Remove the encoded genes of S and E from the chromosome

Truncate the resulting chromosome

$\Lambda' = [0, z_S, \lambda_1, ..., \lambda'_{2i+1}, \lambda_{2i+2}, ..., \lambda'_{2k-1}, \lambda_{2k}, ..., \lambda'_{2j-3}, \lambda_{2j-2}, ..., \lambda_{2n}, 0, z_E]$ to

$\Lambda' = [\lambda_1, ..., \lambda'_{2i+1}, \lambda_{2i+2}, ..., \lambda'_{2k-1}, \lambda_{2k}, ..., \lambda'_{2j-3}, \lambda_{2j-2}, ..., \lambda_{2n}]$

The set of odd genes $\lambda_1, \lambda_3, ..., \lambda'_{2i+1}, ..., \lambda'_{2k}, ..., \lambda'_{2j-3}, ..., \lambda_{2n}$ after applying Algorithm 7.2 will now represent a new horizontal alignment. The alleles of the even genes $\lambda_{2l}, l = i+1, ..., j-1$, which are the elevations of intersection points, will be changed according to different rules depending on whether the gradient constraint is satisfied between V_i and V_j. If the gradient constraint is violated, there is no room to adjust elevations at the control points between V_i and V_j since the allowable mutation ranges for elevations are all nil. In such a case, the even genes $\lambda_{2l}, l = i+1, ..., j-1$ are set on the straight line connecting V_i and V_j on the HZ plane. On the contrary, if the gradient constraint is satisfied, then the rule for mutating the even genes is determined by a randomly generated binary digit $r_d[0,1]$.

(1) The random binary digit $r_d[0,1] = 0$

In this case, the alleles of even genes are set as close as possible to the ground elevations of their corresponding control points, which is similar to the way we generate the third type of initial population. The idea behind this strategy is that the earthwork cost can be minimized if a highway is built right on the existing ground elevations. However, the constraint for minimum required lengths of vertical curves may be violated, which leads to a high penalty cost.

(2) The random binary digit $r_d[0,1] = 1$

In this case, the alleles of the even genes $\lambda_{2l}, l = i+1, \ldots\ldots, j-1$ are changed from the first one (i.e., λ_{2i}) to the last one (i.e., λ_{2j}), while satisfying the gradient constraints. This is similar to the way of generating the even genes for the second type of initial population mentioned previously. Although this strategy may result in a higher earthwork cost due to the increase in the difference between ground elevations and road elevations, it may reduce the penalty costs for violating the minimum length constraints for vertical curves since it is more likely to generate a satisfactory alignment.

The aforementioned procedure for determining the alleles of even genes of a selected chromosome can be considered a counterpart of the "curve elimination procedure", which deals with horizontal alignments. For convenience, we call it an "elevation determination procedure" because it assigns the elevations of the intersection points, which eventually determines the corresponding vertical alignment. Since this procedure is also used in other operators, we now summarize it as follows:

Algorithm 7.3 Elevation determination procedure for Model 3

(1) Step 1: Generate the horizontal alignment

Use the set of odd genes to generate a horizontal alignment by applying Algorithm 4.1

(2) Step 2: Check for violations of the gradient constraint between V_i and V_j

If $\left| (\lambda_{2j} - \lambda_{2i}) \Big/ \sum_{k=i}^{j-1} d_g(k) \right| > G_{\max}/100$, then go to step 3;

otherwise, go to step 4.

(3) Step 3: Set the alleles of the even genes on the straight line connecting V_i and V_j on the HZ plane

$$\lambda'_{2l} = \lambda_{2i} + \sum_{k=i}^{l-1} d_g(k) \frac{(\lambda_{2j} - \lambda_{2i})}{\sum_{k=i}^{j-1} d_g(k)}, \text{ for all } l = i+1, \ldots\ldots, j-1 \tag{7.94}$$

Stop

(4) Step 4: Select strategy

Generate a random binary digit $r_d[0,1]$.

```
(5) Step 5: Set the alleles of the even genes according to $r_d[0,1]$
    5-1 Set $l = i+1$
    5-2 If $l \le j-1$, then continue step 5-3. Otherwise stop
    5-3 Calculate allowable mutation range for $\lambda_{2l}$
```

$$z_{lL} = \max\left\{\left(z_{V_{l-1}} - d_g(l-1)\frac{G_{\max}}{100}\right), \left(z_{V_{n+1}} - \sum_{k=i}^{n} d_g(k)\frac{G_{\max}}{100}\right)\right\} \tag{7.95}$$

$$z_{lU} = \min\left\{\left(z_{V_{l-1}} + d_g(l-1)\frac{G_{\max}}{100}\right), \left(z_{V_{n+1}} + \sum_{k=i}^{n} d_g(k)\frac{G_{\max}}{100}\right)\right\} \tag{7.96}$$

```
    5-4 If $r_d[0,1] = 0$, then set $\lambda_{2l}$ as close as possible to the ground elevation
        5-4-1 If $z_g(l) < z_{lL}$, then $\lambda'_{2l} = z_{lL}$
        5-4-2 If $z_{lL} \le z_g(l) \le z_{lU}$, then $\lambda'_{2l} = z_g(l)$
        5-4-3 If $z_{lU} < z_g(l)$, then $\lambda'_{2l} = z_{lU}$
        5-4-4 Set $l = l+1$; go to step 5-2
    5-5 If $r_d[0,1] = 1$, then generate $\lambda_{2l}$ from the allowable mutation range
```

$$\lambda'_{2l} = r_c[z_{lL}, z_{lU}] \tag{7.97}$$

```
        Set $l = l+1$; go to step 5-2
```

For a given chromosome, the uniform mutation operator changes the alleles of the selected odd gene according to eqn (7.91). Then we apply Algorithm 7.1 to eliminate horizontal curves. For the new horizontal alignment, Algorithm 7.2 is further applied to determine the alleles of corresponding even genes.

7.5.2 Straight mutation

Let $\Lambda = [\lambda_1, \lambda_2, \ldots\ldots, \lambda_{2n-1}, \lambda_{2n}]$ denote the chromosome to be mutated. We randomly generate two discrete numbers i and j, where $i = r_d[0, n+1]$, $j = r_d[0, n+1]$, $i \ne j$, and $i < j$. For notational convenience, if $i = 0$ (which represents the start point S), we set $\lambda_{2i-1} = \lambda_{-1} = 0$ and $\lambda_{2i} = \lambda_0 = z_S$. Similarly, if $j = n+1$ (which denotes the end point E), we set $\lambda_{2j-1} = \lambda_{2n+1} = 0$ and $\lambda_{2j} = \lambda_{2n+2} = z_E$. Then the alleles of the intermediate genes between the $(2i)^{th}$ and $(2j-1)^{th}$ genes will be replaced by

$$\lambda'_{2l-1} = \lambda_{2i-1} + (l-i)\frac{(\lambda_{2j-1} - \lambda_{2i-1})}{j-i}, \text{ for all } l = i+1, \ldots\ldots, j-1, \tag{7.98}$$

$$\lambda'_{2l} = \lambda_{2i} + (l-i)\frac{(\lambda_{2j} - \lambda_{2i})}{j-i}, \text{ for all } l = i+1, \ldots\ldots, j-1. \tag{7.99}$$

Note that the set of new intermediate genes represents a straight line connecting the i^{th} and the j^{th} intersection points in 3-dimensional space. The corresponding intersection points of these new genes are placed at equal spacing.

7.5.3 Non-uniform mutation

Let $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_{2n-1}, \lambda_{2n}]$ be the chromosome for non-uniform mutation at the k^{th} intersection point, where $k = r_d[1, n]$. We first generate a random binary digit $r_d[0,1]$. Then the allele of the corresponding odd gene λ'_{2k-1} is determined by the following rules:

$$\begin{array}{l} \text{If the random digit } r_d[0,1] = 0 \text{, then } \lambda'_{2k-1} = \lambda_{2k-1} - f(t, \lambda_{2k-1} - d_{kL}) . \\ \text{If the random digit } r_d[0,1] = 1 \text{, then } \lambda'_{2k-1} = \lambda_{2k-1} + f(t, d_{kU} - \lambda_{2k-1}) . \end{array} \quad (7.100)$$

where f is defined in eqn (5.14).

In order to prevent the resulting alignment from sticking at a local optimum, Algorithm 7.1 is applied to eliminate curves between two randomly generated intersection points (say P_i and P_j). Then the corresponding even genes of the intersection points between P_i and P_j, which represent the elevations of P_l, $l = i+1, ..., j-1$, are mutated according to Algorithm 7.3.

7.5.4 Whole non-uniform mutation

This operator applies the eqn (7.100) to each odd gene of a given chromosome in a randomly generated sequence. This is equivalent to changing the overall configuration of the decoded horizontal alignment. Then Algorithm 7.3 with $i = 0$ and $j = n+1$ is further applied to determine the alleles of the even genes for the selected chromosome.

7.5.5 Simple crossover

Given two parents $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, ..., \lambda_{i(2n)}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, ..., \lambda_{j(2n)}]$, if the genes are crossed after position $2k$, where $k = r_d[1, n]$, then the resulting offspring are

$$\Lambda'_i = [\lambda_{i1}, \lambda_{i2}, ..., \lambda_{i(2k)}, \lambda_{j(2k+1)}, ..., \lambda_{j(2n)}] , \quad (7.101)$$

$$\Lambda'_j = [\lambda_{j1}, \lambda_{j2}, ..., \lambda_{j(2k)}, \lambda_{i(2k+1)}, ..., \lambda_{i(2n)}] . \quad (7.102)$$

7.5.6 Two-point crossover

Let $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, ..., \lambda_{i(2n)}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, ..., \lambda_{j(2n)}]$ denote the two parents to be crossed between two randomly generated positions $2k$ and $2l$, where $k = r_d[1, n]$, $l = r_d[1, n]$, $k \neq l$, and $k < l$. Then the resulting offspring are

$$\Lambda'_i = [\lambda_{i1}, \lambda_{i2}, ..., \lambda_{i(2k)}, \lambda_{j(2k+1)}, ..., \lambda_{j(2l)}, \lambda_{i(2l+1)}, ..., \lambda_{in}] , \quad (7.103)$$

$$\Lambda'_j = [\lambda_{j1}, \lambda_{j2}, ..., \lambda_{j(2k)}, \lambda_{i(2k+1)}, ..., \lambda_{i(2l)}, \lambda_{j(2l+1)}, ..., \lambda_{jn}] . \quad (7.104)$$

7.5.7 Arithmetic crossover

Let $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots\ldots, \lambda_{i(2n)}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots\ldots, \lambda_{j(2n)}]$ be the two parents to be crossed. The operator generates the following two offspring:

$$\Lambda_i' = \omega\Lambda_i + (1-\omega)\Lambda_j , \tag{7.105}$$

$$\Lambda_j' = \omega\Lambda_j + (1-\omega)\Lambda_i , \tag{7.106}$$

where $\omega = r_c[0,1]$.

7.5.8 Heuristic crossover

Given two parents $\Lambda_i = [\lambda_{i1}, \lambda_{i2}, \ldots\ldots\ldots, \lambda_{i(2n)}]$ and $\Lambda_j = [\lambda_{j1}, \lambda_{j2}, \ldots\ldots\ldots, \lambda_{j(2n)}]$, where we assume $C_T(\Lambda_i) \le C_T(\Lambda_j)$ (i.e., Λ_i is better or at least as good as Λ_j), then the operator generates a single offspring Λ' according to the following rule:

$$\Lambda' = \omega\left(\Lambda_i - \Lambda_j\right) + \Lambda_i , \tag{7.107}$$

where $\omega = r_c[0,1]$.

If the offspring violates the domain constraints given in eqn (7.80), a new offspring is regenerated and the domain constraints are checked again. If after a predetermined number of attempts, no new offspring can meet the constraints, the operator gives up and returns Λ_i as the offspring.

Chapter 8

Model and solution algorithms for optimizing backtracking 3-dimensional alignments

In this chapter, we present a model that optimizes not only 3-dimensional alignments, but also backtracking alignments, which is not possible with the models developed in the previous chapters. The proposed model can be regarded as an extension to Model 2 as well as Model 3, and is the most general of these models. In fact, except for Model 0, which is the base model formulated by the parametric representation of alignments, all other models presented in the previous chapters are only special cases of the proposed model. The formulation approach for describing a backtracking 3-dimensional alignment is presented in the next section. The cost function is briefly discussed in section 8.2, followed by the final model and its properties. The genetic encoding of the chromosomes and the initial population are discussed in section 8.4. Finally, we develop appropriate genetic operators to facilitate the work of the search algorithm.

8.1 Representation of alignment

The approach for describing a backtracking 3-dimensional alignment is to fit the alignment to a set of intersection points in 3-dimensional space. Unlike Model 3, where the intersection points are restricted to vertical cutting planes, the intersection points in the proposed model can be anywhere in the region of interest. Assume that n intersection points are used to describe the alignment and let $P_i(x_{P_i}, y_{P_i})$ be the i^{th} intersection point. Then given the set of points P_i, $i = 0,......., n+1,$ where $P_0 = S$ and $P_{n+1} = E,$ the corresponding horizontal alignment is obtained with Algorithm 4.1. Recall from chapter 7 that the elevations of the middle points of circular curves are controlled by the Z coordinates of the set of intersection points P_i's. The corresponding vertical alignment can be then generated by Algorithm 7.1.

Note that in Model 3, the location of an intersection point is determined by 2 decision variables. The intersection points in the proposed model require 3 decision variables (i.e., the X, Y, and Z coordinates) to be precisely located. Since intersection points can be everywhere in the study region, their X and Y coordinates must be limited to the following feasible ranges:

$$x_O \le x_{P_i} \le x_{\max}, \quad \forall\, i = 1, \ldots\ldots, n, \tag{8.1}$$

$$y_O \le y_{P_i} \le y_{\max}, \quad \forall\, i = 1, \ldots\ldots, n, \tag{8.2}$$

where (x_O, y_O) = the X, Y coordinates of the bottom-left corner of the study region

(x_{P_i}, y_{P_i}) = the X, Y coordinates of intersection point P_i

$(x_{\max}, y_{\max})$ = the X, Y coordinates of the top-right corner of the study region.

The Z coordinates of intersection points do not have any domain constraints. Rather, they are restricted by the constraints for gradients and minimum lengths of vertical curves as discussed in section 7.3.

Note that the alignment represented by the above general modeling approach can be backtracking or non-backtracking, depending on the positions of the set of intersection points. The configuration of the optimized alignment for a highway design project depends on the complexity of the geographic and topographic space, and the relative importance of various components of the cost function.

8.2 Cost function

The components in the cost function of a 3-dimensional alignment can be categorized into location-dependent, length-dependent, earthwork, and user costs. The detailed formulations of each cost component are presented in sections 4.3 to 4.6 and 7.2.

8.2.1 Location-dependent cost

Location-dependent costs include costs for land acquisition, soil stabilization, environmental impact, etc, provided the information is available. The location-dependent costs for tangent sections and circular curves of an alignment alternative can be obtained with Algorithms 4.2 and 4.3.

8.2.2 Length-dependent cost

Length-dependent costs are those costs proportional to the length of a highway alignment, such as pavement cost, maintenance cost and VMT-based environmental cost. Length-dependent cost is computed by multiplying total alignment length by unit-length-dependent cost. For detailed formulation of length-dependent costs, please refer to section 4.5.

8.2.3 Earthwork cost

In optimizing 3-dimensional alignments, earthwork cost can be directly estimated rather than included in location-dependent costs. The earthwork volume of a highway project is estimated by "average-end-area" method. The total earthwork cost consists of earth excavation, embankment, and transportation costs for shipping extra earth from a borrow pit or to a landfill. The detailed methods for computing earthwork cost are given in section 7.2.

8.2.4 User costs

User costs include fuel-consumption cost, travel time cost, and accident cost. The key factor in estimating vehicle operating costs and travel time costs is the average running speed of the proposed alignment. For a 3-dimensional alignment, the estimation of average running speed must involve the information of the vertical profile, as discussed in section 7.3. The detailed calculations of user costs are presented in section 4.6.

8.3 Final model and its properties

The objective function of the proposed model is the summation of each of the cost components mentioned in the previous sections, plus the penalty costs for the violations of the constraints for gradients and minimum lengths of vertical curves. The decision variables are the coordinates of the set of intersection points. Thus, the model formulation can be expressed as:

Model 4 – Model for optimizing backtracking horizontal alignments

$$\underset{x_{P_1}, y_{P_1}, z_{P_1}, \ldots\ldots, x_{P_n}, y_{P_n}, z_{P_n}}{\text{Minimize}} \quad C_T = C_N + C_L + C_U + C_E + C_G + C_M \,, \tag{8.3}$$

$$\text{subject to} \quad x_O \le x_{P_i} \le x_{\max}, \quad \forall\, i = 1, \ldots.., n \,, \tag{8.4}$$

$$y_O \le y_{P_i} \le y_{\max}, \quad \forall\, i = 1, \ldots.., n \,, \tag{8.5}$$

where the horizontal alignment is produced with Algorithm 4.1 while the vertical alignment is generated with Algorithm 7.1

C_T = total cost ($)

C_N = location-dependent cost ($), determined with Algorithms 4.2 and 4.3

C_L = length-dependent cost ($), given in eqn (4.35)

C_U = user cost ($), given in eqn (4.71) with updated average running speed

C_E = earthwork cost ($), obtained with eqn (7.67)

C_G = penalty costs ($) for violating gradient constraints, given in eqn (7.76)

C_M = penalty costs ($) for violating minimum lengths of vertical curves, obtained with eqn (7.78)

(x_O, y_O) = the X, Y coordinates of the bottom-left corner of the study region

(x_{P_i}, y_{P_i}) = the X, Y coordinates of intersection point P_i

$(x_{\max}, y_{\max})$ = the X, Y coordinates of the top-right corner of the study region.

Except for the decision variables and the capability of locating a backtracking alignment, Model 4 duplicates Model 3, presented in section 7.3. For evaluating the objective function in Model 3, the coordinates of intersection points on vertical cutting planes must be transformed into the Cartesian coordinate system, which is not necessary for Model 4. Note that if n intersection points are employed for both models, then the dimension of the search space will be $2n$ for Model 3 and $3n$ for Model 4. From this standpoint, these two models are quite different.

Model 4 has several good properties. First, it considers most dominating and sensitive costs associated with a highway alternative. Important constraints such as minimum radius, maximum gradient, minimum lengths of vertical curves are considered as well. Second, the resulting alignment satisfies boundary conditions given in eqns (3.11) and (3.12), the continuity condition (eqn (3.2)) and the first continuously differentiable condition (eqn (3.3)). Third, it can simultaneously optimize 3-dimensional alignments with possible backward bends, while avoiding inaccessible regions. Finally, the search space is continuous, which complies with the nature of the problem. If an efficient solution algorithm is developed for Model 4, then one can see that Model 4 will satisfy all the necessary conditions of a good model for optimizing highway alignments, as defined in section 2.7.

The proposed solution algorithm for Model 4 is also based on the concept of Genetic Algorithms. However, it is much more complex than those developed for the previous 3 models due to the additional dimensions of the search space. In the next section, we will discuss the genetic encoding and the initial population for Model 4, based on which appropriate genetic operators must be developed to perform the search.

8.4 Genetic encoding and initial population

In Model 4, each intersection point is determined by three decision variables, namely its *X*, *Y*, and *Z* coordinates. For an alignment represented by n intersection points, the encoded chromosome is composed of $3n$ genes. Thus, the chromosome is defined as

$$\Lambda = [\lambda_1, \lambda_2, \lambda_3, \ldots\ldots, \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}] = [x_{P_1}, y_{P_1}, z_{P_1}, \ldots\ldots, x_{P_n}, y_{P_n}, z_{P_n}], \qquad (8.6)$$

where Λ = chromosome

λ_i = the i^{th} gene, for all $i = 1, \ldots\ldots, 3n$

$(x_{P_i}, y_{P_i}, z_{P_i})$ = the coordinates of the i^{th} intersection point, for all $i = 1,.......,n$.

The mappings between the genes in a chromosome and the coordinates of the intersection points are

$$\lambda_{3i-2} = x_{P_i}\ ,\ \forall i = 1,.......,n\ , \tag{8.7}$$

$$\lambda_{3i-1} = y_{P_i}\ ,\ \forall i = 1,.......,n\ , \tag{8.8}$$

$$\lambda_{3i} = z_{P_i}\ ,\ \forall i = 1,.......,n\ . \tag{8.9}$$

The alleles of λ_{3i-2} and λ_{3i-1}, $\forall i = 1,.......,n$ will be limited to the feasible range of the corresponding X and Y coordinates, namely

$$x_O \le \lambda_{3i-2} \le x_{\max}, \quad \forall\, i = 1,.....,n\ , \tag{8.10}$$

$$y_O \le \lambda_{3i-1} \le y_{\max}, \quad \forall\, i = 1,.....,n\ . \tag{8.11}$$

The above equations are the domain constraints of Model 4 and will be always satisfied throughout the solution algorithm by restricting the mutation range of the genes.

Five types of solutions are included in the initial population to increase the varieties of genes.

(1) Intersection points lying on the straight line connecting the start and end points

The individual in this type of population represents a straight alignment, which reduces length-dependent cost to a minimum. Assume that the intersection points are placed on the line segment connecting S and E at equal distance, and let O_i be the i^{th} points numbered from S (O_i is also the origin of the abscissa on the i^{th} vertical cut for Model 3). Then the coordinates of O_i, $\forall i = 1,....,n$ are determined by

$$\begin{bmatrix} x_{O_i} \\ y_{O_i} \\ z_{O_i} \end{bmatrix} = \begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} + \frac{i}{n+1} \begin{bmatrix} x_E - x_S \\ y_E - y_S \\ z_E - z_S \end{bmatrix},\ \forall i = 1,...,n\ , \tag{8.12}$$

where $(x_{O_i}, y_{O_i}, z_{O_i})$ = the coordinates of O_i
(x_S, y_S, z_S) = the coordinates of S
(x_E, y_E, z_E) = the coordinates of E.

The chromosome is then defined as

$$\Lambda = [\lambda_1, \lambda_2, \lambda_3 ...., \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}] = [x_{O_1}, y_{O_1}, z_{O_1}, ..., x_{O_n}, y_{O_n}, z_{O_n}]. \tag{8.13}$$

(2) Intersection points lie randomly on the vertical cuts with random elevations

In this case, the first two encoded genes of an intersection point are generated by the coordinate transformation of a random number from the interval defined by the boundaries of the corresponding vertical cut:

$$\lambda_{3i-2} = x_{O_i} + r_c[d_{iL}, d_{iU}] \cos\theta, \forall i = 1,...., n, \tag{8.14}$$

$$\lambda_{3i-1} = y_{O_i} + r_c[d_{iL}, d_{iU}] \sin\theta, \forall i = 1,...., n, \tag{8.15}$$

where $r_c[d_{iL}, d_{iU}]$ = random number from a continuous uniform distribution whose domain is within the interval $[d_{iL}, d_{iU}]$

d_{iL} and d_{iU} are lower bound and upper bound of the i^{th} vertical cut, as defined in eqns (4.4) to (4.7)

θ = the angle of vertical cuts, given in eqn (4.3).

Given λ_{3i-2} and λ_{3i-1}, $\forall i = 1,...., n$, the corresponding horizontal alignment can be obtained with Algorithm 4.1. The third encoded genes λ_{3i}, $\forall i = 1,...., n$, which carry the elevation information of the set of intersection points P_i's, are generated from the first one (i.e., λ_3) to the last one (i.e., λ_{3n}), while satisfying the gradient constraints. The feasible range of λ_{3i}, denoted by the interval $[z_{iL}, z_{iU}]$, is determined according to eqns (7.88) and (7.89). Then, λ_{3i}, $\forall i = 1,...., n$ are generated by

$$\lambda_{3i} = r_c[z_{iL}, z_{iU}], \forall i = 1,...., n. \tag{8.16}$$

(3) Intersection points lying randomly on the vertical cuts with elevations as close as possible to the existing ground elevations

This type of population is similar to the previous category. For an individual in this population, the encoded genes λ_{3i-2} and λ_{3i-1}, $\forall i = 1,...., n$ are generated by eqns (8.14) and (8.15). The third encoded genes λ_{3i}, $\forall i = 1,...., n$ are set as close as possible to the existing ground elevations of the corresponding control points V_i's, but should be within the allowable range given in eqns (7.88) and (7.89).

(4) Intersection points scattered randomly within the study region with random elevations

This type of population carries useful information for backtracking alignments since the intersection points can be everywhere in the study region. The first two encoded genes λ_{3i-2} and λ_{3i-1}, $\forall i = 1,...., n$ of intersection points P_i 's are defined as follows:

$$\lambda_{3i-2} = r_c[x_O, x_{\max}], \forall i = 1,....,n, \tag{8.17}$$

$$\lambda_{3i-1} = r_c[y_O, y_{\max}], \forall i = 1,....,n. \tag{8.18}$$

The third encoded genes λ_{3i} , $\forall i = 1,....,n$ are generated by the same rules stated for the second population type.

(5) Intersection points scattered randomly within the study region with elevations as close as possible to the existing ground elevations

The encoded genes λ_{3i-2} and λ_{3i-1} , $\forall i = 1,....,n$ of the individuals in this population are generated similarly to the fourth population type to carry possible information for backtracking alignments. The third encoded genes λ_{3i}, $\forall i = 1,....,n$ are taken as close as possible to the existing ground elevations with the same rules used for the third population type.

As in other models presented earlier, the population size is set to be proportional to the number of decision variables to reflect the complexity of the search space. If n intersection points are used to represent the alignment, the total number of genes in a chromosome will be $3n$. Therefore, a population size $n_p = 15n$ is recommended.

8.5 Genetic operators

The genetic operators employed for Model 4 and the logic behind each operator are the same as in Model 3. Each operator is designed to work on the decoded intersection points rather than individual genes to conform to the problem. Note that each intersection point in Model 3 is encoded with 2 genes, whereas three continuous genes are needed to represent an intersection point in Model 4. Due to this difference, several modifications are made to the genetic operators for Model 4. The operators are briefly discussed below:

8.5.1 Uniform mutation

Let $\Lambda = [\lambda_1, \lambda_2, \lambda_3,......, \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}]$ be the chromosome to be mutated at the encoded genes of the k^{th} intersection point, where $k = r_d[1, n]$, Then λ_{3k-2} and λ_{3k-1} will be replaced by

$$\lambda'_{3k-2} = r_c[x_O, x_{\max}], \tag{8.19}$$

$$\lambda'_{3k-1} = r_c[y_O, y_{\max}]. \tag{8.20}$$

Next, a curve elimination procedure is applied to the chromosome to prevent the resulting alignment from getting trapped at a local optimum. The procedure is summarized as follows:

Algorithm 8.1 Curve elimination procedure for Model 4

(1) Step 1: Encode S and E into the chromosome
Expand the chromosome
$\Lambda = [\lambda_1, \lambda_2, \lambda_3 .... \lambda'_{3k-2}, \lambda'_{3k-1}, \lambda_{3k}, ..., \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}]$ to
$\Lambda = [\lambda_{-2}, \lambda_{-1}, \lambda_0, \lambda_1, ..., \lambda'_{3k-2}, \lambda'_{3k-1}, \lambda_{3k}, ... \lambda_{3n}, \lambda_{3n+1}, \lambda_{3n+2}, \lambda_{3n+3}]$
$= [x_S, y_S, z_S, \lambda_1, ..., \lambda'_{3k-2}, \lambda'_{3k-1}, \lambda_{3k}, ... \lambda_{3n}, x_E, y_E, z_E]$

(2) Step 2: Generate two independent random loci i and j

$$i = r_d[0, k-1] \text{ and } j = r_d[k+1, n+1]$$

(3) Step 3: Change the values of the first and second encoded genes of the intermediate intersection points between P_i and P_k

$$\lambda'_{3l-2} = \lambda_{3i-2} + (l-i)\frac{(\lambda'_{3k-2} - \lambda_{3i-2})}{k-i}, \text{ for all } l = i+1, ..., k-1 \tag{8.21}$$

$$\lambda'_{3l-1} = \lambda_{3i-1} + (l-i)\frac{(\lambda'_{3k-1} - \lambda_{3i-1})}{k-i}, \text{ for all } l = i+1, ..., k-1 \tag{8.22}$$

(4) Step 4: Change the values of the intermediate genes between the k^{th} and j^{th} genes

$$\lambda'_{3l-2} = \lambda_{3k-2} + (l-k)\frac{(\lambda_{3j-2} - \lambda'_{3k-2})}{j-k}, \text{ for all } l = k+1, ..., j-1 \tag{8.23}$$

$$\lambda'_{3l-1} = \lambda_{3k-1} + (l-k)\frac{(\lambda_{3j-1} - \lambda'_{3k-1})}{j-k}, \text{ for all } l = k+1, ..., j-1 \tag{8.24}$$

(5) Step 5: Remove the encoded genes of S and E from the chromosome
Truncate the resulting chromosome
$\Lambda' = [x_S, y_S, z_S, \lambda_1, ..., \lambda'_{3i+1}, \lambda'_{3i+2}, \lambda_{3j+3} ...... \lambda'_{3j-5}, \lambda'_{3j-4}, \lambda_{3j-3}, ..., \lambda_{3n}, x_E, y_E, z_E]$
to $\Lambda' = [\lambda_1, ..., \lambda'_{3i+1}, \lambda'_{3i+2}, \lambda_{3j+3} ...... \lambda'_{3j-5}, \lambda'_{3j-4}, \lambda_{3j-3}, ..., \lambda_{3n}]$

Recall from Model 3 that the “curve elimination procedure” is immediately followed by an “elevation determination procedure” to mutate the encoded genes of the elevations of the intersection points P_l, $\forall l = i+1, ..., j-1$. For Model 4, the procedure is to change the alleles of λ_{3l}, $\forall l = i+1, ..., j-1$ according to the following rules:

Algorithm 8.2 Elevation determination procedure for Model 4

(1) Step 1: Generate the horizontal alignment
Use the first two encoded genes λ_{3i-2} and λ_{3i-1}, $\forall i = 1, ..., n$ to generate a horizontal alignment by applying Algorithm 4.1

(2) Step 2: Check for violations of the gradient constraint between V_i and V_j

If $\left| (\lambda_{2j} - \lambda_{2i}) \Big/ \sum_{k=i}^{j-1} d_g(k) \right| > G_{\max}/100$, then go to step 3;

otherwise, go to step 4.

(3) Step 3: Set the alleles of the third encoded genes λ_{3l} on the straight line connecting V_i and V_j on the HZ plane

$$\lambda'_{3l} = \lambda_{3i} + \sum_{k=i}^{l-1} d_g(k) \frac{(\lambda_{3j} - \lambda_{3i})}{\sum_{k=i}^{j-1} d_g(k)}, \text{ for all } l = i+1, \ldots\ldots, j-1 \tag{8.25}$$

Stop

(4) Step 4: Select strategy

Generate a random binary digit $r_d[0,1]$.

(5) Step 5: Set the alleles of the third genes according to $r_d[0,1]$

5-1 Set $l = i+1$

5-2 If $l \le j-1$, then continue to step 5-3. Otherwise stop

5-3 Calculate allowable mutation range for λ_{3l}

$$z_{lL} = \max\left\{ \left(z_{V_{l-1}} - d_g(l-1)\frac{G_{\max}}{100} \right), \left(z_{V_{n+1}} - \sum_{k=i}^{n} d_g(k)\frac{G_{\max}}{100} \right) \right\} \tag{8.26}$$

$$z_{lU} = \min\left\{ \left(z_{V_{l-1}} + d_g(l-1)\frac{G_{\max}}{100} \right), \left(z_{V_{n+1}} + \sum_{k=i}^{n} d_g(k)\frac{G_{\max}}{100} \right) \right\} \tag{8.27}$$

5-4 If $r_d[0,1] = 0$, then set λ_{3l} as close as possible to the ground elevation

5-4-1 If $z_g(l) < z_{lL}$, then $\lambda'_{3l} = z_{lL}$

5-4-2 If $z_{lL} \le z_g(l) \le z_{lU}$, then $\lambda'_{3l} = z_g(l)$

5-4-3 If $z_{lU} < z_g(l)$, then $\lambda'_{3l} = z_{lU}$

5-4-4 Set $l = l+1$; go to step 5-2

5-5 If $r_d[0,1] = 1$, then generate λ_{3l} from the allowable mutation range

$$\lambda'_{3l} = r_c[z_{lL}, z_{lU}] \tag{8.28}$$

Set $l = l+1$; go to step 5-2

8.5.2 Straight mutation

Let $\Lambda = [\lambda_1, \lambda_2, \lambda_3, \ldots\ldots, \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}]$ be the chromosome subjected to mutation. We randomly generate two independent discrete random numbers *i* and j, where $i = r_d[0, n+1]$, $j = r_d[0, n+1]$, $i \neq j$, and $i < j$. Then the intermediate genes between the $(3i)^{th}$ and $(3j-2)^{th}$ will be replaced by

$$\lambda'_{3l-2} = \lambda_{3i-1} + (l-i)\frac{(\lambda_{3j-2} - \lambda_{3i-2})}{j-i}, \text{ for all } l = i+1,......, j-1, \tag{8.29}$$

$$\lambda'_{3l-1} = \lambda_{3i-1} + (l-i)\frac{(\lambda_{3j-1} - \lambda_{3i-1})}{j-i}, \text{ for all } l = i+1,......, j-1, \tag{8.30}$$

$$\lambda'_{3l} = \lambda_{3i} + (l-i)\frac{(\lambda_{3j} - \lambda_{3i})}{j-i}, \text{ for all } l = i+1,......, j-1. \tag{8.31}$$

In the above equations, if $i = 0$ (which represents the start point S), then we set $\lambda_{3i-2} = \lambda_{-2} = x_S$, $\lambda_{3i-1} = \lambda_{-1} = y_S$, and $\lambda_{3i} = \lambda_0 = z_S$. Similarly if $j = n+1$ (which denotes the end point E), then $\lambda_{3j-2} = \lambda_{3n+1} = x_E$, $\lambda_{3j-1} = \lambda_{3n+2} = y_E$, and $\lambda_{3j} = \lambda_{3n+3} = z_E$. Note that the operator places all intermediate intersection points between P_i and P_j at equal distances on the line segment connecting these two points.

8.5.3 Non-uniform mutation

Let $\Lambda = [\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}]$ be the chromosome to be mutated at the encoded genes of the k^{th} intersection point, where $k = r_d[1, n]$. We first generate two random binary digit $r_d[0,1]$. Then the alleles of λ'_{3k-2} and λ'_{3k-1} in the resulting offspring $\Lambda' = [\lambda_1, \lambda_2, \lambda_3, ..., \lambda'_{3k-2}, \lambda'_{3k-1}, \lambda_{3k}, ..., \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}]$ are determined by the following rules:

(1) If the first random digit $r_d[0,1] = 0$, then

$$\lambda'_{3k-2} = \lambda_{3k-2} - f(t, \lambda_{3k-2} - x_O), \tag{8.32}$$

$$\lambda'_{3k-2} = \lambda_{3k-2} + f(t, x_{\max} - \lambda_{3k-2}). \tag{8.33}$$

where f is defined in eqn (5.14).

(2) If the second random digit $r_d[0,1] = 0$, then

$$\lambda'_{3k-1} = \lambda_{3k-1} - f(t, \lambda_{3k-1} - y_O), \tag{8.34}$$

$$\lambda'_{3k-1} = \lambda_{3k-1} + f(t, y_{\max} - \lambda_{3k-1}). \tag{8.35}$$

After mutating λ_{3k-2} and λ_{3k-1}, Algorithm 8.1 is applied to eliminate curves between two randomly generated intersection points (say P_i and P_j). Then the corresponding genes λ_{3l}, $\forall l = i+1, ..., j-1$ of the intersection points between P_i and P_j are mutated according to Algorithm 8.2.

8.5.4 Whole non-uniform mutation

This operator applies the non-uniform operator to each intersection point of a given chromosome in a randomly generated sequence to change the entire configuration of the corresponding horizontal alignment. Then Algorithm 8.2 with $i = 0$ and $j = n + 1$ is further applied to determine the elevations of the intersection points.

8.5.5 Simple crossover

Let two parents $\Lambda_i = [\lambda_{i1},......,\lambda_{i(3n)}]$ and $\Lambda_j = [\lambda_{j1},......,\lambda_{j(3n)}]$ be crossed after a randomly generated position $3k$, where $k = r_d[1,n]$. Then the resulting offspring are

$$\Lambda'_i = [\lambda_{i1},\lambda_{i2},\lambda_{i3},...,\lambda_{i(3k)},\lambda_{j(3k+1)}...,\lambda_{j(3n-2)},\lambda_{j(3n-1)},\lambda_{j(3n)}], \tag{8.36}$$

$$\Lambda'_j = [\lambda_{j1},\lambda_{j2},\lambda_{j3},...,\lambda_{j(3k)},\lambda_{i(3k+1)}...,\lambda_{i(3n-2)},\lambda_{i(3n-1)},\lambda_{i(3n)}]. \tag{8.37}$$

8.5.6 Two-point crossover

Let $\Lambda_i = [\lambda_{i1},.....,\lambda_{i(3n)}]$ and $\Lambda_j = [\lambda_{j1},......,\lambda_{j(3n)}]$ be the two parents to be crossed between two randomly generated positions $3k$ and $3l$, where $k = r_d[1,n]$, $l = r_d[1,n]$, $k \neq l$, and $k < l.$ The resulting offspring are

$$\Lambda'_i = [\lambda_{i1},\lambda_{i2},\lambda_{i3},...,\lambda_{i(3k)},\lambda_{j(3k+1)}...,\lambda_{j(3l)},\lambda_{i(3l+1)},...,\lambda_{i(3n)}], \tag{8.38}$$

$$\Lambda'_j = [\lambda_{j1},\lambda_{j2},\lambda_{j3},...,\lambda_{j(3k)},\lambda_{i(3k+1)}...,\lambda_{i(3l)},\lambda_{j(3l+1)},...,\lambda_{j(3n)}]. \tag{8.39}$$

8.5.7 Arithmetic crossover

Given two parents $\Lambda_i = [\lambda_{i1},.....,\lambda_{i(3n)}]$ and $\Lambda_j = [\lambda_{j1},......,\lambda_{j(3n)}]$, the arithmetic crossover reproduces two offspring as follows

$$\Lambda'_i = \omega\Lambda_i + (1-\omega)\Lambda_j, \tag{8.40}$$

$$\Lambda'_j = \omega\Lambda_j + (1-\omega)\Lambda_i. \tag{8.41}$$

where $\omega = r_c[0,1]$.

8.5.8 Heuristic crossover

Let $\Lambda_i = [\lambda_{i1},.....,\lambda_{i(3n)}]$ and $\Lambda_j = [\lambda_{j1},......,\lambda_{j(3n)}]$ be the two parents to be crossed by this operator, where we assume $C_T(\Lambda_i) \leq C_T(\Lambda_j)$ (i.e., Λ_i is at least as good as Λ_j). Then the operator generates a single offspring Λ' according to the following rule:

$$\Lambda' = \omega(\Lambda_i - \Lambda_j) + \Lambda_i, \tag{8.42}$$

where $\omega = r_c[0,1]$.

This operator may generate an infeasible offspring. In such a case, another random number is generated to create new offspring. If after several user-defined trials no new offspring can satisfy the boundary constraints defined in eqns (8.4) and (8.5), the operator gives up and returns Λ_i as the offspring.

Chapter 9

Case study and sensitivity analysis

In this chapter, we design several examples to demonstrate the models and the solution algorithms for simultaneously optimizing 3-dimensional alignments. In the first test example, a sensitivity analysis is also conducted to investigate the influences of different types of genetic operators on the resulting solution and computation time. The performance and properties of the proposed solution algorithms are also explored with this test example. This provides a further insight into the proposed GAs.

The second and third examples are designed to compare the effectiveness and efficiencies of Models 3 and 4. The second example has a non-backtracking optimal alignment, while the third one has a backtracking optimal alignment. The last two examples illustrate the applicability and capability of Models 3 and 4 in different situations.

9.1 Case study 1

The first test example is designed to demonstrate the performance of the proposed solution algorithm and conduct sensitivity analysis of genetic operators. First, the region of interest for optimizing a 3-dimensional alignment is described. Second, the solution found by the proposed algorithm is presented. A methodology is also introduced to test the goodness of the solution. Finally, the influences of genetic operators on the resulting solution and computation time are explored. A statistical approach is employed to conduct the sensitivity analysis.

9.1.1 Problem description

The first example for running a 3-dimensional model is presented in Figure 9.1. In the region of interest, the X and Y coordinates range from 500 to 6500 and from 500 to 4500, respectively. The region is divided into equal cells of $200\,\text{ft} \times 200\,\text{ft}$. The shade of a cell indicates its elevation. The darker the shade, the higher the elevation. The contours and 3D view of the map are displayed in

Figure 9.2 and Figure 9.3, respectively. It can be seen that there are two small hills in the study region. Moreover, the two fields in the top and the bottom-right corner of Figure 9.1 represent residential areas. The location-dependent costs in these two fields are relatively high because of high demolition, relocation, and land acquisition costs. In between the two small hills, there is another field next to the residential area. This field indicates poor soil conditions, and thus results in a high location-dependent cost.

The *XYZ* coordinates of the start and end points of the alignment are (800,1200,100) and (5000,3950,120) respectively. The problem is to find a 3-dimensional alignment with minimal total costs. As we can expect, the alignment should skirt the hill and avoid high cost cells to minimize the total cost. The design parameters for the proposed highway are summarized in Table 9.1.

Table 9.1: Design parameters for test example 1.

Design Parameter	Design Value
Design speed	50 mph
Side friction coefficient	0.16
Superelevation	0.06
Maximum grade	5%
Forward friction coefficient	0.28
AADT	8000 vehicles per day

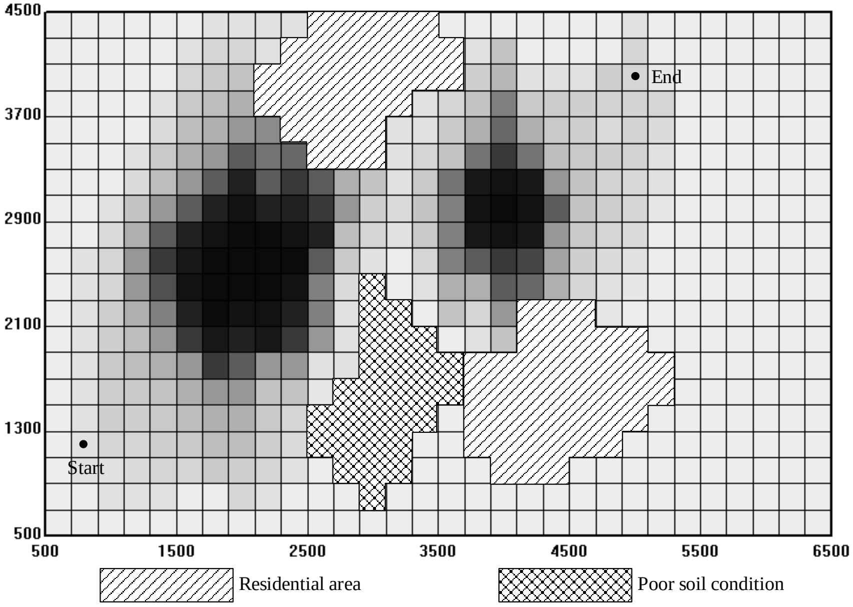


Figure 9.1: Map for test example 1.

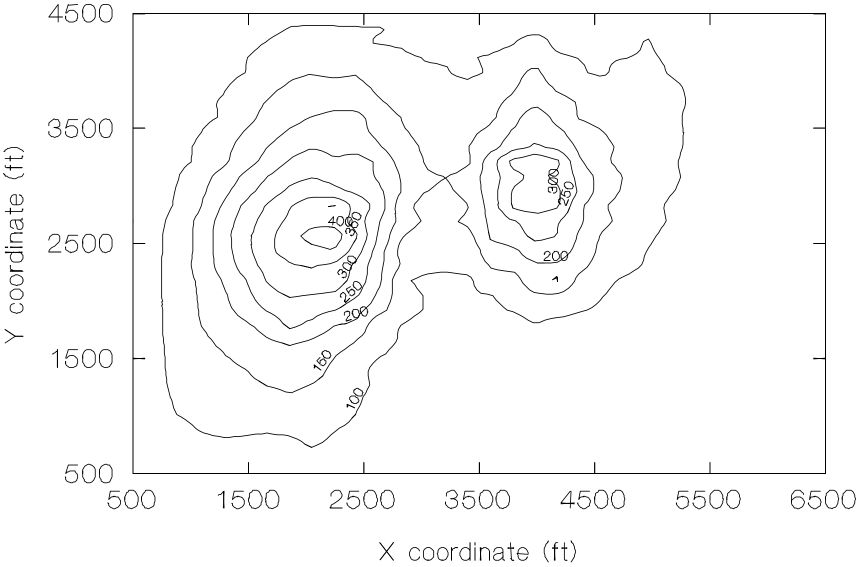


Figure 9.2: Contours of test example 1.

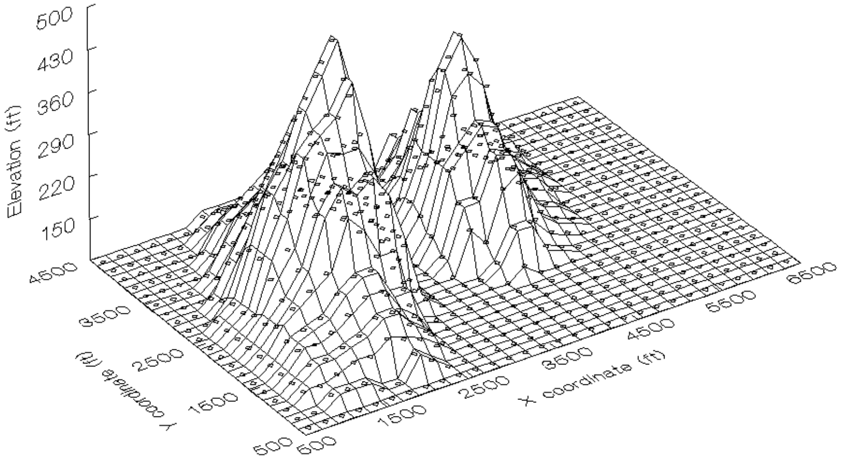


Figure 9.3: 3D view of test example 1.

9.1.2 Solution and goodness test

Since the terrain is not extremely difficult, we may expect that the optimal alignment is non-backtracking. Therefore, Model 3 is employed to run the test example. The parameters for running the proposed genetic algorithm are summarized in Table 9.2.

Table 9.2: Genetic parameters for test example 1.

Parameter	Value
Number of intersection points n	10
Population size n_p	100
Maximum number of generations n_T	1500
Number of operators in each generation	4 for each type of genetic operator
Coefficient of selective pressure q	0.1
Parameter for non-uniform mutation ξ	6

Recall that the solution algorithm is probabilistic and the result is dependent on the random seed. For this reason, we run the program 10 times. The results are given in Table 9.3. The sample mean of these ten runs is 102.873 million dollars and the standard deviation is 0.767 million, about 0.7% of the mean value. A small coefficient of variation (i.e., standard deviation ÷ the mean) indicates that the solutions found with different random seeds are consistent and the algorithm converges to solutions of similar quality. Table 9.3 also shows that the 6th run ends up with the lowest total cost (101.764 million). The graphical presentations of the horizontal and vertical alignments for the 6th run are displayed in Figure 9.4 and Figure 9.5.

Figure 9.4 shows that the optimal horizontal alignment skirts the two small hills to minimize earthwork cost and avoids high cost cells to minimize location-dependent cost. The optimal vertical alignment in Figure 9.5 also shows that the alignment is very close to the ground elevation in order to minimize earthwork excavation and embankment. The results are consistent with our expectations.

Table 9.3: Program results for test example 1.

Run	Objective Value Found (Units: $1,000,000)
1	103.338
2	104.481
3	102.940
4	102.719
5	103.263
6	101.764
7	102.720
8	103.112
9	102.396
10	101.992
Mean	102.873
Standard Deviation	0.767

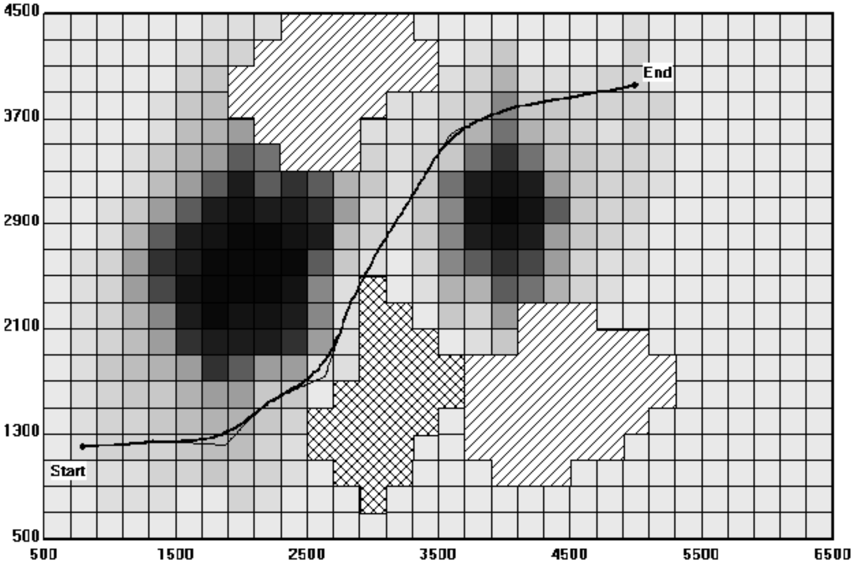


Figure 9.4: The optimized horizontal alignment for test example 1.

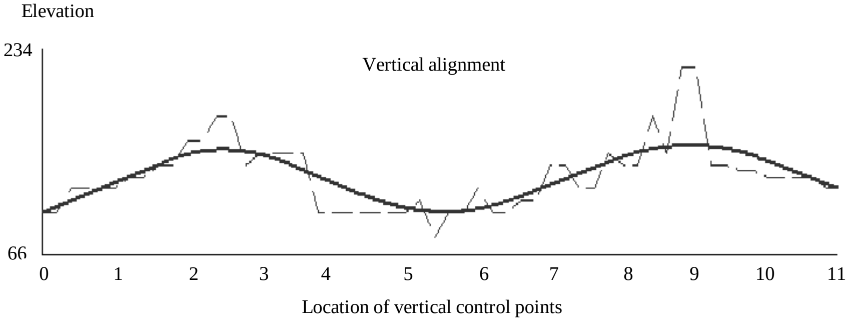


Figure 9.5: The optimized vertical alignment for test example 1.

In order to visualize the evolution of the program, we plot the objective value versus the generation number for the 6th run in Figure 9.6. The figure shows that the objective values in the first few generations are extremely high. However, the value drops sharply down to 120 million at the 10th generation, and to 107 million at the 43rd generation. After that, the improvement in the objective value becomes very slow. The final objective value of $101.764 million is reached at the 1498th generation.

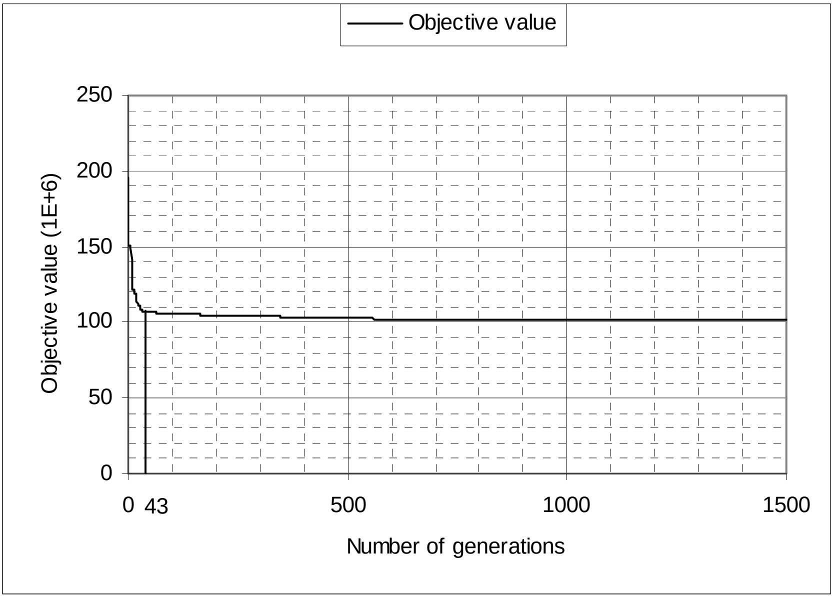


Figure 9.6: Objective value through successive generations for test example 1.

Although the solution found by the proposed algorithm seems to be reasonable, we wish to assess how good the solution is. Since we do not know the exact optimal solution to the problem (note that no existing methods can guarantee finding the global optimum), it is very hard to prove the goodness of the solution found by the proposed algorithm. Therefore, we design an experiment to statistically test the goodness of the algorithm. The experiment is initialed by randomly generating solutions to the problem. For each of them we then evaluate its objective value. This procedure is a sampling process. To maximize the generality and satisfy the statistical requirements, the sample must be created in such a way that the solutions are representative and independent of each other. This is similar to the way we generate the second type of initial population for the genetic algorithm, as presented in section 7.4.

The next step in the experiment is to fit a distribution to the objective values for the random sample. The fitness of the distribution can be checked with the Chi-Square or K-S tests (Neter, *et al*, 1982). Since the sample is randomly generated, the fitted distribution should be able to reflect the actual distribution of the objective value for the real population. Based on this distribution, we can compare the solution found by the proposed algorithm and calculate the cumulative probability of the solution in the distribution. A lower cumulative probability means that most solutions end up with a higher objective value than the one found by the proposed algorithm. The lower the probability, the better the solution.

Following the experiment, we first create a random sample of 40,000 observations. The objective value of the best solution in this sample is 164 million. The worst solution yields an objective value of 2630 million. The sample mean is 602 million and the standard deviation is 234 million. Next, we must fit an appropriate distribution to the objective value. After trying different distributions, it is found that the best fitting one is the Gamma distribution shown below.

$$TC = 164 + \mathrm{Gamma}(129, 3.4). \tag{9.1}$$

The above distribution shows that the objective value has an offset of 164 million. In other words, the minimum value in the distribution of objective value is 164, which is much higher than the solution (101.764) found by the proposed algorithm. Figure 9.7 shows the relative positions of the solution found at the 4th and 1498th generations in the distribution diagram. It is found that after 4 generations, the objective value (150) is lower than the lower bound (164) of the fitted Gamma distribution. Of course, the final solution (101.764) dominates all possible values in the distribution.

The above analysis indicates that the solution found by the proposed algorithm is remarkably good when compared with other possible solutions to the problem. However, the result also raises an interesting question: why can the solution found by the proposed algorithm easily dominate other solutions, even at the first generation. There are two main reasons: (1) the proposed initial population members in the genetic algorithm are generated in such a way that they carry very useful information about the optimal solution, and (2) the genetic operators are designed in such a way that the population is very likely to evolve toward the optimal solution. As discussed earlier, in this book several modifications are made to the classical GAs and special logic is added to the

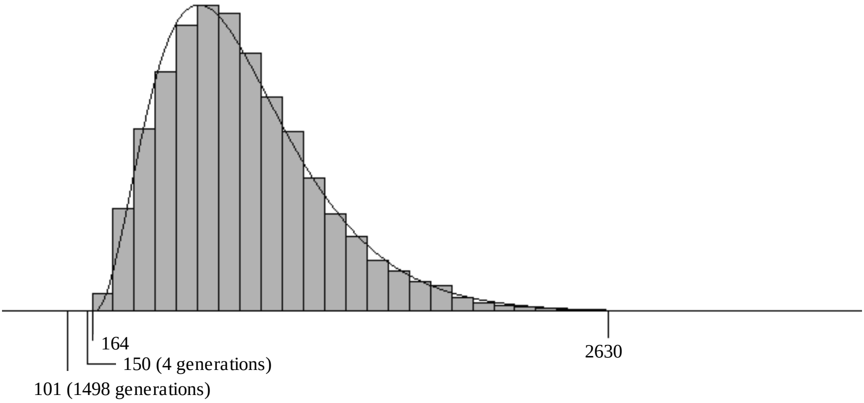


Figure 9.7: The fitted distribution of the objective value for test example 1.

evolution procedure, based on our understanding of the problem's nature. This enables the search algorithm to find a relatively good solution.

9.1.3 Sensitivity analysis of genetic operators

The performance of genetic algorithms is highly dependent on the genetic operators, through which the population can become increasingly adapted to the problem. In this section, we conduct a sensitivity analysis to examine the influence of different types of genetic operators on the solution and the computation effort.

The operators designed for the proposed algorithm can be classified into two main categories: mutation and crossover. To test how different types of operators affect the solution, five scenarios are designed for the sensitivity analysis. The numbers of each genetic operator used in each scenario are listed in Table 9.4. Note that the first four operators in the table are mutation-based, while the remaining four are crossover-based. Also note that scenario 1 is the base case we adopted to solve the problem in the previous section.

The total number of operators in all scenarios is 32. Each of the scenarios emphasizes different types of operators. The first one has equal numbers of mutation-based and crossover-based operators. In the second scenario, there are three times more mutation-based than crossover-based operators. The third one consists only of mutation-based operators, without any crossover-based operators. Scenarios 4 and 5 switch the numbers of mutation-based and crossover-based operators in scenarios 2 and 3. Recall that there exists a high correlation between the final solution and the random seed. Hence, we run 50 replications for each scenario to generalize the results. Moreover, in order to reduce the random fluctuations among different scenarios, each scenario uses the same random seeds as those adopted in scenario 1. "Common Random Numbers" is a popular variance-reduction technique widely used in simulation analysis (Law, 1991). Program outputs are summarized in Table 9.5.

Table 9.4: The numbers of genetic operators in the scenarios for sensitivity analysis.

Scenario	Total no. of operators	Mutation				Crossover			
		Uniform	Straight	Non-uniform	Whole non-uniform	Simple	Two-point	Arith.	Heuristic
1	32	4	4	4	4	4	4	4	4
2	32	6	6	6	6	2	2	2	2
3	32	8	8	8	8	0	0	0	0
4	32	2	2	2	2	6	6	6	6
5	32	0	0	0	0	8	8	8	8

Table 9.5: Program outputs for different scenarios.

Outputs (million)	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**	**Scenario 5**
1st Smallest	101.598	^{1}101.093	108.426	^{4}101.392	114.877
2nd Smallest	101.719	^{2}101.108	109.386	101.472	122.883
3rd Smallest	101.726	^{3}101.371	110.074	101.514	125.433
4th Smallest	101.762	^{5}101.398	110.315	101.578	126.384
5th Smallest	101.764	101.431	110.648	101.669	126.397
5th Largest	104.481	103.411	115.716	104.876	169.496
4th Largest	104.518	103.871	115.838	105.120	170.043
3rd Largest	104.606	104.154	115.871	105.284	171.992
2nd Largest	105.392	104.328	115.935	105.641	175.070
1st Largest	105.430	104.861	116.452	106.713	183.403
Mean	102.989	102.385	113.479	103.038	151.298
Stand Dev.	0.976	0.828	1.870	1.150	14.618

Table 9.5 lists the outputs of the best and worst 5 replications, as well as the sample mean and standard deviation for each scenario. The best solution among all outputs is found in scenario 2 (101.093), while the worst solution occurs in scenario 5 (183.403). The sample mean also indicates that scenario 2 has the lowest mean output and scenario 5 has the highest mean output. Note that four of the five best solutions among all replications are found to occur in scenario 2. Moreover, the mean outputs of scenarios 1, 2, and 4 are very close. The row of standard deviations tells us that the outputs of scenario 2 are more concentrated, whereas the outputs from scenario 5 have high variations. It is also found that the outputs of scenario 5 (around 150) are almost 150% as those of the other four scenarios (around 100). Since scenario 5 does not employ any types of mutation operators, we can conclude that without introducing mutation operators the proposed GA does not yield satisfactory results.

In addition to the program outputs, the computation times for running each replication of each scenario are also recorded. Note that all scenarios are tested on a Pentium II 266 Personal Computer (PC). Table 9.6 indicates that the mean computation time for scenario 3 is the highest among all scenarios, while that for scenario 5 is the lowest. The standard deviations also show that the computation times for scenario 3 vary considerably, while those for scenario 5 are very similar. The most interesting result in the table is that the computation times for scenario 5 are extremely small compared with those for other scenarios. Moreover, from both Table 9.4 and Table 9.6, it is found that as more mutation-based operators are used in the model, the computation times increase. That is because mutation-based operators involve extensive computations on real numbers. (In computing terminology, these computations are referred to as "floating point" operations.) On the other hand, crossover-based operators are just swapping and reassigning gene values in the selected chromosomes, and do

Table 9.6: Computation times for different scenarios.

Computation time (sec)	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Min	70	91	101	48	7
Max	159	177	200	99	12
Mean	97.88	108.48	144.06	60.2	9.9
Stand Dev.	24.98133	22.96336	36.21276	14.34701	1.035098

not require complex calculations. Therefore, the computation times for crossover-based operators are relatively small.

Recall from Table 9.5 that the program outputs for the first four scenarios do not show a big difference in objective values. To visualize the variations, we plot the output of each replication in Figure 9.8. The diagram shows that scenarios 1, 2 and 4 yield very similar results. The outputs of scenario 3 are relatively high when compared with those resulting from other scenarios.

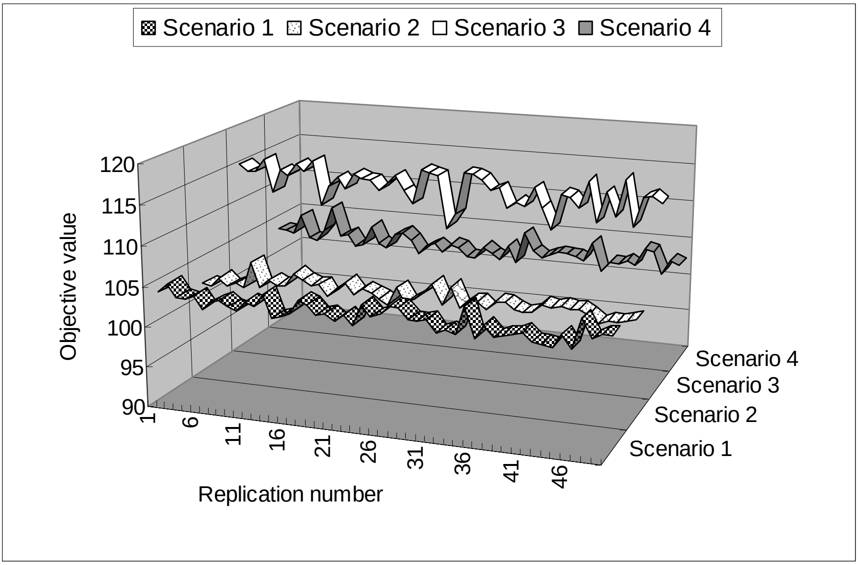


Figure 9.8: Program outputs for the first four scenarios.

For a more rigorous analysis, we make statistical inferences about the difference between the program outputs for each pair of scenarios. Since each scenario has the same number of replications and uses the same random seed, the program outputs can be considered as the observations from matched samples. Let o_{ik} and o_{jk} denote the outputs of the k^{th} replication for scenario i and j.

Then the difference between o_{ik} and o_{jk}, denoted by w_k is treated as an observation from a single population. The value of w_k is computed by:

$$w_k = o_{ik} - o_{jk}\,,\ \forall k = 1,......,n_o \tag{9.2}$$

where n_o = number of observations.

We denote the mean output of the population of differences by μ_w. Then

$$\mu_w = E(w_k) = E(o_{ik} - o_{jk}) = E(o_{ik}) - E(o_{jk}) = \mu_i - \mu_j \tag{9.3}$$

where μ_i and μ_j are the mean output of scenarios i and j.

Since μ_i and μ_j are usually unknown, the sample mean of w_k, denoted by $\overline{w}$ is employed to estimate μ_w. Note that $E(\overline{w}) = E(w) = \mu_w$ and thus $\overline{w}$ is an unbiased estimator of μ_w. Next, we calculate the variance of $\overline{w}$ with the following equation:

$$s^2(\overline{w}) = \frac{s_w^2}{n_o} \tag{9.4}$$

where $s^2(\overline{w})$ = sample variance of $\overline{w}$
s_w^2 = sample variance of w.

It can be proved (Neter, *et al*, 1982) that $s^2(\overline{w})$ is an unbiased estimator of population variance of $\overline{w}$. With the estimators $\overline{w}$ and $s^2(\overline{w})$, we are able to test the following hypothesis:

$$H_0:\ \mu_i - \mu_j = 0 \text{ or } \mu_i = \mu_j \tag{9.5}$$

$$H_1:\ \mu_i - \mu_j \neq 0 \text{ or } \mu_i \neq \mu_j. \tag{9.6}$$

The test statistic for the preceding matched samples is:

$$t^* = \frac{\overline{w} - 0}{s(\overline{w})} = \frac{\overline{w}}{s(\overline{w})} \tag{9.7}$$

where t^* = t distribution with degree of freedom $n_o - 1$.

Following the above approach, we test the difference between the program outputs for each pair of scenarios. The results are given in Table 9.7. Note that the statistical test for the hypothesis stated in eqn (10.5) is a two-tailed test.

Table 9.7: Inferences about differences between the outputs for each pair of scenarios.

Pairs	$\overline{w}$	$s^2(\overline{w})$	$\lvert t^* \rvert$	$t(0.975; n_o - 1)$	Inference
μ_1 and μ_2	0.6034	0.0238	3.9112	2.01	H_1
μ_1 and μ_3	-10.4902	0.0458	49.0375	2.01	H_1
μ_1 and μ_4	-0.0497	0.0294	0.2902	2.01	H_0
μ_2 and μ_3	-11.0936	0.0409	54.8875	2.01	H_1
μ_2 and μ_4	-0.6531	0.0284	3.8742	2.01	H_1
μ_3 and μ_4	10.4405	0.0385	53.2096	2.01	H_1

Accordingly, the test statistic must be taken as the absolute value of t^* given in eqn (10.6).

The statistical tests in Table 9.7 show that $\mu_1 = \mu_4$ while $\mu_i \neq \mu_j$ for all pairs other than scenarios 1 and 4, at the 0.05 significance level. This result is consistent with Table 9.5. Based on this analysis, we conclude that $\mu_2 < \mu_1 = \mu_4 < \mu_3 << \mu_5$. It must be mentioned that the result does not definitely apply to other problems and is only applicable to the test example. We are unable to make a general conclusion unless we test many different problems beyond the scope of this book. However, the methodology introduced in this section should enable researchers to conduct different kinds of sensitivity analysis on the parameters used in GAs.

One important result of the above sensitivity analysis is that a scenario without both mutation-based and crossover-based operators yields poor solutions. This phenomenon should be universal for all GA implementations. In the test example, the mean outputs for scenario 3 (without crossover-based operators) and scenario 5 (without mutation-based operators) are worse than the other scenarios. To explain this result, we must investigate the functions of mutation and crossover operators. Recall from chapter 5 that a mutation operator is supposed to help in exploring the search space. If a GA does not include any mutation operator, the gene pool will lose diversity and the solution will stick in a local optimum. On the other hand, if a GA only consists of mutation operators, the offspring will lose the resemblance to its ancestors and cannot inherit good genes (useful information) to adapt to the problem system. However, if the evolution is long enough, a GA with only mutation operators will yield better results than the one with only crossover operators because mutation operators may explore all possible solutions to the problem, similar to a random search. This can explain why scenario 3 produces a better solution than scenario 5.

To illustrate the above analysis, we now employ the first replication as an example to compute the standard deviation of the objective values of the population for scenarios 3 and 5. Figure 9.9 shows that the standard deviation for scenario 5 is much lower than for scenario 3. It is also found that the standard deviation for scenario 3 fluctuates around 100 after 500 generations and that for scenario 5 almost becomes 0 after 15 generations. These results illustrate that the individuals in a GA population that includes only crossover-based operators will soon become identical as the GA evolves.

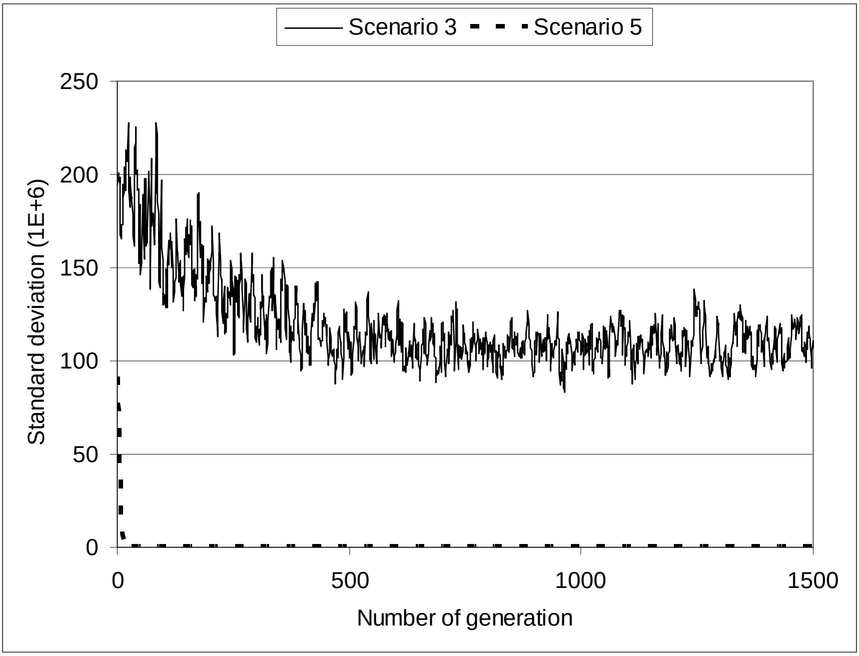


Figure 9.9: Standard deviation of program outputs for scenarios 3 and 5.

Before closing this section, we would like to investigate the difference in the alignment configurations for the best five alternatives among the 250 solutions. The corresponding horizontal and vertical alignments are displayed in Figure 9.10 and Figure 9.19. It can be seen from these figures that the horizontal alignments are almost in the same corridor, and the vertical alignments are also similar. Basically, the alignments get around these two hills and follow the ground profiles to minimize earthwork cost. For this test example, the proposed algorithm seems to get similar solutions. Aside from scenarios 3 and 5, which yield poor results, the other three scenarios have very small standard deviations (see Table 9.5), meaning that the solutions are consistent. However, for other problems, it is quite possible that solutions with nearly similar objective values but significantly different alignments would be obtained, and that such solutions would offer some valuable choices to decision makers. In fact, given some subjective, intangible, or political factors associated with alignment selections, it may be useful to present decision makers with several alternatives which are nearly optimal in their objective function values but significantly different in their locations. This possibility is pursued in chapter 10.

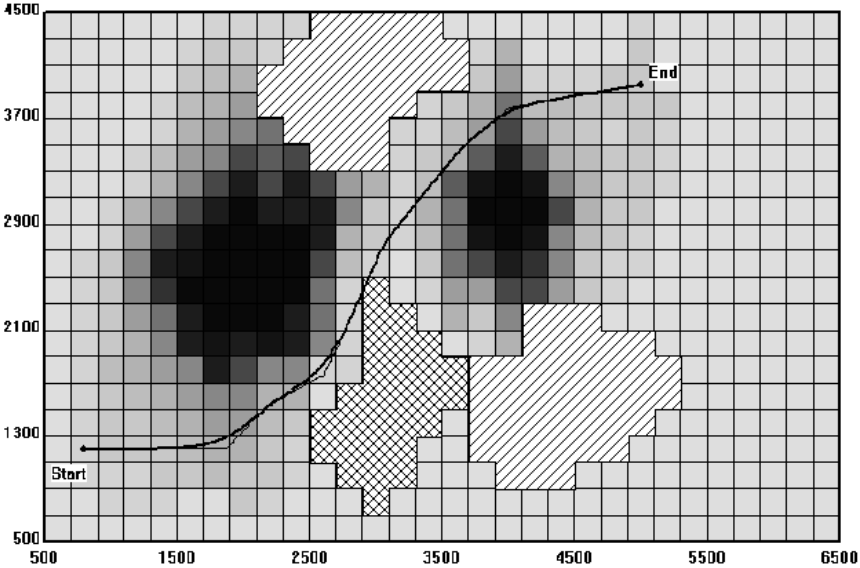


Figure 9.10:The horizontal alignment of the first best solution for test example 1.

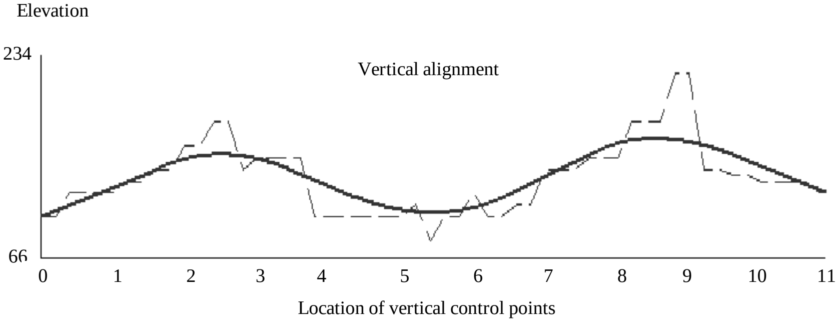


Figure 9.11: The vertical alignment of the first best solution for test example 1.

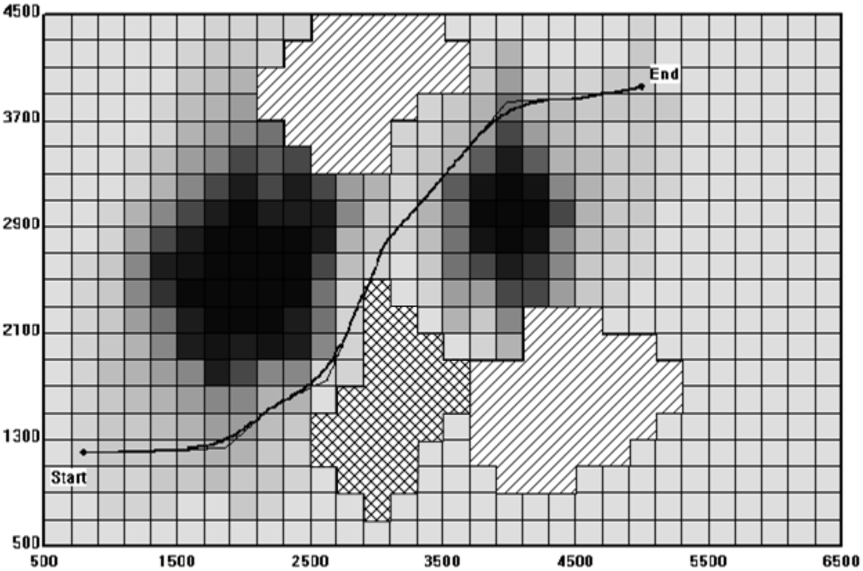


Figure 9.12: The horizontal alignment of the second best solution for test example 1.

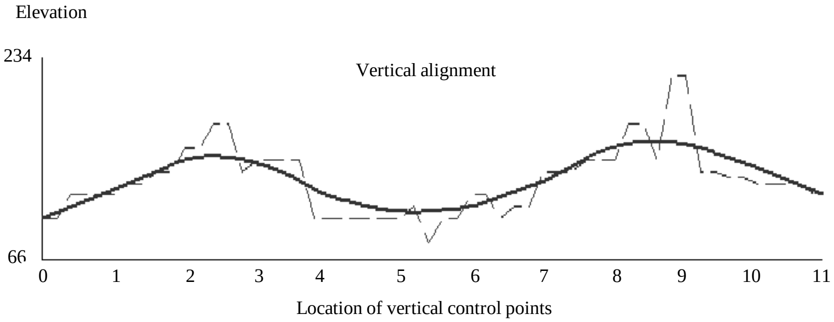


Figure 9.13:The vertical alignment of the second best solution for test example 1.

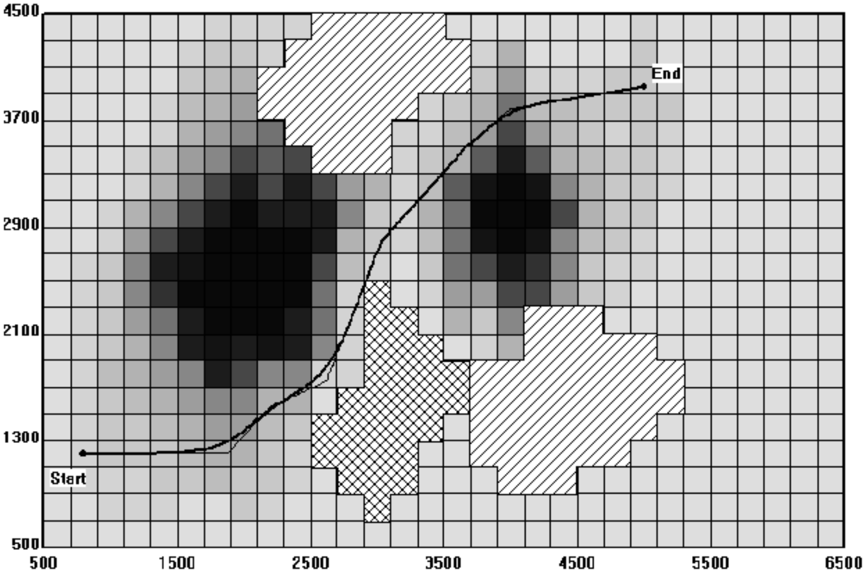


Figure 9.14: The horizontal alignment of the third best solution for test example 1.

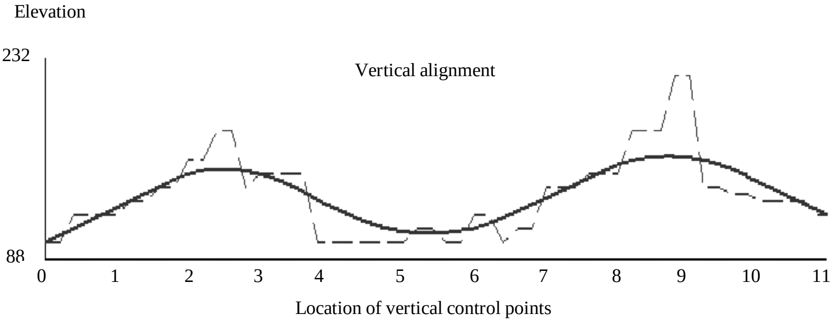


Figure 9.15: The vertical alignment of the third best solution for test example 1.

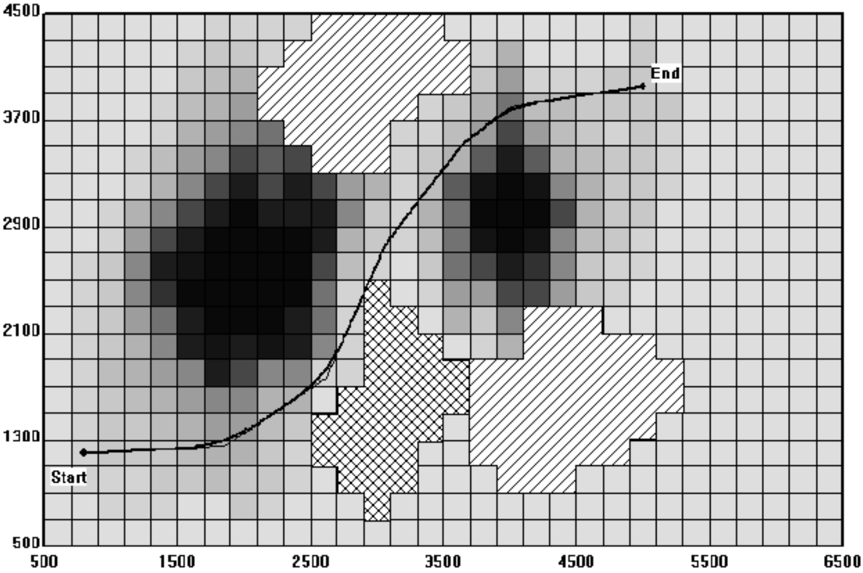


Figure 9.16: The horizontal alignment of the fourth best solution for test example 1.

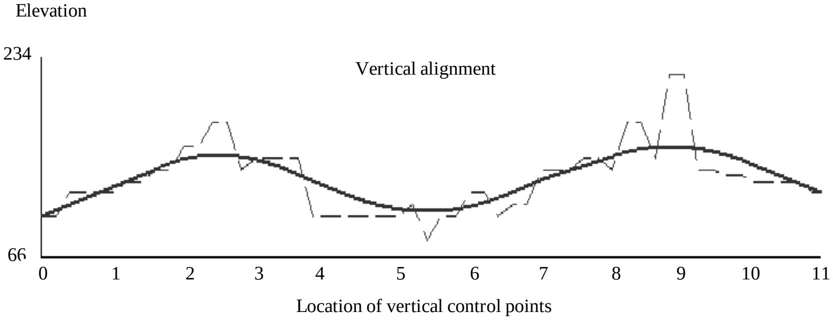


Figure 9.17: The vertical alignment of the fourth best solution for test example 1.

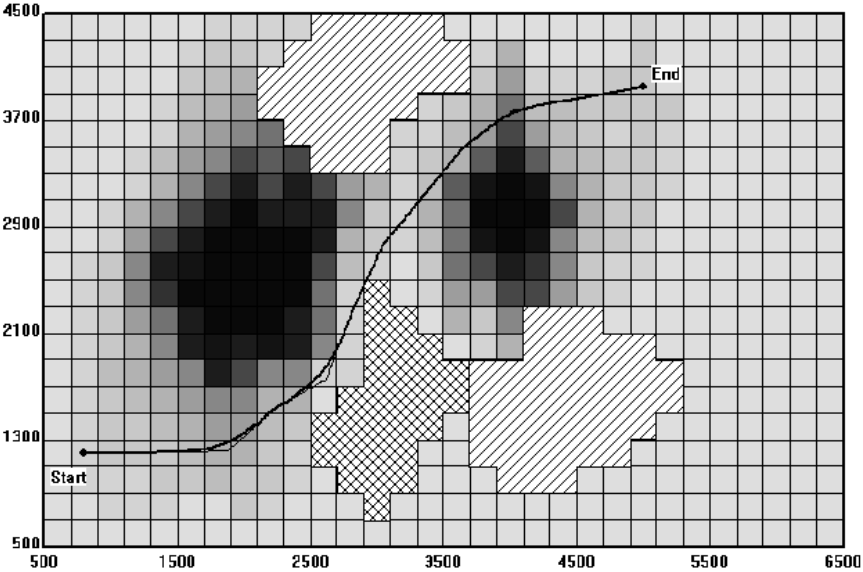


Figure 9.18:The horizontal alignment of the fifth best solution for test example 1.

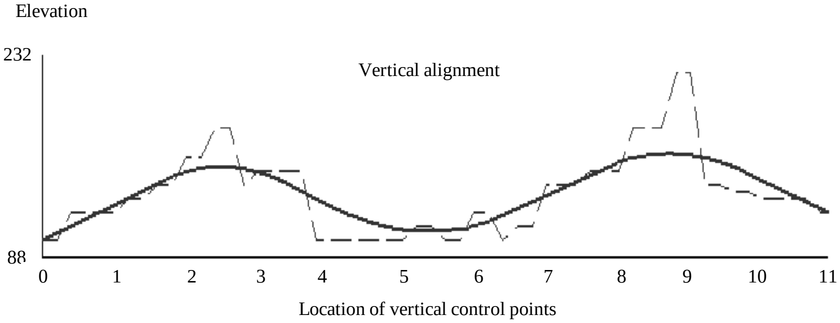


Figure 9.19: The vertical alignment of the fifth best solution for test example 1.

9.2 Case study 2

The second test example is intended to investigate the effectiveness and efficiencies of Models 3 and 4 in optimizing non-backtracking alignments. The terrain through which the alignment is optimized is illustrated first. The solutions found by Models 3 and 4 are then presented. A comparison is also made to illustrate the differences of program outputs and computation efforts between these two models.

9.2.1 Problem description

The plan view of the study region for test case 2 is displayed in Figure 9.20. The X and Y coordinates of the region range from 1000 to 3200 and from 1000 to 2500, respectively. The region is divided into $100\,\text{ft} \times 100\,\text{ft}$ cells. The shades of the map show a mountain on its top and a hill on its bottom. The contours and a 3D view of the map are presented in Figure 9.21 and Figure 9.22. These illustrate the topography through which the alignment is optimized.

The start and end points of the alignment are located at (1150, 2450, 200) and (2950, 1150, 250) respectively. Since they are located in two different mountains, the alignment should circle around the north mountain and then shift to the south hill at some place where the elevations are smooth between them. The design speed is limited to 40 mph because the alignment traverses a mountainous area. The design parameters for the proposed highway are summarized in Table 9.8.

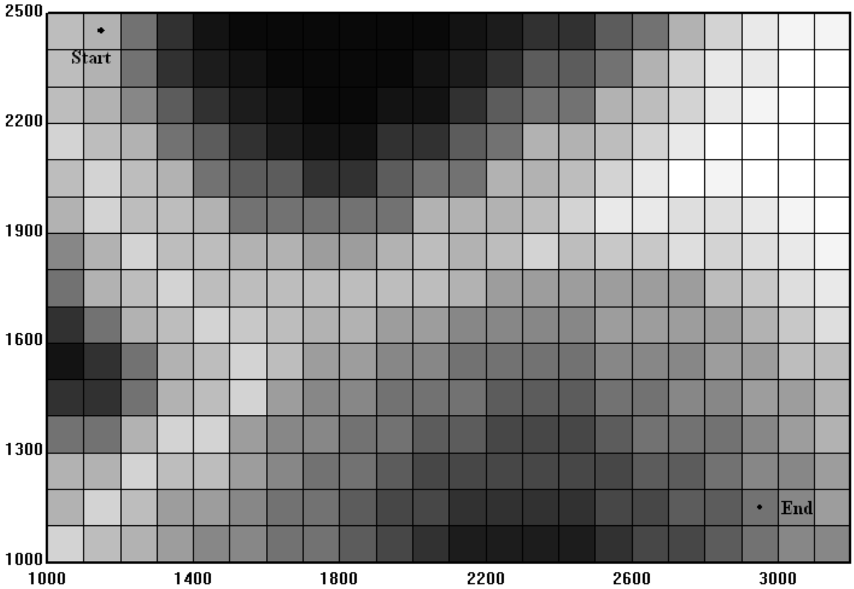


Figure 9.20: Map for test example 2.

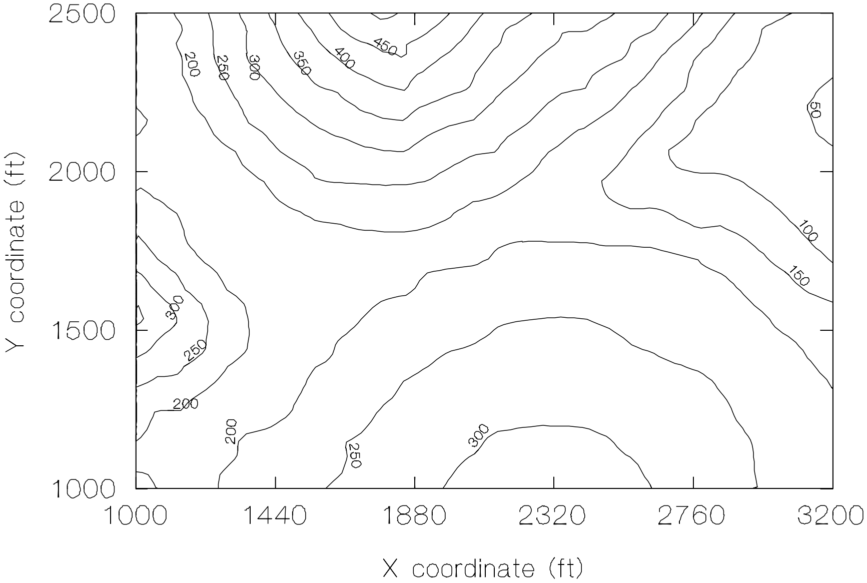


Figure 9.21: Contours of test example 2.

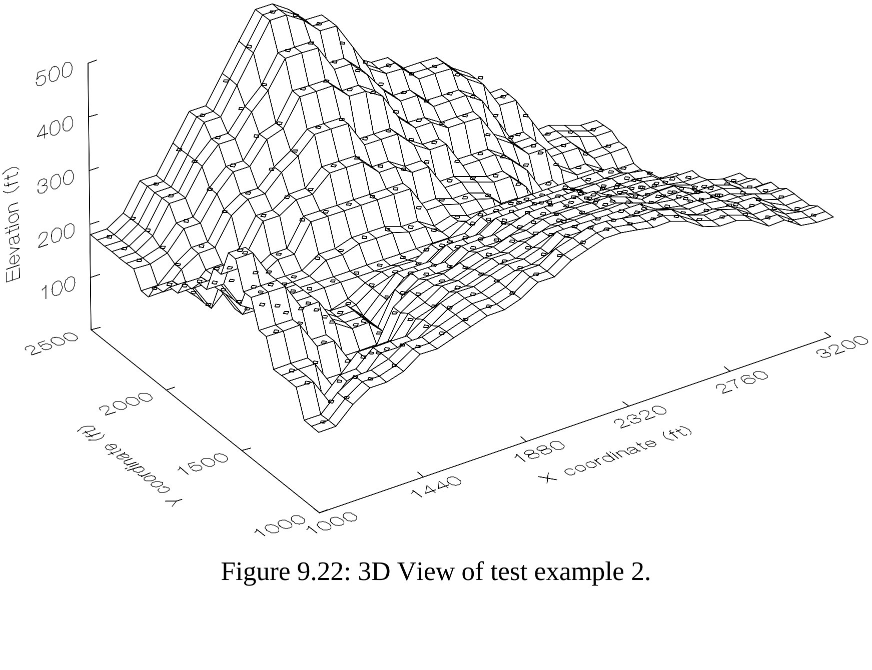


Figure 9.22: 3D View of test example 2.

Table 9.8: Design parameters for test example 2.

Design Parameter	Design Value
Design speed	40 mph
Side friction coefficient	0.16
Superelevation	0.06
Maximum grade	5%
Forward friction coefficient	0.28
AADT	3000 vehicles per day

9.2.2 Comparisons of solutions found by Models 3 and 4

For this test example, we first run Model 3 ten times and record the program outputs and computation times for each run. Then we use the same random seeds to run Model 4 for ten replications so that the results can be compared for the same base. The program parameters in this test example are also the same for both models. We use 10 intersection points and run the program for 1000 generations. There are four genetic operators of each type in each generation. Both models are tested on a Pentium II 266 PC.

Table 9.9 shows that the best solution found by Model 3 is $7.262 million, which is slightly above 7.063, the best solution found by Model 4. However, the worst solution found by Model 3 (7.912) is better than that found by Model 4 (7.951). The mean values indicate the average objective value of Model 4 (7.597) is lower than that of Model 3 (7.722), but not very different. The standard deviations show that the solutions found by Model 3 are more concentrated than those by Model 4.

The computation times in Table 9.9 show that the computation efforts for running Model 4 are about double those for running Model 3. Recall that for the same number of intersection points, the number of decision variables in Model 4 is higher than that in Model 3. Thus, Model 4 needs more time to run the program through the same number of generations.

Since the program outputs for the two models do not differ much, we plot a bar chart of the outputs for both models in Figure 9.23 to visualize the differences. Note that column 11 represents the mean value of outputs 1 to 10. It can be seen that at the second and fifth runs, Model 3 yields better solutions (lower objective values), while at the other runs, Model 4 has better results. The maximum difference occurred at the sixth run at which the best solution of Model 4 is found.

Using the approach introduced in the previous section, we can make a statistical inference about the differences between these two models. The t^* value for this example is 3.395 (computed with eqn (9.7)). If the significance level is 0.05, the critical t value is $2.262 < 3.395$. Therefore, we conclude that on average, Model 4 produces better results than Model 3 for this example.

Table 9.9: Program outputs and computation times of Models 3 and 4 for test example 2.

Run	Model 3		Model 4	
	Output (Million $)	**Run Time (Sec.)**	**Output (Million $)**	**Run Time (Sec.)**
1	7.857	44	7.639	85
2	7.262	44	7.774	77
3	7.865	41	7.645	85
4	7.687	42	7.552	86
5	7.582	44	7.951	83
6	7.912	41	7.063	80
7	7.734	41	7.504	76
8	7.837	49	7.603	89
9	7.740	44	7.711	72
10	7.747	44	7.524	83
Max	7.912	49	7.951	89
Min	7.262	41	7.063	72
Mean	7.722	43.4	7.597	81.6
Stand Dev.	0.189	2.413	0.230	5.253

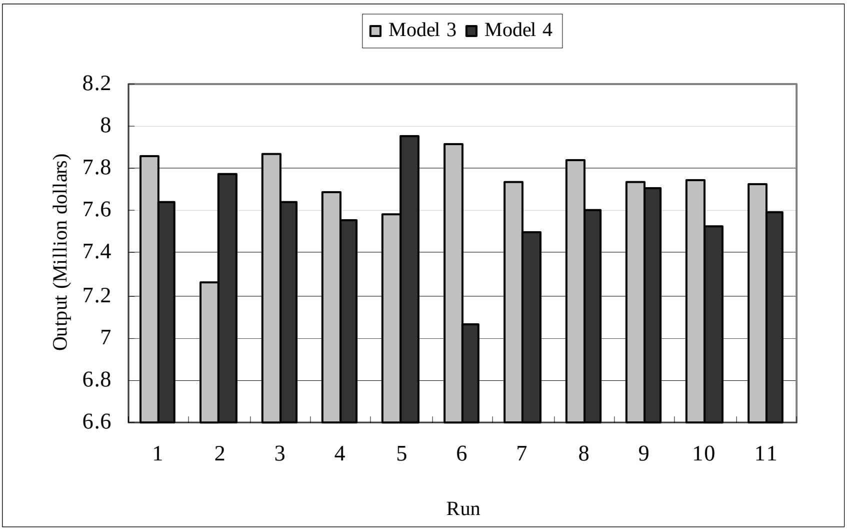


Figure 9.23: Program outputs of Models 3 and 4 for test example 2.

To compare the configurations of the best solutions found by both models, we plot the optimal horizontal and vertical alignments for both models in Figure 9.24 to Figure 9.27. It can be seen that the best alignments found by both models are very similar. The alignment wends along the contour line of the north mountain and then crosses the low elevation area in the middle of the map to reach the south hill. Since the slope of the south hill is not very sharp, the alignment ascends along the terrain to reach the end point. Both models yield similar and reasonable results.

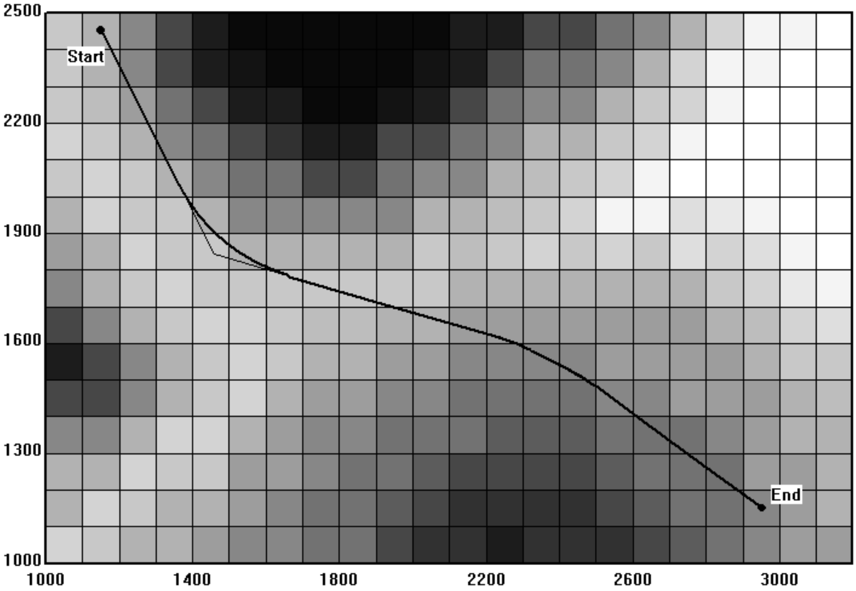


Figure 9.24: The optimized horizontal alignment found by Model 3 for test example 2.

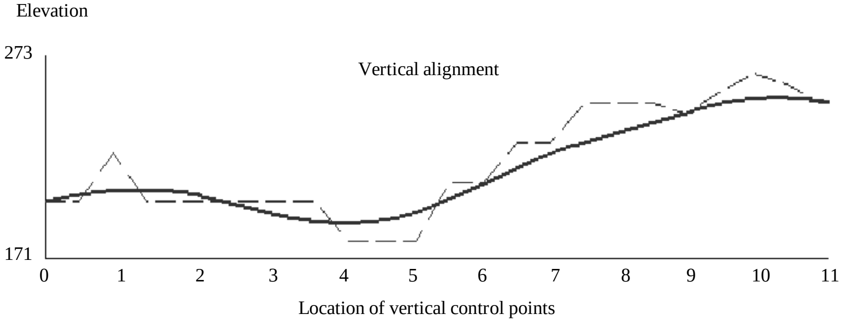


Figure 9.25: The optimized vertical alignment found by Model 3 for test example 2.

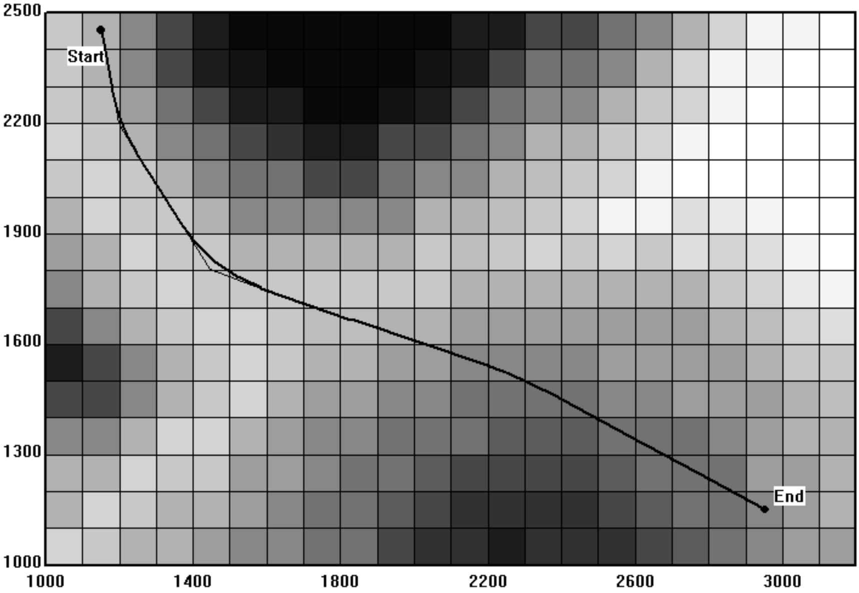


Figure 9.26: The optimized horizontal alignment found by Model 4 for test example 2.

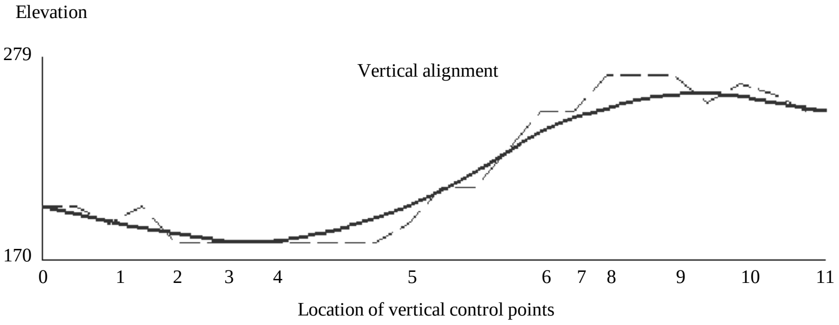


Figure 9.27: The optimized vertical alignment found by Model 4 for test example 2.

The objective value in successive generations for the best runs of both models is displayed in Figure 9.28. It shows that in this test example, Model 4 converges faster than Model 3 and ends up with a slightly better solution. The picture also indicates the proposed algorithms improve the objective very quickly in the first 50 generations. After that, the improvements come much more slowly.

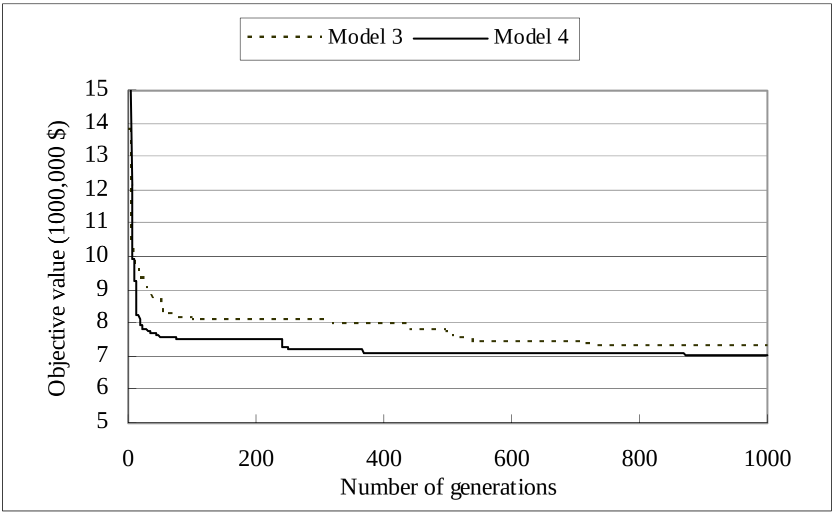


Figure 9.28: Objective value through successive generations for test example 2.

9.3 Case study 3

In the previous section, the performances of Models 3 and 4 have been tested on a non-backtracking optimal alignment. In this section, another example in which the optimal alignment is backtracking is employed to investigate the effectiveness and efficiencies of Models 3 and 4 again. Through this study, we may have a general idea about the applicability and capability of Models 3 and 4 in different situations.

9.3.1 Problem description

The plan view of the region of interest for test example 3 is presented in Figure 9.29. The X and Y coordinates of the study region extend from 2000 to 4200 and 1000 to 2500, respectively. The dimension of the cells in the region is $100\,\text{ft}\times100\,\text{ft}$. The shade of the map indicates that there are three mountains in the region. The cross-section field in the middle of the map represents a lake. Any alignment passing through the lake will be charged a very high penalty cost. The contours and 3D view of the map are displayed in Figure 9.30 and Figure 9.31. It can be seen that the terrain is extremely irregular and the ground elevations are not smooth.

The start and end points of the proposed highway alignment are located at (2150,1250,100) and (3850,2350,250). The elevation difference between the start and end points is relatively high. In addition, the mountain and the lake obstruct direct paths between the start and end points. Accordingly, they cannot be directly connected. The optimal alignment should wind around the mountain at the bottom-left corner and then along the bank of the lake to reach another mountain. This test example is the most difficult when compared with the previous two examples. In fact, the optimal alignment should be backtracking.

The design speed is 40 mph due to the difficult terrain. Other design parameters are the same as those shown in Table 9.8 for test example 2.

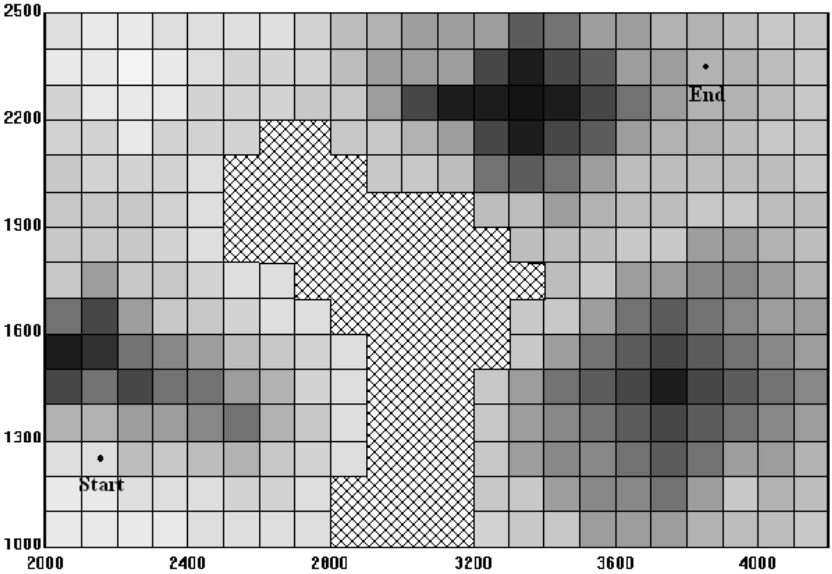


Figure 9.29: Map for test example 3.

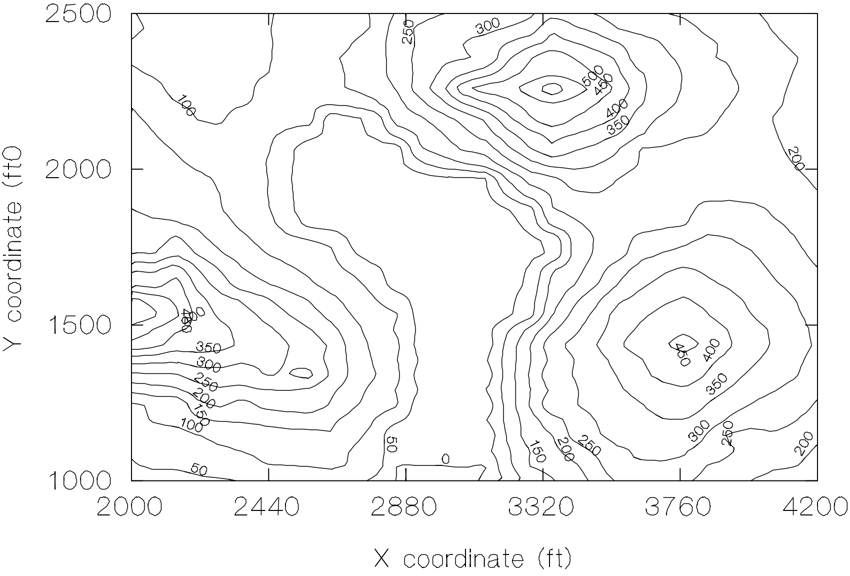


Figure 9.30: Contours of test example 3.

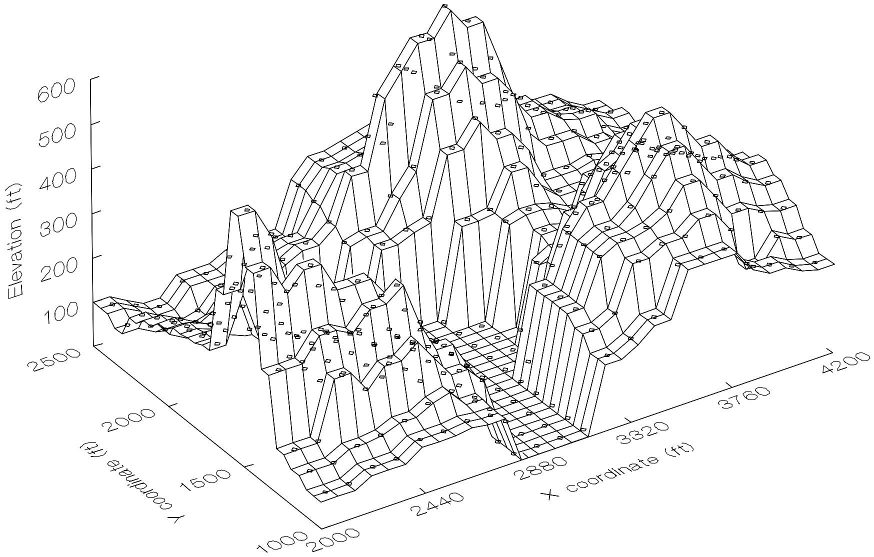


Figure 9.31: 3D View of test example 3.

9.3.2 Comparisons of solutions found by Models 3 and 4

Since the terrain is irregular, the maximum number of generations is set at 2000. All other program-input parameters are the same as the second test example shown in section 9.2.2. Assume that 10 intersection points are used to generate the alignment for both Models 3 and 4. To have the same comparison base, we use common random seeds to run the program on PC Pentium II ten times for both models. The program outputs and computation efforts are listed in Table 9.10.

Table 9.10 shows that the solutions found with Model 4 dominate those found with Model 3. The best solution found with Model 3 is $31.945 million, which is still higher than the worst solution with Model 4 ($27.323 million). The results are mainly due to the limitation of model structures. Since Model 3 cannot deal with backtracking and here the optimal alignment is supposed to be backtracking, the resulting solutions of Model 3 are definitely worse than those of Model 4. The sample means indicate the average objective value of Model 4 ($24.827 million) is about 57% of that of Model 3 ($42.397 million). Moreover, the standard deviations show that the objective values found by Model 4 are more concentrated than those by Model 3.

As in the second example, the computation times for Model 4 are about double those for Model 3 due to the increase in the number of decision variables. In addition, the standard deviations of computation times also indicate that the running time of Model 4 has a larger variation, but the difference in running time between the two models is not significant.

Table 9.10: Outputs and computation times of Models 3 and 4 for test example 3.

Run	Model 3		Model 4	
	Output (Million $)	**Run Time (Sec.)**	**Output (Million $)**	**Run Time (Sec.)**
1	52.135	91	27.202	166
2	36.011	93	24.412	173
3	53.418	89	27.323	172
4	40.629	128	25.370	168
5	34.101	98	21.086	182
6	52.272	120	23.154	168
7	31.945	99	23.709	188
8	37.723	93	25.223	216
9	52.445	94	24.845	182
10	33.287	94	25.942	189
Max	53.418	128	27.323	216
Min	31.945	89	21.086	166
Mean	42.397	99.9	24.827	180.4
Stand Dev.	9.077	13.169	1.879	15.072

The best horizontal and vertical alignments found by both models are presented in Figure 9.32 to Figure 9.35. Figure 9.32 shows that the horizontal alignment found by Model 3 passes through the mountain at the bottom-left corner of the map and the curve there is quite sharp. This is due to the limitation of Model 3, which cannot allow a backward bending alignment to get around the mountain. It can also be seen in Figure 9.33 that the ground elevation between the 2nd and 3rd intersection points is much higher than the road elevation. That is the place where the alignment passes through the mountain. After the 4th intersection point, the alignment basically follows the ground elevation and winds along the lake and the north mountain to reach to end point.

The optimized horizontal alignment presented in Figure 9.34 seems reasonable. The alignment skirts the mountain at the bottom-left corner and then winds along the bank of the lake to the north mountain. Figure 9.35 shows that the optimized vertical alignment closely follows the existing ground elevation to minimize earthwork cost. The above results indicate that Model 4 is capable of optimizing backtracking 3-dimensional alignments for irregular terrain.

The objective values in successive generations for the best runs of both models are plotted in Figure 9.36. The figure shows that in the first 120 generations, the objective values for both models are almost the same, while between 160 and 320 generations, the objective values of Model 3 are better than those of Model 4. However, after 320 generations, Model 4 dominates Model 3. The figure also reveals a phenomenon: even after the improvements in the objective values of Model 4 become insignificant, possible improvements may still be found at later generations. That means that the objective function over the search space is extremely irregular so that during many generations the solution may be trapped at a local optimum. However, eventually, the algorithm forces the solution jump from one local optimum to a better one. This illustrates the power of the proposed algorithm in optimizing a difficult problem with many local optima.

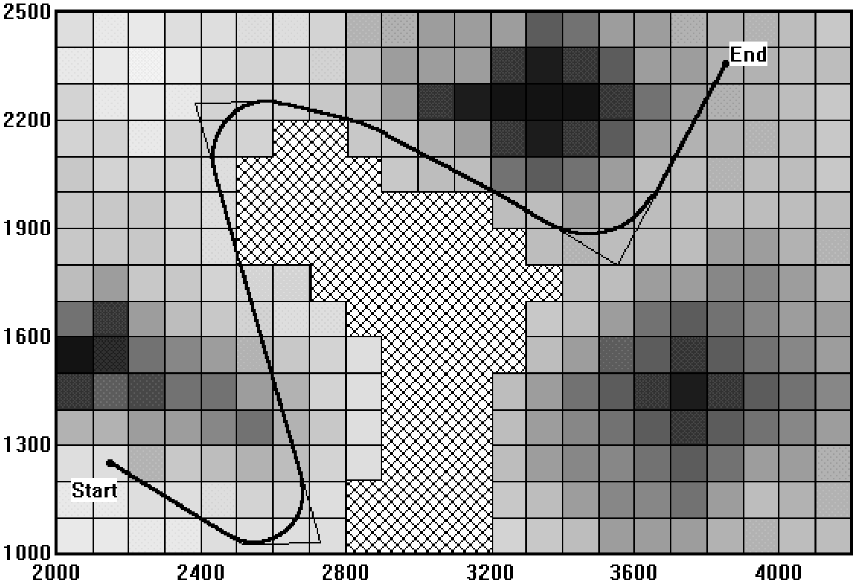


Figure 9.32: The optimized horizontal alignment found by Model 3 for test example 3.

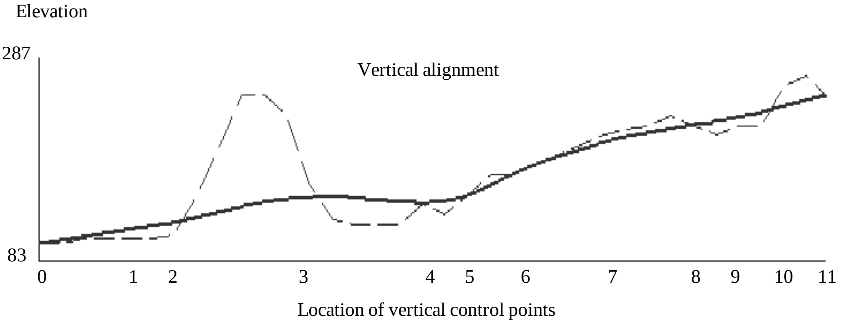


Figure 9.33: The optimized vertical alignment found by Model 3 for test example 3.

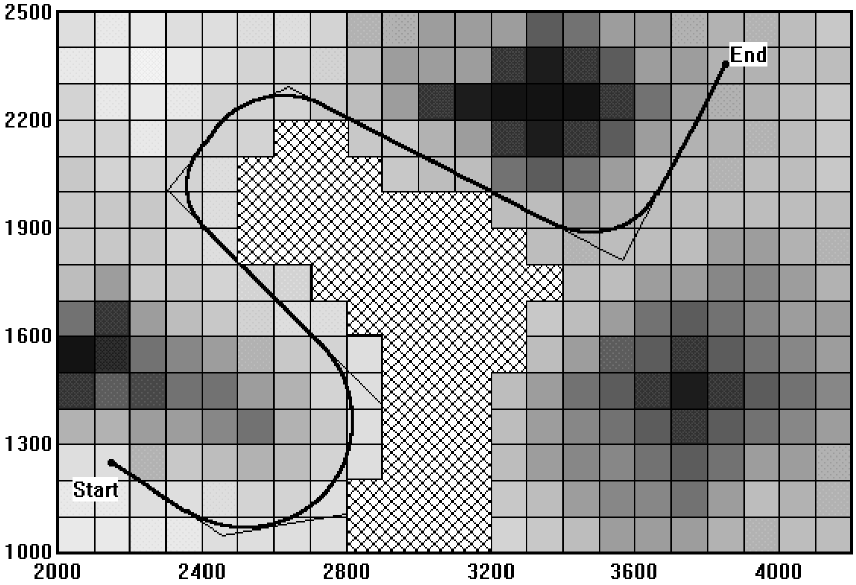


Figure 9.34: The optimized horizontal alignment found by Model 4 for test example 3.

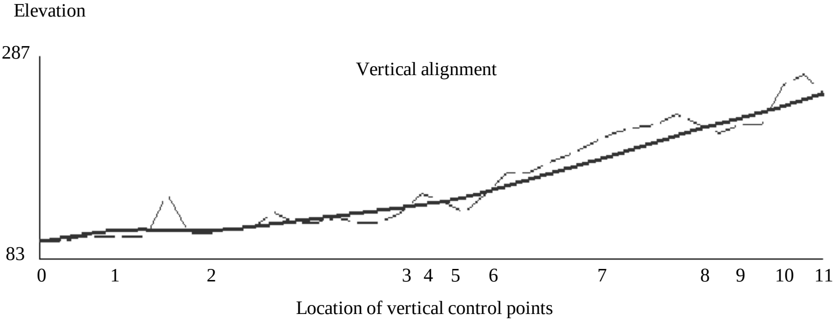


Figure 9.35: The optimized vertical alignment found by Model 4 for test example 3.

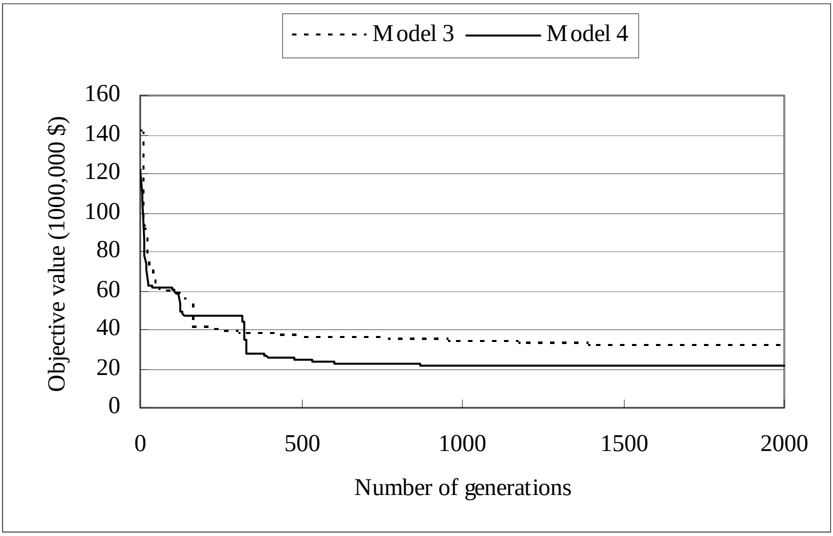


Figure 9.36: Objective value through successive generations for test example 3.

9.3.3 Goodness test for the best solution found by Model 4

Since this test example is relatively difficult, we would like to test the goodness of the best solution found by Model 4 to investigate the performance of the proposed search algorithm. Following the experiment design introduced in section 9.1.2, a set of representative and independent solutions is randomly generated. For Model 4, the most general solutions are the members of the fourth population type in the initial population (see chapter 8), i.e., the intersection points scatter randomly within the study region with random elevations.

After creating a random sample of 40,000 solutions, we observe that best solution yields an objective value $171 million, while the objective value of the worst one is $1702 million. The sample mean is around $809 million and the standard deviation is $253 million. After trying various distributions, it is found that the following Weibull distribution best fits the sample.

$$TC = 171+\text{Weibull}(713, 2.71) \tag{9.8}$$

Eqn (10.7) shows that the distribution has an offset of $171 million, which is much higher than the best solution $(21.086 million) found with Model 4, i.e., the best solution dominates all possible solutions in the distribution. The relative position of the best solution in the distribution is shown in Figure 9.37.

$$TC = \text{Normal}(809,253) \tag{9.9}$$

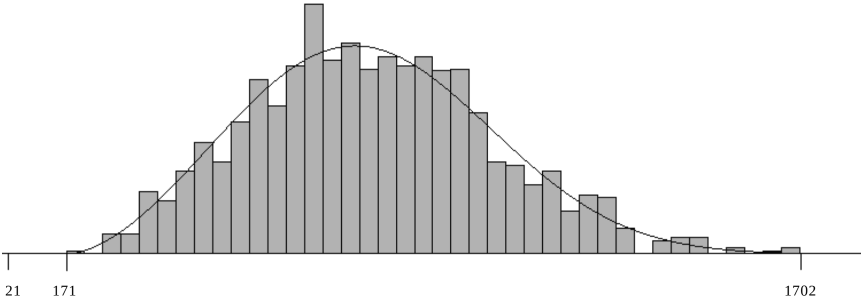


Figure 9.37: The fitted Weibull distribution of the objective value of Model 4 for test example 3.

Since the distribution has a bell shape, we can also try to fit the sample with a normal distribution so that we can calculate the cumulative probability of the best solution found by Model 4. The fitted normal distribution is given in eqn (9.9) and displayed in Figure 9.38.

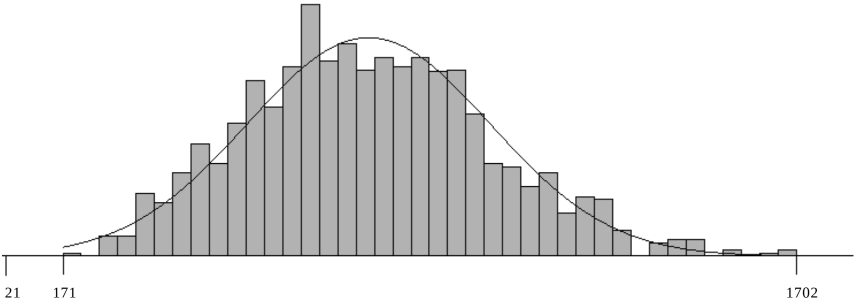


Figure 9.38: The fitted Normal distribution of the objective value of Model 4 for test example 3.

The cumulative probability of the best solution (21.086) found by Model 4 in the above normal distribution is

$$P(TC \leq 21) = P(\frac{TC-809}{253} \leq \frac{21-809}{253}) = P(z \leq -3.11) = 1 - P(z \leq 3.11) = 0.0009\,.$$

In other words, the solution (21.086) dominates 99.91% of the solutions in the distribution, meaning that the solution found with Model 4 is excellent when compared to other possible solutions to the problem. We should remember, however, that 21.086 is 8.11 times smaller than the best of 40,000 randomly generated solutions. Such results give us confidence in the proposed search algorithm.

Chapter 10

Alignment optimization with GIS

In this chapter we discuss procedures to optimize highway alignments with GIS. The developed model uses genetic algorithms for optimization and is also connected with a GIS for efficient real-world application. A digital GIS map is required for performing the optimal search. The case studies presented in this book are based on the GIS data from Maryland. In Maryland a desktop electronic property map called MDProperty View is developed. It contains scanned property maps with detailed information on property boundary, assessment value, area, zoning, and many other relevant parameters.

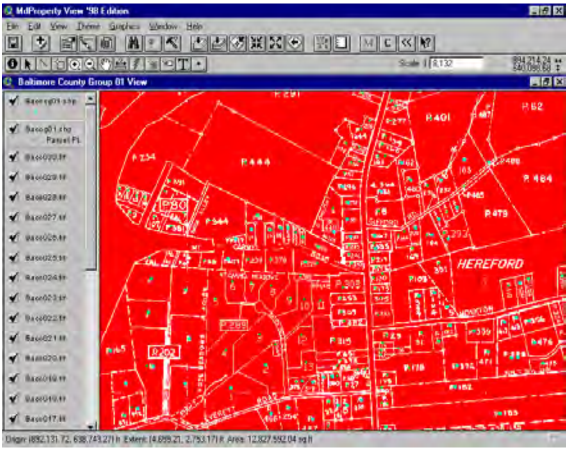
10.1 An overview of MDProperty View

MDProperty View is a collection of ArcView^{TM} GIS-based property datasets (Figure 10.1) which include ownership of land, land-area, assessment value of the lands, land-use, and information on the residential properties such as detailed data describing the characteristics of improvements. On a GIS map, geographic features can be represented as points (such as cities on a country map), lines (for example, road networks), or polygons (such as land parcels). In Arcview^{TM} GIS, information about map features are stored in a database, which are linked to map features. This information is referred to as *attribute information*, or simply *attributes*. The attribute information is stored in a table called *attribute table* or *theme table*. The attributes of a building might include its name, owner type, size, or building permit number. The property characteristics in MDProperty View are linked with the property maps (Figure 10.2) in the form of two separate databases, (1) the assessment database and (2) the computer aided mass appraisal (CAMA) database. The assessment database has 66 fields. Some of the fields relevant to this study are area, zoning, and appraised values of land parcels.

Besides property maps, listed below are some other useful maps which are also included in MDProperty View:

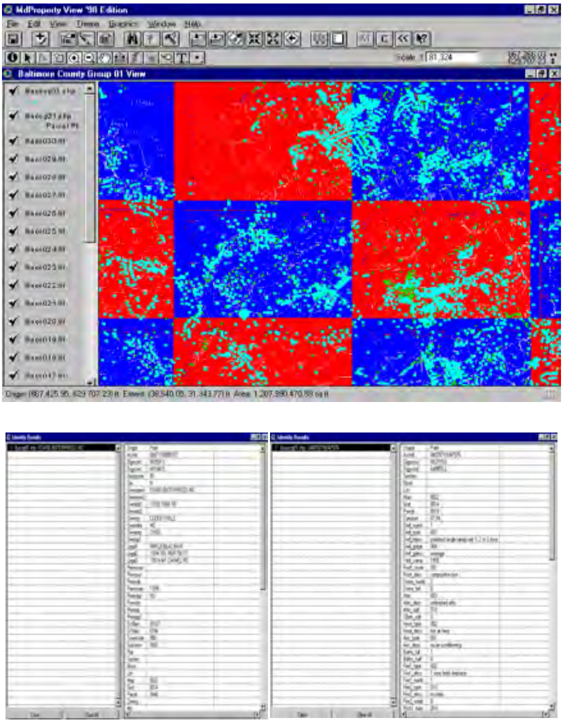
- digital roads and streams

- landsat satellite true color composite images, and
- spot satellite images


MdProperty View '98 Edition
File Edit View Theme Graphics Window Help
Scale 1: 8,132
Baltimore County Group 01 View
HEREFORD

Origin (892,131.72, 638,743.27) ft Extent (4,699.21, 2,753.17) ft Area 12,827,592.04 sq ft

Figure 10.1: Property maps in MDProperty View.


MdProperty View '98 Edition
File Edit View Theme Graphics Window Help
Scale 1: 81,324
Baltimore County Group 01 View
Identify Results
Identify Results

Origin (867,425.95, 629,707.23) ft Extent (38,540.05, 31,343.77) ft Area 1,207,990,470.59 sq ft

Figure 10.2: Zoomed property image with assessment and CAMA data.

10.2 Other GIS database and maps

Besides maps and databases obtained from MDProperty View, a number of other GIS features are used in this work, which are obtained from the Maryland State Highway Administration (SHA). SHA maintains a number of GIS databases. Some of the databases which may be relevant to highway optimization include the following:

(1) digital orthophoto quarter quad maps
(2) floodplains
(3) wetlands
(4) 1-meter digital ortho quadrangle
(5) Chesapeake Bay critical area
(6) federal urban area boundaries in Maryland
(7) land use/land cover data
(8) road network data
(9) watersheds
(10) streams, and
(11) soils

Of these GIS datasets only floodplain, wetland, and soil maps are considered in this study.

10.3 Environmental issues

The effects of unusual environmental features such as wetlands and floodplains are taken into consideration by assigning them a very high unit cost so that they will be penalized during the search. However, since the environmental data were available as separate GIS layers the following procedure was developed to combine them with the property data:

Procedure 10.1 Consideration of environmental factors

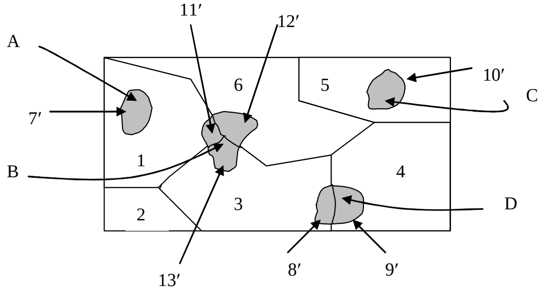
Step 1: Overlay the environmental layers (i.e. layers for floodplains and wetlands) on the digitized property layer. (Digitization of properties is discussed.)

Step 2: Redraw the environmental layers and treat them as separate parcels called environmental parcels on the property layer. This would require splitting some of the properties and readjusting unit costs (see Figure 10.3).

Step 3: Modify the attribute table by assigning the environmental parcels a high unit cost.

In Figure 10.3, the original properties are designated as 1-6. The wetland and floodplains are designated as A-D. After overlaying floodplains and wetlands (i.e., A-D) on the original properties, 7 new environmental parcels (represented by 7′-13′) are obtained. The attribute table of the resulting map is modified and the environmental parcels are assigned a high unit cost.

Effects of air/water pollution and increased noise levels were not considered in this research. It was assumed that environmental cost arising due to air/water pollution and increased noise level was incidental to the project cost and that selection of alignments will not be influenced by these costs. It was also assumed that the air quality impacts were not localized. Environmental cost due to floodplain and wetland impacts is treated as a part of right-of-way cost in this work.



Modified Attribute Table

Parcel #	Area (sq. ft.)	Unit Cost ($/sq. ft.)
1	.	.
2	.	.
3	.	.
4	.	.
5	.	.
6	.	.
7′	.	1000
8′	.	1000
9′	.	1000
10′	.	1000
11′	.	1000
12′	.	1000
13′	.	1000

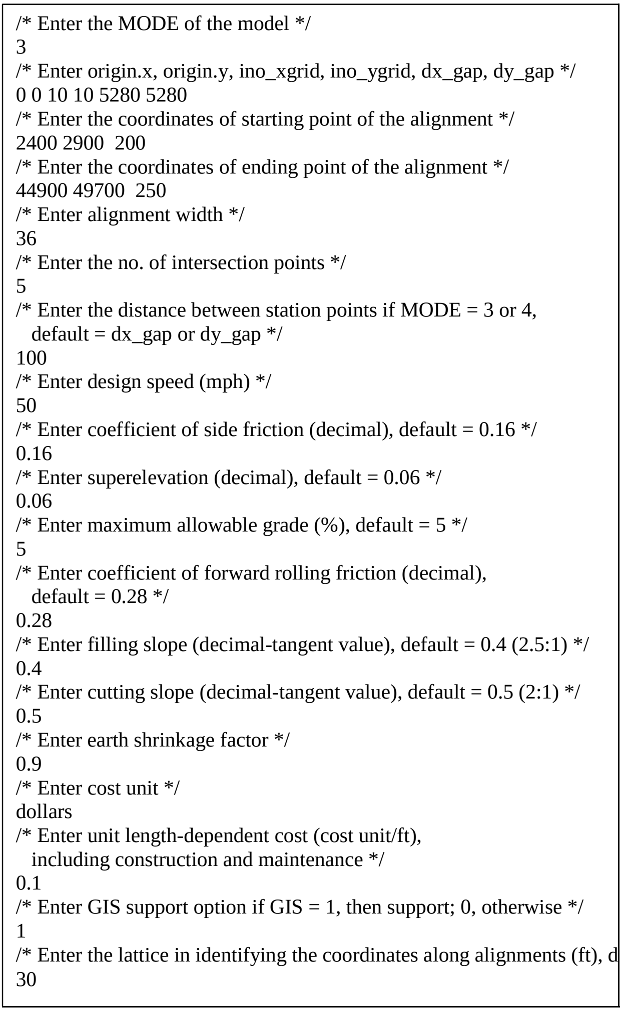
Figure 10.3: Digitization and superimposition of floodplains and wetlands.

10.4 Developing solution algorithms with GIS

Developing solution algorithms with GIS for alignment optimization requires several steps, such as preprocessing of the GIS map, developing algorithms based on spatial relations, integrating GIS with genetic algorithms, and compactness analysis for right-of-way calculation. These are discussed next.

10.4.1 Obtaining input data and maps

In the optimization model, a number of program inputs are required which are supplied manually to the model. A typical input file for applying the model for a grid-shaped property having 54 uniform rectangular land parcels (9×6) is shown in Figure 10.4. It should be noted that while most of the input parameters have single values, the cost and elevation data is required for every land parcel (i.e. for every cell in the grid). The unit cost and elevation matrices are shown in Figure 10.5 and Figure 10.6, respectively. When connected to a GIS, real property values are used directly from the GIS database in lieu of the cost matrix (Figure 10.5).



```
/* Enter the MODE of the model */
3
/* Enter origin.x, origin.y, ino_xgrid, ino_ygrid, dx_gap, dy_gap */
0 0 10 10 5280 5280
/* Enter the coordinates of starting point of the alignment */
2400 2900  200
/* Enter the coordinates of ending point of the alignment */
44900 49700  250
/* Enter alignment width */
36
/* Enter the no. of intersection points */
5
/* Enter the distance between station points if MODE = 3 or 4,
  default = dx_gap or dy_gap */
100
/* Enter design speed (mph) */
50
/* Enter coefficient of side friction (decimal), default = 0.16 */
0.16
/* Enter superelevation (decimal), default = 0.06 */
0.06
/* Enter maximum allowable grade (%), default = 5 */
5
/* Enter coefficient of forward rolling friction (decimal),
  default = 0.28 */
0.28
/* Enter filling slope (decimal-tangent value), default = 0.4 (2.5:1) */
0.4
/* Enter cutting slope (decimal-tangent value), default = 0.5 (2:1) */
0.5
/* Enter earth shrinkage factor */
0.9
/* Enter cost unit */
dollars
/* Enter unit length-dependent cost (cost unit/ft),
  including construction and maintenance */
0.1
/* Enter GIS support option if GIS = 1, then support; 0, otherwise */
1
/* Enter the lattice in identifying the coordinates along alignments (ft), d
30
```

Figure 10.4: Input data required for the optimization model (continued overleaf).

```
/* Enter cost matrix */
 50   50   50   40   50   20   55   50   50
 50   50   50   40   50   30   40   60   50
 50   50   50   40   50   50   40   60  250
 60   50   50   50   50   50   35   80  250
 60   50   55   50   60   60   30   80   80
 70   50   50   50   60   60   40   85   80

/* Enter elevation matrix if MODE = 3 or MODE = 4*/
 80   80   80   80   80   80  100  100  100
 80   80   80   80   80  100  120  120  100
 80   80   80   80   80  120  140  150  130
 80   80   80   80  100  140  160  180  200
 80   80   80   80  110  160  180  200  210
 80   80   80  100  140  180  200  250  230
/* Enter unit cost for diesel fuel (cost unit/gallon) */
0.85
/* Enter unit cost for gasoline (cost unit/gallon) */
1.25
/* Enter Average Accident cost (cost unit/per accident)*/
20000
/* Enter unit cutting cost (cost unit/cub yd) */
35
/* Enter unit filling cost (cost unit/cub yd) */
20
/* enter unit transportation cost for moving earth from a borrow pit
   (cost unit/cub yd) */
2
/* Enter unit transportation cost for moving earth to a landfill
   (cost unit/cub yd) */
3
/* Enter analysis period (years), default = 30 */
30
/* Enter interest rate (decimal), default = 0.06 */
0.06
/* Enter Annual Average Daily Traffic AADT */
5000
/* Enter traffic growth rate (decimal), default = 0 */
0.005
```

Figure 10.4: Input data required for the optimization model (continued).

50	50	50	40	50	20	55	50	50
50	50	50	40	50	30	40	60	50
50	50	50	40	50	50	40	60	250
60	50	50	50	50	50	35	80	250
60	50	55	50	60	60	30	80	80
70	50	50	50	60	60	40	85	80

Figure 10.5: Example of a cost matrix used by the optimization model without GIS integration.

80	80	80	80	80	80	10	10	10
80	80	80	80	80	10	12	12	10
80	80	80	80	80	12	14	15	13
80	80	80	80	10	14	16	18	12
80	80	80	80	11	16	18	20	21
80	80	80	10	14	18	20	25	23

Figure 10.6: Example of an elevation matrix used by the optimization model.

A number of input parameters, such as total number of generations, coefficients of selective pressure and non-uniform mutation, and the epoch type, are associated with Genetic Algorithms (GA), which provide the optimal search approach used in this research. A discussion of these is available in Jong (1998).

For real applications it is very time consuming to manually populate the cost and elevation matrices used in the input file required for running Jong's model. This is because in reality there are thousands of land parcels as well as lakes, mountains, and other geographic features in the search space. Manually filling the unit costs and elevations for each of these geographic entities is simply not possible. Therefore, this process must be automated.

In this work property maps are obtained from MDProperty View, which stores raster property maps for all Maryland counties. In order to use the costs from the MDProperty View's assessment database directly, the attribute tables for the assessment database are modified to include the following information necessary for implementing the proposed optimization model:

- area of land parcels, which is computed using GIS-based algorithm (ft^2)
- unit cost of land parcels ($\$/ft^2$)
- zoning of land parcels (i.e., land-use type. For example, land-use types may be residential, commercial, agricultural, etc.)
- cost of structures (if any)

The procedure for obtaining area and unit costs of parcels is discussed in section 10.4.2.

The ground elevations are available in the form of GIS layers which were obtained from SHA,s GIS database. The elevation maps were then overlaid on the property maps (obtained from MDProperty View). The elevation data were then added to the modified attribute table for the properties. The procedure for including the elevation data is similar to Procedure 10.2 which will be described in section 10.4.2. The elevation data were also extracted as a separate file which can be used as an input to the model. Thus the cost and elevation matrices shown in the input file (Figure 10.4) were removed and the main program in the optimization model was modified to receive this information as separate input files. Since the optimization program uses the cost matrix only to compute the right-of-way cost, for which a separate GIS-based algorithm is developed in this research, a separate input file for cost is not necessary. This is true because the program reads the cost data directly from the modified attribute table of property maps during the right-of-way cost computation.

The elevation data however is stored in the modified attribute table and also as a separate input file due to the following:

(1) Some decisions such as whether to construct overpasses, underpasses, at-grade intersections, tunnels (if the ground elevation is too high) can be made in the GIS environment. Therefore the elevation information must be available in the GIS.
(2) The earthwork costs are computed in the cost module in the "C" environment which requires the elevation data to be available as an input file.

Since the attribute table in a GIS is a database file which can be easily read by any spreadsheet (such as Excel^{TM}) or database (such as Access^{TM}) software, the elevation data can easily be extracted and saved as a separate file. This process is not affected by the number of land parcels/geographic entities in the search space. Thus the cost and elevation matrices are separated from the input file (Figure 10.4) used by the optimization model and are made available in an automated fashion as described above.

In addition to using cost and elevation matrices, the proposed model also uses soil data since the unit cut and fill costs as well as side slopes and ditches are functions of soil type. A separate file for soil type is created in the same way as described for the elevation file above, i.e., overlaying the GIS layer for soil on property map, modifying the property attribute table, and saving the soil type as a separate file. Thus the following three databases are used in this research, which are directly obtained from a GIS:

- unit land-cost
- ground elevation
- soil-type

The entire process of input requirements can be summarized into the following steps:

1. Modify the main program in the optimization model so that it reads the following three separate files:
   A. The main input file (similar to Figure 10.4 by eliminating cost and elevation matrices)
   B. The elevation file
   C. The soil file
2. Modify the attribute table which is linked to the GIS map. The modified attribute table should have area (ft^2), unit cost (\$/ft^2), and elevation (ft^2) of geographic entities.

10.4.2 Preprocessing input data and maps

As stated earlier the maps and databases obtained from MDProperty View as well as other sources (such as elevation and soil maps) need to be preprocessed before the proposed optimization model can be implemented. There are two key preprocessing steps required:

(1) Digitization of raster maps obtained from MDProperty View

A typical Property map and its attribute table (corresponding to the assessment data) obtained from MDProperty View is shown in Figure 10.7. Notice that the attribute table is linked to the centroid of the land parcels, which stores information such as parcel #, area (ft^2), cost ($), and ($x$, y) coordinates of centroids, among other information.

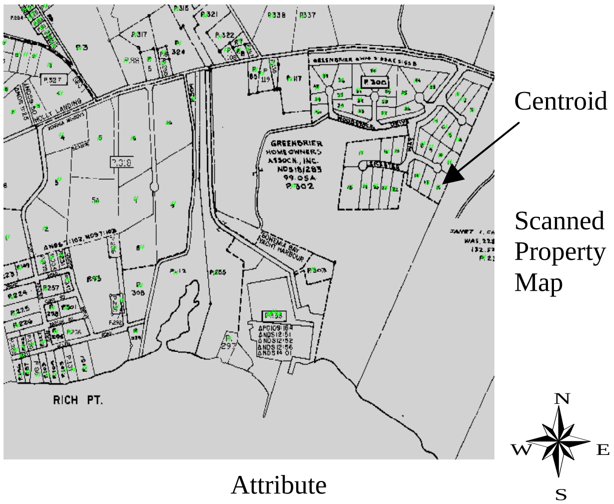
The digitization process used in this study consists of manually redrawing boundaries of *properties* in the region of interest in the ArcView^{TM} GIS environment (Figure 10.8). This method however, may not be appropriate for large-scale applications since manually redrawing boundaries of thousands of properties may be very time consuming. This process can be made simpler by using commercial digitizers such as line tracing equipment or laserscan, which are not used here.

(2) Modification of attribute table of geographic entities

The modification of the attribute table linked to the digitized maps was briefly discussed before. The area and perimeters of polygons representing geographic entities were computed using GIS-based algorithm. While the assessment database in MDProperty View contained the area of geographical entities, some discrepancies were found between these values and values computed using GIS-based algorithm. In this study area and perimeter values computed using GIS-based algorithm was used. GIS-based algorithms were also developed to automatically populate the "Parcel" Field in the attribute table. The following are the necessary fields of the attribute table of the digitized map:

Shape: This field is automatically generated
Parcel: Run the script called parcel.ave to automatically populate this field. Computer programs written in ArcView^{TM} GIS are called script.
Area: Computable using the script called "area.ave."
Perimeter: Computable using the script called "area.ave."
Unit Cost: Follow Procedure 10.2, which is discussed in the next section.

The unit costs are obtained based on spatial selection from MDProperty View. In order to achieve this and automate most of the steps, several GIS-based programs in the *Avenue* language were written. Avenue is the programming language used by ArcView GIS^{TM}. The following procedure is developed to populate the unit costs of parcels in the digitized maps with corresponding unit costs of parcels obtained from the MDProperty View based on spatial relations.



Attribute Table

Property #	**Area (sq. ft.)**	**Cost ($)**	**X-Coordinate**	**Y-Coordinate**
1	.	.	.	.
2	.	.	.	.
.	.	.	.	.

Figure 10.7: Property image in MDProperty View.

Procedure 10.2 Assigning Unit Costs to Parcels in the Digitized Map

Step 1: Obtain the taxmap for the search space from the MdProperty View and add it to a new project

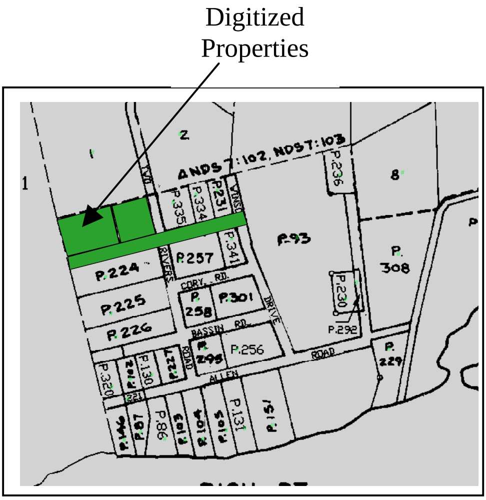
Step 2: Obtain the shapefile for the assessment data from MDProperty View. A shapefile represents geographic entities in ArcView as points, lines, or polygons.

Step 3: Digitize the portion of the map to be used as the search space and save it as a separate theme

Step 4: Add a new field named “Cost” to the attribute table of the digitized map

Step 5: Add a new field named “Cost” to the attribute table of the shapefile for the assessment data

Step 6: Run the avenue code named “cost.ave” to automatically populate the cost field of the digitized map



Attribute Table

Property #	Area (sq. ft.)	Unit cost ($/sq. ft.)
.	.	.
.	.	.
.	.	.

Figure 10.8: Digitization of properties.

In order to compute right-of-way cost (discussed in section 10.4.5) the ratio of area and square of perimeter (A/P^2) must also be added in the attribute table of the digitized map. Thus a revised attribute table is similar to Table 10.1.

A digitized map linked to a revised attribute table is shown in Figure 10.9. Note that unit costs of existing roads and streets are assumed to be very low ($0.05/ft^2) in this research. This assumption is made because by using existing roads highway agencies can avoid paying for additional right-of-way for new road construction. By assigning low unit costs to existing roads it is expected that the optimized alignment will be encouraged to pass through existing roads. Similarly, in order to avoid the lakes and streams, a high unit cost is assigned to these geographic entities ($1000/ft^2) so that the optimized alignment will avoid high cost areas during the search.

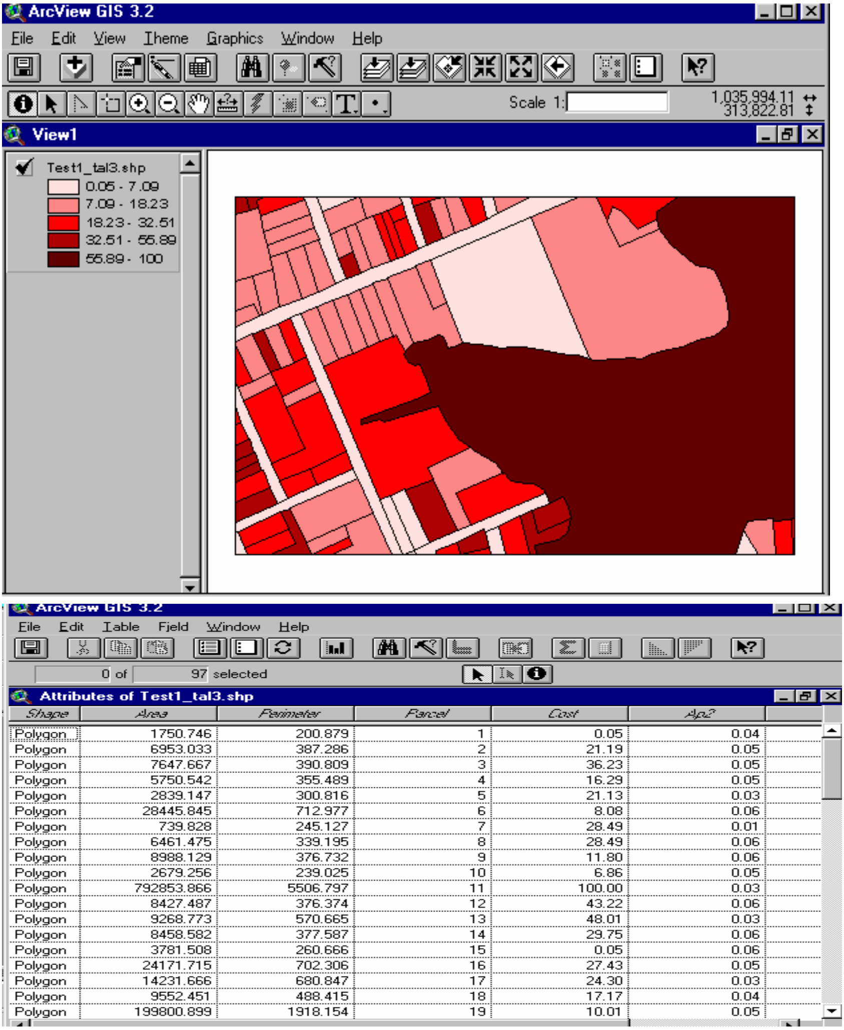
Instead of avoiding streams, lakes, and other water conduits completely during the search, it may be cost effective to construct bridges. The cost of bridges will depend on several factors such as length and type of bridges selected. These factors may depend on the detailed geographic characteristics such as soil type, land-use characteristics in the vicinity, and terrain elevation. Modeling bridges, tunnels, intersections, and interchanges are discussed in Part C of the book.

Table 10.1: A revised attribute table linked to a digitized map.

AREA	PERIMETER	PARCEL	COST	AP2	ELEV
1750.746	200.879	1	0.05	0.04	80.00
6953.033	387.286	2	21.19	0.05	150.00
7647.667	390.809	3	36.23	0.05	100.00
5750.542	355.489	4	16.29	0.05	90.00
2839.147	300.816	5	21.13	0.03	100.00
28445.845	712.977	6	8.08	0.06	100.00
739.828	245.127	7	28.49	0.01	110.00
6461.475	339.195	8	28.49	0.06	180.00
8988.129	376.732	9	11.80	0.06	80.00
2679.256	239.025	10	6.86	0.05	80.00
792853.866	5506.797	11	100.00	0.03	75.00
8427.487	376.374	12	43.22	0.06	90.00
9268.773	570.665	13	48.01	0.03	80.00
8458.582	377.587	14	29.75	0.06	70.00

Note: (1) Cost represents unit cost ($/ft^2)
(2) AP2 is the ratio of area and square of perimeter (dimensionless quantity)
(3) Units of area, perimeter, and elevation are ft^2, ft, and ft, respectively

While the maps and property data for this research are obtained from MDProperty View, which stores maps and data for Maryland counties, the proposed highway alignment optimization model can also be applied to other jurisdictions if the properties maps and essential associated data such as area and unit cost of properties are available. A simple way to achieve this may be to use aerial photogrammetry techniques to obtain maps of geographic entities which may include land parcels, structures, lakes, streams, existing roads, and bridges. The maps can then be digitized and area of the digitized geographic entities can be automatically computed by running the ArcView^{TM} GIS-based computer program named “area.ave” which is developed in this study.



Shape	Area	Perimeter	Parcel	Cost	Ap2
Polygon	1750.746	200.879	1	0.05	0.04
Polygon	6953.033	387.286	2	21.19	0.05
Polygon	7647.667	390.809	3	36.23	0.05
Polygon	5750.542	355.489	4	16.29	0.05
Polygon	2839.147	300.816	5	21.13	0.03
Polygon	28445.845	712.977	6	8.08	0.06
Polygon	739.828	245.127	7	28.49	0.01
Polygon	6461.475	339.195	8	28.49	0.06
Polygon	8988.129	376.732	9	11.80	0.06
Polygon	2679.256	239.025	10	6.86	0.05
Polygon	792853.866	5506.797	11	100.00	0.03
Polygon	8427.487	376.374	12	43.22	0.06
Polygon	9268.773	570.665	13	48.01	0.03
Polygon	8458.582	377.587	14	29.75	0.06
Polygon	3781.508	260.666	15	0.05	0.06
Polygon	24171.715	702.306	16	27.43	0.05
Polygon	14231.666	680.847	17	24.30	0.03
Polygon	9552.451	488.415	18	17.17	0.04
Polygon	199800.899	1918.154	19	10.01	0.05

Figure 10.9: A digitized map with modified attribute table.

10.4.3 Developing algorithms based on spatial relations

In this study a GIS is also used to develop algorithms based on spatial relations. A detailed right-of-way cost formulation is developed in section 10.4.5. Here a GIS-based algorithm is developed for computing the damage to the properties, C_{DP}. Since the search space may also contain streams, lakes, mountains, and existing roads in addition to properties, a general term *geographic entities* is used to represent these entities.

The steps of the algorithm are as follows:

Algorithm 10.1 Computation of damage to geographic entities, C_{DP}

Step 0: Initialize, Set the digitized property map which is linked to the modified attribute table (similar to Figure 10.9)

Step 1: Read (x, y) coordinates of points in the initial population generated by genetic algorithms that makes up the alignment. The points also include the start (S) and end (E) points.

Step 2: Generate the centerline of the alignment in the initial population

Step 3: Create a buffer along the centerline at a distance equal to half the total width of the road, which represents the proposed alignment

Step 4: Identify geographic entities falling in the buffer created in Step 3. See Figure 10.10.

Step 5: Compute fractional area and costs of geographic entities taken by the proposed alignment

Step 6: Compute damage to the geographic entities with eqn (10.5)

Step 7: Save the damage cost, CDP in an external file and pass it to the cost module

The above algorithm computes the damage and passes it to the cost module where other costs are computed. The connection of GIS, genetic algorithms and cost modules will be discussed next.

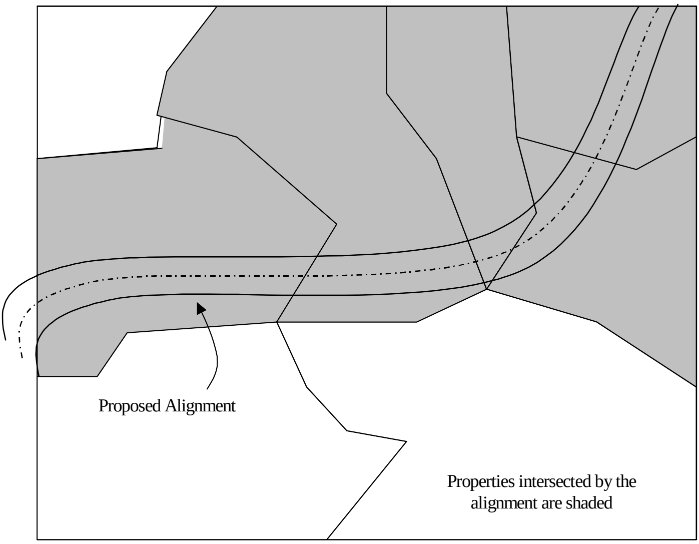


Figure 10.10: Properties intersected by an alignment.

10.4.4 Integrating GIS with genetic algorithms

Since the cost and optimization modules are part of a "C" environment and GIS module is part of GIS environment, it was necessary to link the two environments together for smooth exchange of data. This was achieved by exploiting an ArcView^{TM} based Dynamic Link Library (DLL) called *Avhelp.DLL* (ESRI, 1997). *AVHELP.DLL* has two functions *AVRun* and *AVScript*. The function *AVRun* was used to send ArcView^{TM} commands from a "C" environment. The following procedure was developed to establish communication between "C" and Arcview^{TM}.

Procedure 10.3 Communication between "C" and ArcView^{TM}

Step 0: Initialize, Place Avhelp.DLL in windows system directory

Step 1: Create a new project in ArcViewTM with digitized map with modified attribute table as view and program for algorithm 5.1 as script1. In ArcViewTM GIS a view displays maps and algorithms are programmed as scripts using GIS-based programming language called Avenue.

Step 2: Using the following code trigger ArcViewTM from "C":

```
AVRun("av.run(\"script1\",\"\")");
```

10.4.5 Compactness analysis

The right-of-way costs are difficult to compute since they also involve costs associated with appraisal, damage, and relocation (AASHTO, 1962; ILAC, 1992; IRWA, 1990). The office of Real Estate of the Maryland State Highway Administration (SHA) suggests that appraisal, damage, and relocation costs in many cases account for a significant portion of the total right-of-way cost. Therefore, a right-of-way cost based on the fractional area taken by an alignment is not a true representation of the actual costs incurred by highway agencies.

The location and uses of properties are also important in right-of-way cost computation. For example, right-of-way costs in urban areas and for residential properties may be higher than rural and agricultural properties. Here we seek to provide a comprehensive formulation for right-of-way cost computation. Some right-of-way cost assumptions are based on Maryland SHA practices and may require adjustments for other jurisdictions.

10.4.5.1 Right-of-Way cost formulation

Computing right-of-way cost is very complex since it depends on the before and after values of affected properties and structures, cost associated with *temporary easement*, and appraisal fees. A temporary easement is defined as the partial taking of a property during construction, which is returned to the owner upon completion of the construction.

The significant right-of-way cost components are shown in Figure 10.11 (AASHTO, 1962; ILAC, 1992; IRWA, 1990). In general, the right-of-way cost for parcel i, C_{RW_i} can be expressed as:

$$C_{RW_i} = C_{TE_i} + C_{JC_i} + C_{AF_i} \ , \tag{10.1}$$

where C_{TE_i} = cost of the fraction of property *i* taken for temporary easement
C_{JC_i} = just compensation paid for property *i*
C_{AF_i} = appraisal fees associated with property *i*.

The total right-of-way cost is thus:

$$C_{RW} = \sum_{i=1}^{n} C_{RW_i} \ , \tag{10.2}$$

where *n* is the total number of parcels affected by the alignment.

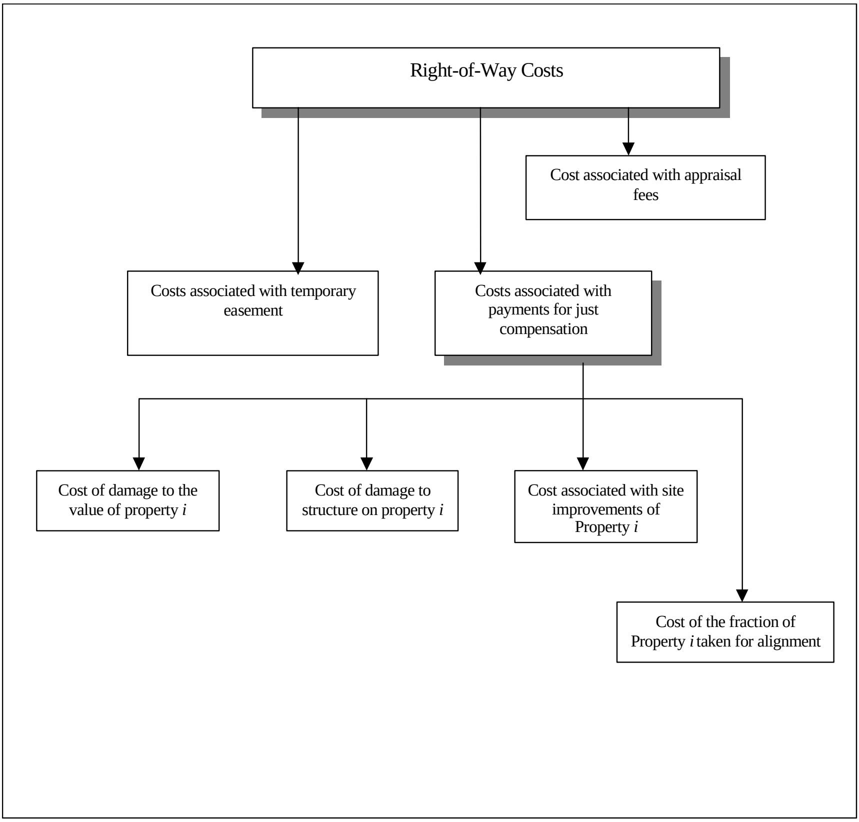

Right-of-Way Costs
Cost associated with appraisal fees
Costs associated with temporary easement
Costs associated with payments for just compensation
Cost of damage to the value of property *i*
Cost of damage to structure on property *i*
Cost associated with site improvements of Property *i*
Cost of the fraction of Property *i* taken for alignment

Figure 10.11: Components of right-of-way costs.

Highway agencies are required to pay just compensation to the property owner if a property is affected by the new highway (IRWA, 1990). In many cases the remainder of the property becomes unusable, requiring the relocation of the owner. Compensation to the owner may include payments equivalent to the cost of the new property and improvement as well as the transportation cost. Some jurisdictions treat transportation cost associated with relocation as an *inconsequential fee* (IRWA, 1990), which is not compensable. Since the Maryland SHA pays for transportation, this cost is also included in our formulation.

10.4.5.2 Model for right-of-way cost

To develop a model for computing right-of-way cost, assume that a property P_i is intersected by an alignment as in Figure 10.12. The section of the proposed alignment intersected by the property has a centerline length of l_i.

The area of the property is A_i (ft^2) and the fractional area of the property taken by the alignment is A_{il}. The shaded areas are the temporary easement areas. In general, the easement areas on both sides of the alignment may be assumed to be approximately equal and the total easement area can be expressed as:

$$A'_{TE_i} \cong 2 \times A_{TE_i} = 2 \times l_i \times w_{TE_i}, \qquad (10.3)$$

where w_{TE_i} = width of the temporary easement in feet.

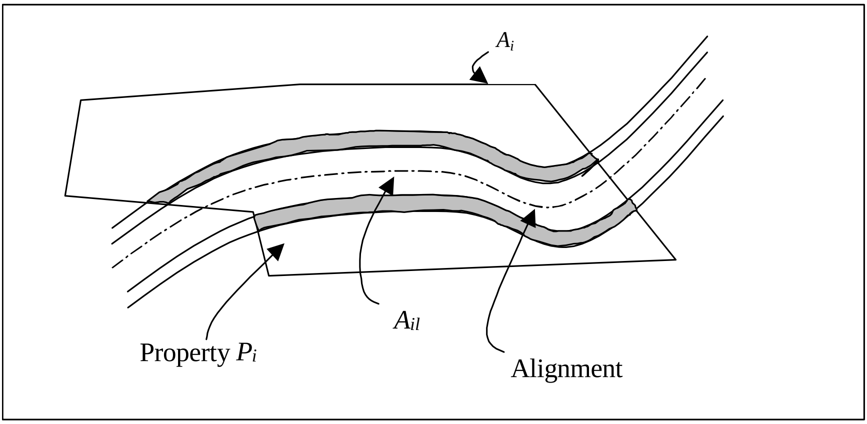


Figure 10.12: A property intersected by a highway alignment.

If C_i is the before value of the property, A_i is property area, and the intersection results in n pieces of the property (excluding the alignment itself), the cost of the j^{th} remainder piece, C_{ij} ($) is:

$$C_{ij} = C_{ui} \times A_{ij}, \quad \forall \ j \in n, \qquad (10.4)$$

where $C_{ui} = C_i / A_i$ = unit cost of property (\$/ft^2);
A_{ij} = fractional area of the j^{th} fraction of the property.

If the unit cost for temporary easement is x% of the unit cost of the property, then the total cost of the temporary easement is:

$$C_{TE_i} = \frac{xC_{ui} A'_{TE_i}}{100}. \tag{10.5}$$

In Maryland the unit cost for temporary easement is assumed to be 10% of the unit cost of the property.

The just compensation depends on *highest and best use* of the remainder of the property (IRWA, 1990). The just compensation for property *i*, C_{JC_i}, can be expressed as

$$C_{JC_i} = C_{DP_i} + C_{DS_i} + C_{SI_i} + C_{F_i}, \tag{10.6}$$

where C_{DP_i} = cost of damage to the value of property i
C_{DS_i} = cost of damage to structures on property i
C_{SI_i} = cost associated with site improvements of property i
C_{F_i} = cost of the fraction of property i taken for the alignment.

The before values of properties and structures on them (if any) can be found in a GIS. The after values of properties and structures depend on the manner in which they are affected as well as a number of other factors such as changes in land use pattern after the construction of the new highway and number of trips generated. The land use generally depends on the zoning established by the local jurisdictions. It is unlikely that the zoning will change after the construction of the highway. For example, if a property is zoned “residential,” it should probably remain residential after the construction of the highway. The zoning should influence number of trips attracted to the new highway. The after value may be correlated to the number of trips or the annual vehicle miles of travel (AVMT) on the new highway. The damage to properties may be correlated to their area, unit cost, and cost of structures on them. Here we formulate the after value of properties based on the manner in which they are intersected by the new highway.

If the shape of the remaining pieces of properties is irregular or if their area is less than the acceptable value (which depends on zoning type) then they may be deemed unusable. The highway agencies must pay for these unusable pieces. In order to select a usable shape, a compactness analysis is performed. In this analysis the ratio of area (A) and square of perimeter (P^2) determine the compactness of the shapes of the remaining pieces of properties. A larger A / P^2 value represents a more compact shape.

If the minimum acceptable area and A/P^2 values for zoning type *k* needed for the usability are X_{ik} (ft^2) and Y_{ik} (dimensionless), respectively, then damage to the property can be expressed as:

$$C_{DP_{ik}} = \begin{Bmatrix} (C_{F_{ik}} + \sum_j A_{ijk} \times C_{uik}) \quad if \ A_{ijk} < X_{ik} \ and \ \dfrac{A_{ijk}}{P_{ijk}^{\ 2}} < Y_{ik}, \forall j \in n \\ C_{F_{ik}} \quad otherwise \end{Bmatrix}, \qquad (10.7)$$

where $C_{F_{ik}}$ = cost of the fraction of the ith property taken by the alignment in zone k

A_{ijk} = area of the jth fraction of the ith property in zone k

C_{uik} = unit cost of the ith property in zone k

P_{ijk} = perimeter of the jth fraction of the ith property in zone k.

The values of X_{ik} and Y_{ik} depend on zoning. For example, the value of X_{ik} for properties commercially zoned may be higher than for properties residentially zoned.

If structures on the property are damaged, their damage can be expressed as:

$$C_{DS_i} = C_{S_i} + (A_{il} \times C_{ui}) + C_{ti}, \qquad (10.8)$$

where C_{S_i} is the value of the structure and C_{ti} is the transportation cost associated with relocation. An algorithm based on compactness analysis is developed next.

Algorithm 10.2 Compactness analysis

```
Step 1: Compute area A and perimeter P of all pieces resulting due to
        intersection (see, Figure 10.13)
Step 2: Compute fractional cost of the property taken by the alignment, $C_{F_i}$
Step 3: Compute $\left(A/P^2\right)_j$ ,  $j = 1,2$ , for remainder pieces 1 and 2
Step 4: Compute fractional costs $C_{ij}$, $j = 1,2$ of remainder pieces 1 and 2
Step 5: Initialize $sum = C_{F_i} + C_{S_i}$ , where $C_{S_i}$ is the value of the structure on the
        property
Step 6:
         $for \ j = 1; \ j \le 2; \ j++$
        {
        If $A_j < X$  $and$  $\left(A/P^2\right)_j < Y$ then $sum = sum + C_{ij}$;
        else
            $sum = sum$;
        }
        end
        Note: X and Y have user specified value
```

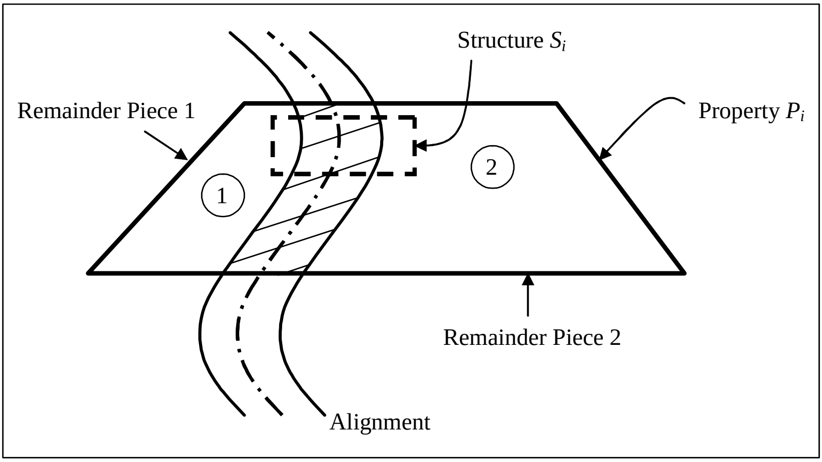


Figure 10.13: An example of compactness analysis.

10.4.5.3 Estimating appraisal fees

The costs associated with appraisal are obtained through a regression analysis using actual cost data for a recently completely highway expansion project for Route 458 in Prince George's County, Maryland. Data from 13 properties are used for the analysis. The resulting regression model is:

$$C_{AF} = 148.6 - 1.27A_F + 0.03C_F + 0.005C_S - 0.000055C_{TB}, R^2 = 0.92\,. \quad (10.9)$$

The transportation cost depends on the round trip distance between the old and new property. If the one-way distance is d_i and the unit transportation cost in \$/mi is K_{ti}, then the transportation cost associated with relocation is

$$C_{ti} = 2 \times t_i \times K_{ti} \times d_i\,, \quad (10.10)$$

where t_i is the number of round-trips required for relocation whose value will depend on the detailed assessment of structure on the property.

For properties commercially or residentially zoned, the cost associated with site improvement C_{SI_i} depends on costs associated with driveway improvements. If $j'=1,..., n'$ are the remainder pieces that are useful after the property is intersected by the alignment (i.e., their area $A_{ij'} > X_i$), $w_{ij'}$ and $l_{ij'}$ are the widths and lengths of their driveways and $K_{P_{ij'}}$ are their unit pavement cost (\$/ft^2), then

$$C_{SI_i} = \sum_{j'=1}^{n'} K_{P_{ij'}} w_{ij'} l_{ij'}\,. \quad (10.11)$$

For properties agriculturally zoned, the cost associated with site improvements can be neglected.

10.5 Case studies with real maps using GIS

We perform a number of case studies to demonstrate the effectiveness of the GIS approach. Those studies are discussed below.

10.5.1 Baltimore county example

In this example a study region located in Baltimore County is chosen to test the applicability of the proposed approach to real maps. The map of the study region is obtained from MDProperty View, which is shown in Figure 10.14 and Figure 10.15. The map is digitized and the attribute table of the digitized map is modified following procedures developed in preceding sections. Figure 10.16 shows the digitized cost map, in which darker shades represent higher cost parcels.

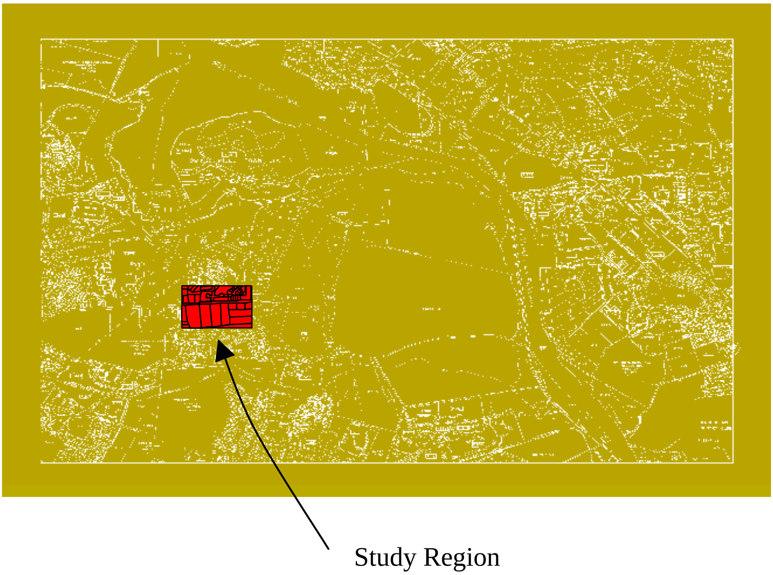


Figure 10.14: Study region from Baltimore county.

The optimized alignment obtained at the 500th generation is shown in Figure 10.17. It is notable that the optimized alignment is able to avoid expensive properties.

10.5.2 Talbot county example

The proposed integrated model is used to optimize horizontal alignments for a new 24 feet wide 2-lane road with 6 feet shoulders in Talbot County, Maryland (Figure 10.18).

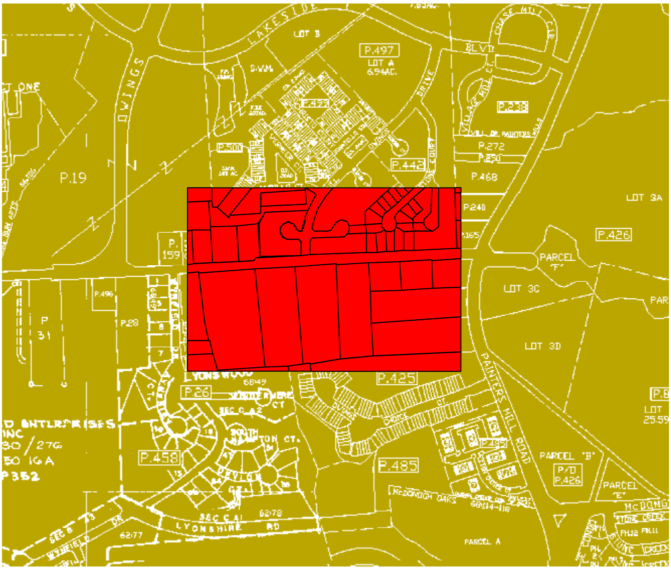


Figure 10.15: Zoomed image of the study region.

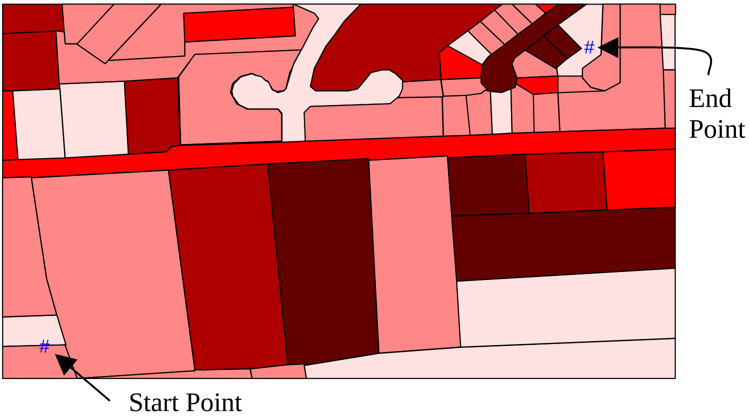


Figure 10.16: Cost map of the study region.

The proposed road is to be built as a bypass between given end points. The *X* and *Y* coordinates of the search space range from 1035893 to 1037690 and from 312906 to 314055, respectively. The coordinates of start and end points are (1036782, 313042) and (1036900, 314022), respectively. The Euclidean distance between the start and end points is 987.08 feet. The geographical shapes with darker shades represent higher cost regions. The largest darkest region represents the Chesapeake Bay, which is assigned a unit cost value of \$1000/ft^2. The search space shown in Figure 10.8 represents a digitized map with a revised attribute table. The area, perimeter, and unit cost fields of the attribute table are shown in Table 10.2. For brevity, entries for only selected records are shown. The Chesapeake Bay is represented by parcel 11, which occupies an area of 7,92,853.866 ft^2 in the study region.

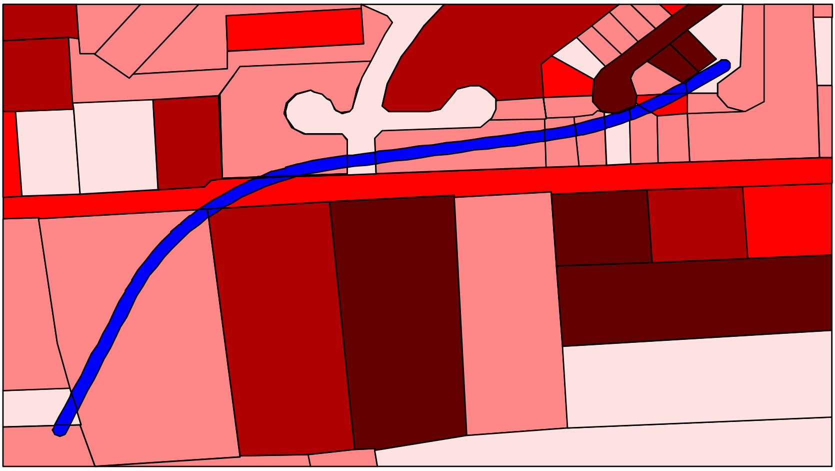


Figure 10.17: Optimized alignment at the 500th generation for the Baltimore county case.

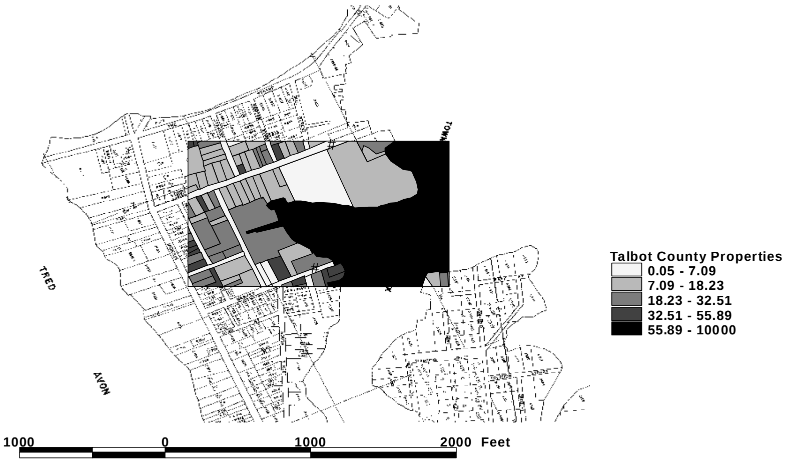


Figure 10.18: Application of the integrated model to a section of Talbot County, Maryland.

Table 10.2: Geographical data for the Talbot County example.

PARCEL	AREA (ft^2)	PERIMETER (ft)	UNIT COST ($/ft^2)
1	1,750.746	200.879	0.05
2	6,953.033	387.286	21.19
3	7,647.667	390.809	36.23
4	5,750.542	355.489	16.29
5	2,839.147	300.816	21.13
6	28,445.845	712.977	8.08
7	739.828	245.127	28.49
8	6,461.475	339.195	28.49
9	8,988.129	376.732	11.80
10	2,679.256	239.025	6.86
11	7,92,853.866	5,506.797	1000.00
.	.	.	.
.	.	.	.
.	.	.	.
90	3,217.124	298.718	15.02
91	2,293.361	208.530	0.05
92	4,646.068	319.049	16.32
93	26,369.512	683.263	20.06
94	32,434.890	894.927	21.00
95	54,714.061	2,459.549	0.05
96	31,870.884	1748.690	0.05
97	12,1634.435	2040.139	21.50

10.5.2.1 Compactness analysis

In order to demonstrate the effectiveness of compactness analysis, right-of-way costs were computed using Eq. (10.2). The right-of-way costs were also computed without compactness analysis, i.e. considering only the fractions of properties taken by the alignment. Recall from the discussion of compactness analysis (section 10.4.5) that values of affected structures and remainder pieces of properties are considered in this analysis, resulting in higher right-of-way costs. A cost map for existing structures (i.e., houses, office spaces, etc.) in the study region is shown in Figure 10.19. The darker shades represent expensive houses. The ocean front and existing roads and streets have light shades since they do not contain any structure. A revised attribute table is created in GIS for compactness analysis. The attribute table has several new fields such as cost of structure, ratio of area and perimeter squared etc. The revised attribute table for this example is shown in Figure 10.20. The land-use types in the study region are either residential or commercial. The numbers "1" and "2" are used in the attribute table to represent the residential and commercial properties, respectively. Existing roads and the ocean front are assigned a value of "3."

The difference in right-of-way cost (expressed as percentage) is plotted over successive generations in Figure 10.21. It is observed that compactness analysis results in higher right-of-way cost indicating that values of unusable properties and structures are considered in the cost computation. At the 28th generation the difference is about 11%, which is quite significant. It is notable that the difference in right-of-way cost with and without compactness will vary from one case to another. In dense urban areas with small land parcels and expensive houses the compactness analysis will result in significantly higher right-of-way costs.

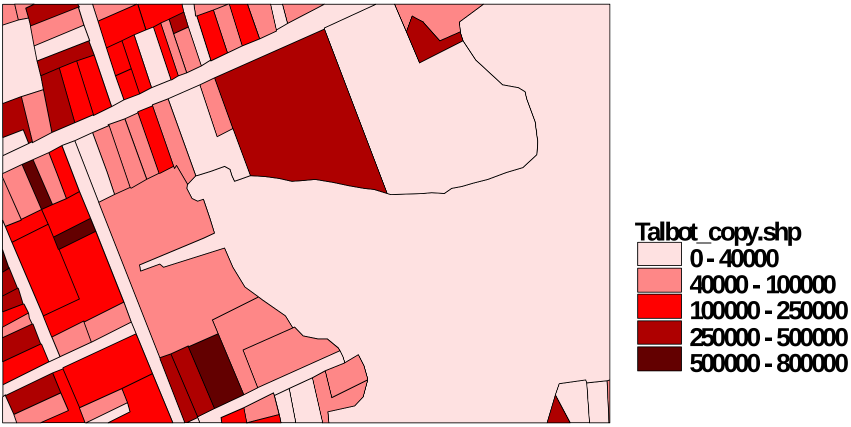
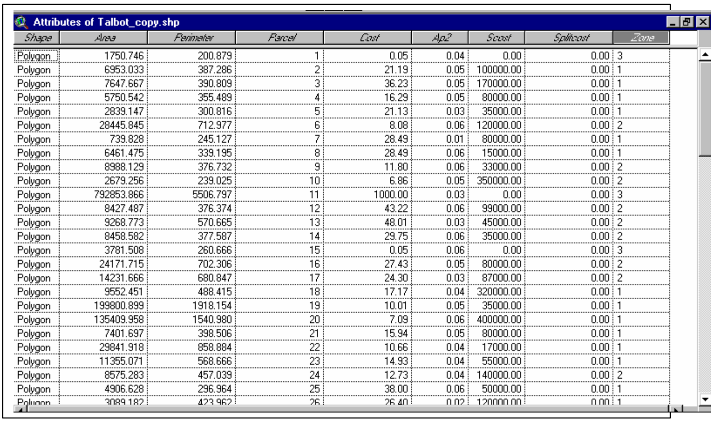


Figure 10.19: Cost of structures in the study region.



Shape	Area	Perimeter	Parcel	Cost	Ap2	Scost	Splitcost	Zone
Polygon	1750.746	200.879	1	0.05	0.04	0.00	0.00	3
Polygon	6953.033	387.286	2	21.19	0.05	100000.00	0.00	1
Polygon	7647.667	390.809	3	36.23	0.05	170000.00	0.00	1
Polygon	5750.542	355.489	4	16.29	0.05	80000.00	0.00	1
Polygon	2839.147	300.816	5	21.13	0.03	35000.00	0.00	1
Polygon	28445.845	712.977	6	8.08	0.06	120000.00	0.00	2
Polygon	739.828	245.127	7	28.49	0.01	80000.00	0.00	1
Polygon	6461.475	339.195	8	28.49	0.06	15000.00	0.00	1
Polygon	8988.129	376.732	9	11.80	0.06	33000.00	0.00	2
Polygon	2679.256	239.025	10	6.86	0.05	350000.00	0.00	2
Polygon	792853.866	5506.797	11	1000.00	0.03	0.00	0.00	3
Polygon	8427.487	376.374	12	43.22	0.06	99000.00	0.00	2
Polygon	9268.773	570.665	13	48.01	0.03	45000.00	0.00	2
Polygon	8458.582	377.587	14	29.75	0.06	35000.00	0.00	2
Polygon	3781.508	260.666	15	0.05	0.06	0.00	0.00	3
Polygon	24171.715	702.306	16	27.43	0.05	80000.00	0.00	2
Polygon	14231.666	680.847	17	24.30	0.03	87000.00	0.00	2
Polygon	9552.451	488.415	18	17.17	0.04	320000.00	0.00	1
Polygon	199800.899	1918.154	19	10.01	0.05	35000.00	0.00	1
Polygon	135409.958	1540.980	20	7.09	0.06	400000.00	0.00	1
Polygon	7401.697	398.506	21	15.94	0.05	80000.00	0.00	1
Polygon	29841.918	858.884	22	10.66	0.04	17000.00	0.00	1
Polygon	11355.071	568.666	23	14.93	0.04	55000.00	0.00	1
Polygon	8575.283	457.039	24	12.73	0.04	140000.00	0.00	2
Polygon	4906.628	296.964	25	38.00	0.06	50000.00	0.00	1
Polygon	3089.182	423.962	26	26.40	0.02	120000.00	0.00	1

Figure 10.20: Revised attribute table.

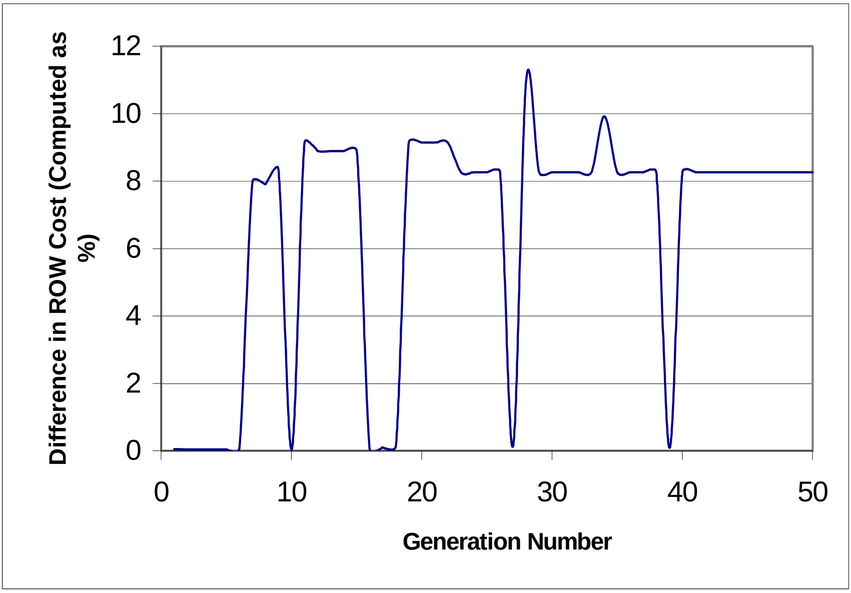


Figure 10.21: Effects of compactness analysis.

10.5.2.2 Analysis of optimized results

The total cost of alignment (including supplier and user costs) at generation 1 is $44.9 million. The improvement in the cost of the best alignment becomes negligible long before the search is stopped at the 100th generation. The solutions for the entire population over 100 generations are plotted in Figure 10.22. The minimum objective function value at successible generations is also plotted, as shown in Figure 10.23. Figure 10.24 and Figure 10.25 show the mean and standard deviation of the solutions over successive generations.

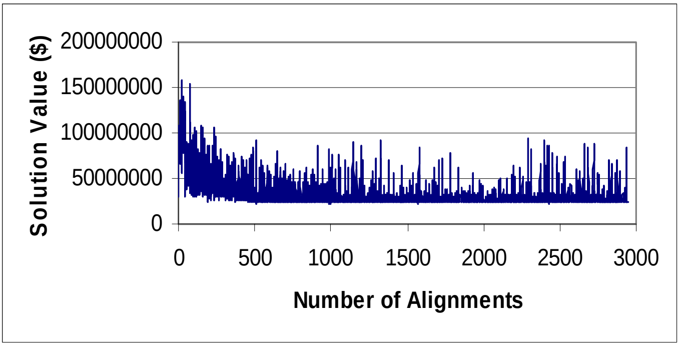


Figure 10.22: Plot of candidate alignments for the entire population.

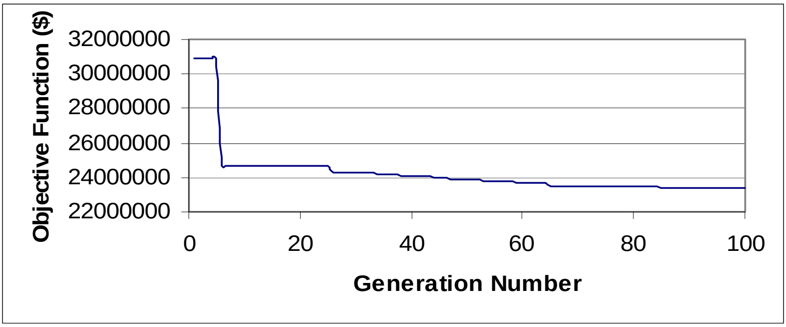


Figure 10.23: Changes in the objective function value through successive generations.

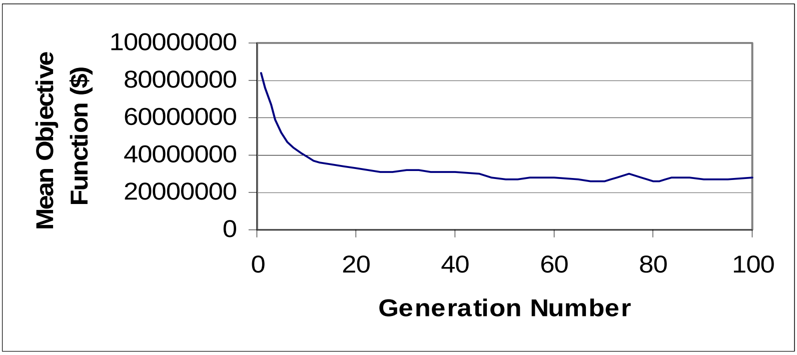


Figure 10.24: Mean objective function over successive generations.

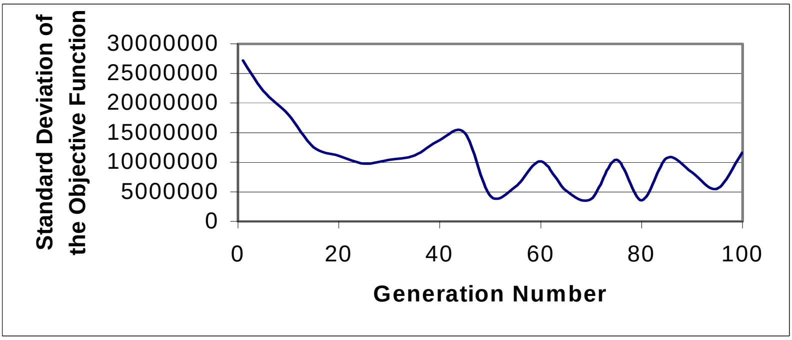


Figure 10.25: Standard deviation of objective function over successive generations.

Figure 10.23 implies that the changes in objective function value becomes very small after about 64 generations. Figure 10.24 shows that the average value of the objective function generally decreases over successive generations. Figure 10.25 indicates that while there is some variation among population members at later generations, the variation tends to decrease over successive generations.

The optimized alignment obtained at the 100th generation (Figure 10.26) has circular curves that satsify the AASHTO minimum radius requirements (AASHTO, 2001) for safe movement of traffic at the specified design speed. It also avoids high cost areas. The total optimized alignment length is 1,248.78 feet, its total cost is $23.43 million, and its location-depedent cost is $0.90 million. The total cost function used in optimization consists of five costs, i.e., length-dependent costs (e.g., construction and maintenance costs) and location-dependent costs (e.g., right-of-way cost), which are classified as supplier cost and vehicle operating, travel-time, and accident costs, which are classified as user costs. In the optimized case these costs are $1.54, 0.90, 0.99, 17.31, and 2.69 (in millions), respectively. Thus the total supplier cost over the design life of that road section is $2.44 millions. In a more complex geographic space, with greater land use variability, the optimized alignment would be more winding.

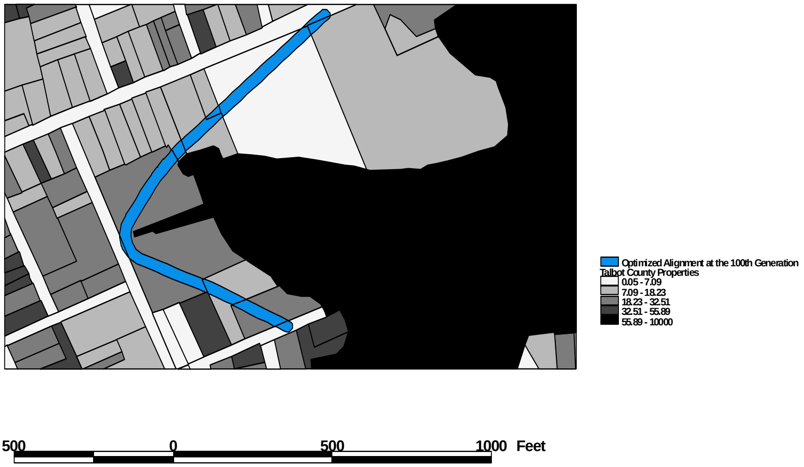


Figure 10.26: Optimized alignment for the Talbot County case obtained at the 100th generation.

10.5.3 Cecil county example

This case is intended to demonstrate how the wetlands and floodplains are considered in the model. For this purpose a real map from Cecil County, Maryland having a number of floodplains and wetlands is used. While the model was capable of handling cases with complex topography and land use, it was difficult to obtain such cases from real databases for Maryland. Figure 10.27 shows the region over which the alignment of a proposed highway is being optimized. The *X* and *Y* coordinates of the region range from 340889 to 346764 and from 181058 to 185189, respectively. The darker shades of land parcel in the map represents higher unit land cost for building a highway. The Great Bohemia Creek runs eastward near the lower end of the map, where location-dependent unit costs are relatively high since expensive bridges would be needed. In addition, some wetlands and flood plains in which the alignment is prohibited are also displayed in a darker shade to represent high environmental impacts. The map has been digitized and preprocessed as discussed in previous sections.

Maryland (MD) Route 213 crosses the Great Bohemia Creek at the lower left corner of the map (Bohemia River Bridge) and intersects MD Route 310 about 1.86 miles north of the river. On the upper-right corner of the map, MD Route 310 intersects MD Route 342. Any vehicles traveling between the Bohemia River Bridge and Route 342 must go through the intersection of Route 213 and Route 310. It is assumed that the traffic is very heavy during peak hours and causes serious congestion at this intersection. Thus, another road is considered to divert traffic between Bohemia River Bridge and the intersection of Routes 310 and 342. For this purpose, the proposed models and algorithms are used to optimize the alignment of the new highway. It can be seen from Figure 10.28 that the straight alignment connecting the start and end points of the highway will go through several high cost land parcels. Therefore the optimal alignment must deviate from the straight line.

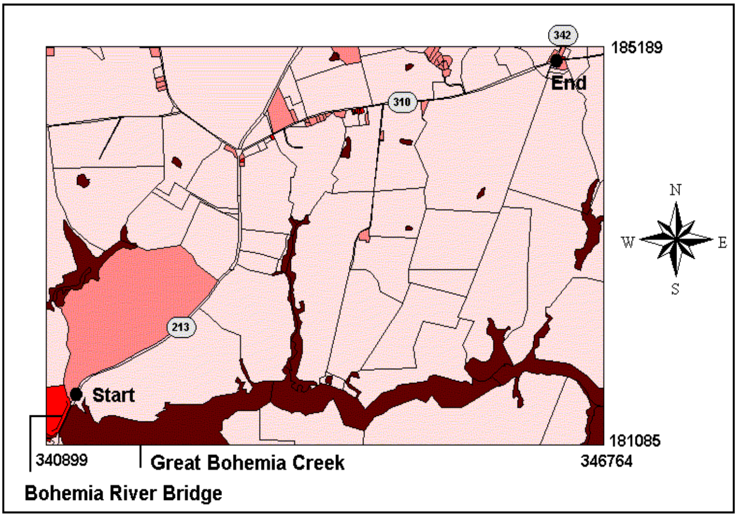


Figure 10.27: Study region for the Cecil County case study.

Here we want to run the genetic algorithms to optimize the alignment and assess the solution. We assume that no particular solution based on engineers' judgments is available for the initial population. Thus, the entire initial population is automatically generated from the program. For this problem, the total number of generations is set to 500 and the number of intersection points is 10 because the land use patterns are not too complex. Note that in regions with much land variability the number of intersection points necessary to obtain a precise solution may be more. To evaluate right-of-way cost for a candidate alignment, the coordinates along the alignment are passed to ArcView^{TM} GIS and the result is sent back to the GA optimizer. The communications between ArcView^{TM} GIS and the proposed GA are bi-directional and programmed so that the process is fully automated by using Procedure 10.3.

The optimized horizontal alignment obtained from the program is shown in Figure 10.28. The alignment is quite straight and smooth. It also satisfies the minimum radius requirement and avoids high environmental impact areas. The total alignment length is 2,132.65 feet with total cost equal to $\$2.99 \times 10^7$. The figure shows that a portion of the alignment almost coincides with the existing MD Route 213, whereas the other section just bypasses the flood plain along the branch of the Great Bohemia Creek and a wetland on the south of MD Route 310. Based on the results, it seems appropriate to add two lanes to existing Route 213 until it diverges from the new two-lane alignment. The change in the objective function value over number of generations is plotted in Figure 10.29. The graph shows that the decrease in the objective function value becomes almost negligible after about 170 generations.

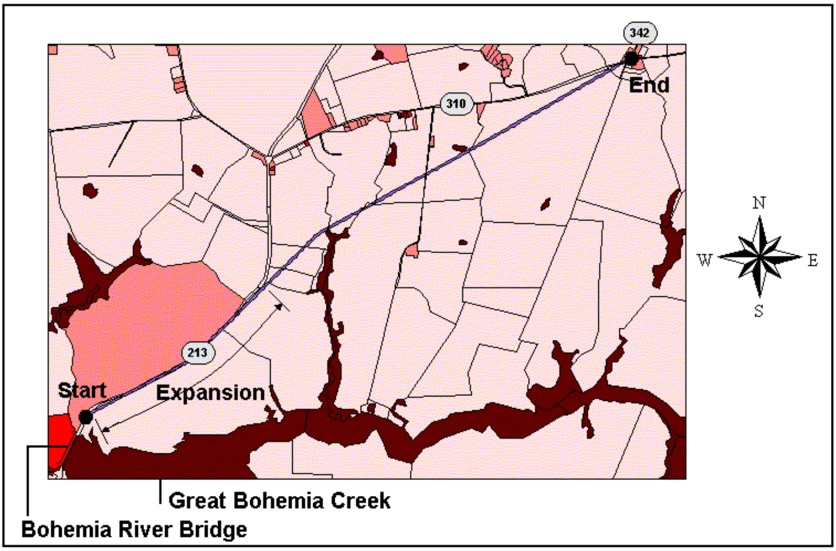


Figure 10.28: Optimized alignment for the Cecil County case study.

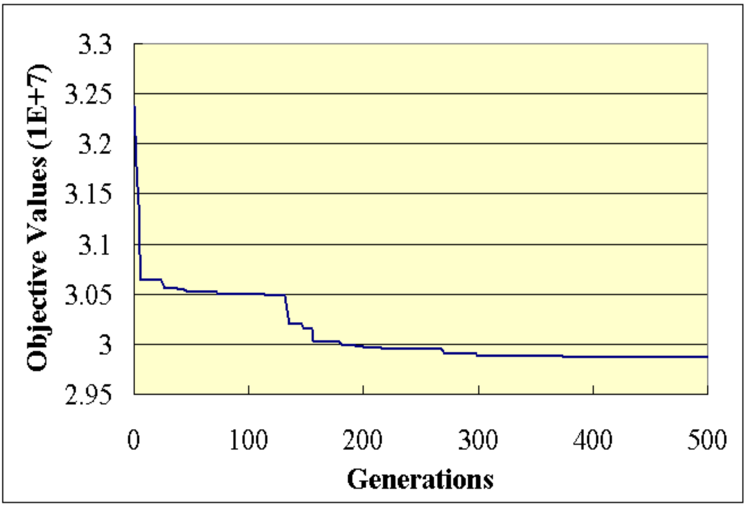


Figure 10.29: Changes in objective function value for the Cecil County case study.

10.5.4 Brookeville bypass project example

The above cases while demonstrating the effectiveness of the developed GIS-based alignment optimization model, are artificial case studies with real geographical maps. Here we apply the developed model to a real project for the Maryland State Highway Administration (MSHA). The MD 97 Brookeville Bypass project is located in the area of Brookeville, Maryland. The project area is located near the town of Brookeville in Montgomery County, approximately ten miles south of I-70 and three miles north of MD 108 and listed on the National Register of Historic Places as a Historic District. MD 97 is an arterial highway providing a direct north-south route between the Pennsylvania state line and Washington D.C., which serves commuter traffic traveling through Carroll, Howard, and Montgomery Counties.

10.5.4.1 Project issues and purpose

According to the previous study for Brookeville bypass project of the MSHA, three issues are relevant in the project area. Table 10.3 summarizes the project needs in Brookeville area. There are safety concerns, since the crash rate in Brookeville (1996 to 1999) exceeds the statewide average crash rate. MD 97 is a two-lane undivided roadway with little to no shoulder and its right-of-way width is not constant within the project area. In addition, due to irregularly posted speed limits and limited sight distance, travel speed in the project area is also variable. There are no exclusive turn lanes along MD 97 in the project area. According to the growth forecast, it is expected that planned residential development in the Brookeville area and to the north will generate increased traffic. The purpose of Brookeville bypass project is to reduce the increasing traffic volumes from the town of Brookeville, improve traffic operation and safety on existing MD 97, and preserve the historic character of the town.

Table 10.3: Issues regarding MD 97 in the Brookeville project area.

	Issues
Access	No access control No exclusive turn lanes
Safety	Inconsistent roadway width Irregular speed limit Limited sight distance Inconsistent travel speed High crash rate above the statewide average
Traffic	Expected traffic volume increasing
Socio-Environment	All traffic is currently routed through a historic district

10.5.4.2 Data preparation

Three major data preprocessing works (horizontal and vertical map digitization and tradeoff in map representation) were conducted before evaluating possible alignments with the Highway Alignment Optimization (HAO) model. Figure 10.30 presents the procedure used in applying the HAO model to the Brookeville bypass project. Details on each data preparation process are described in the following sections.

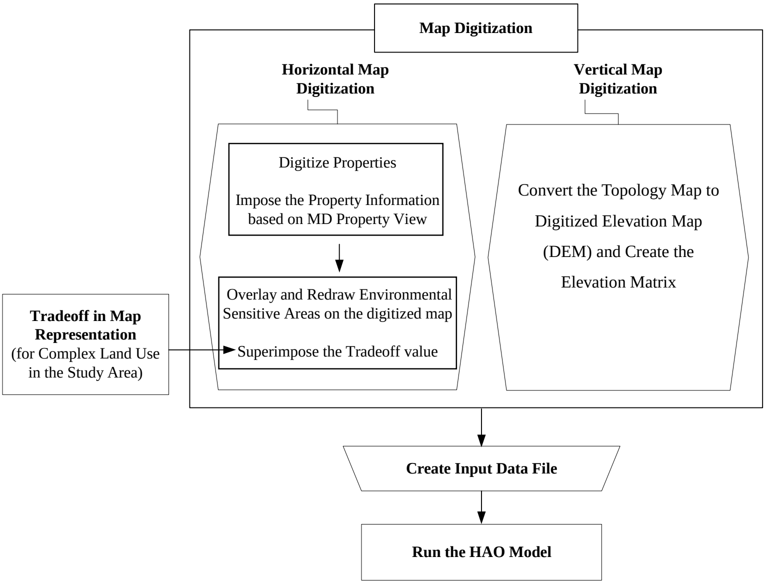

Map Digitization
Horizontal Map Digitization
Vertical Map Digitization
Digitize Properties
Impose the Property Information based on MD Property View
Overlay and Redraw Environmental Sensitive Areas on the digitized map
Superimpose the Tradeoff value
Convert the Topology Map to Digitized Elevation Map (DEM) and Create the Elevation Matrix
Tradeoff in Map Representation
(for Complex Land Use in the Study Area)
Create Input Data File
Run the HAO Model

Figure 10.30: Procedure for the HAO model application.

10.5.4.3 Estimated working time

To reduce working time for preparing geographical information, a study area was defined around the town of Brookeville. Maryland's GIS database from MDProperty View and the Microstation base maps for Brookeville area were used to construct the study area. Property boundaries for the study area, including environmentally or socio-economically sensitive regions, were digitized with the Microstation base map and associated geographic databases containing relevant information (such as, land area, zoning, and land cost) of the study area are referred from MDProperty View. Thus, the study area became the search space of the HAO model application. As shown in Table 10.4, the data preparation time for the HAO application in the Brookeville bypass project was about 250 person hours. Most of that time was spent on the map digitization work for the study area. Model computation time varies depending on input parameters (mainly generation number) and the complexity of land use in the study area.

Table 10.4: Estimated working time.

Tasks			**Working time**
Data preparation time	Horizontal map digitization	Digitize properties	50 hrs
		Impose property Cost	80 hrs
		Tradeoff in map representation	95 hrs
	Vertical map digitization	Create DEM matrix	20 hrs
	Create an input data file		7 hrs
Model computation time on Pentium IV 3.0 GHZ with 512 MB RAM			4.5 to 6.5 hrs for 300generations

For horizontal map digitization, the Microstation base maps which store boundaries of environmentally sensitive areas, such as wetlands, floodplains, and historic resources were used to digitize properties in the study area of Brookeville. This task took about 50 hours. After this task, the property cost was imposed to the digitized properties based on MDProperty View. A relatively longer time (approx. 80 hours) was spent on this step because we manually imposed property information on the digitized map from MD Property View. After the previous two steps, superimposition of tradeoff values for the existing sensitive regions in the study area was applied on the digitized map. This step was quite lengthy, requiring approximately 95 hours.

For vertical map digitization, we obtained a Microstation contour map for Brookeville from the SHA, and converted it to a Digitized Elevation Map (DEM) that provides elevations with grid as a base. This task took about 20 hours; however, it should be noted that if the projection of the Microstation base map and that of MDProperty View are same, the working time for vertical map digitization would be reduced to just using the DEM file for the Brookeville area from the web site: http://data.geocomm.com/dem/demdownload.html.

10.5.4.4 Horizontal map digitization

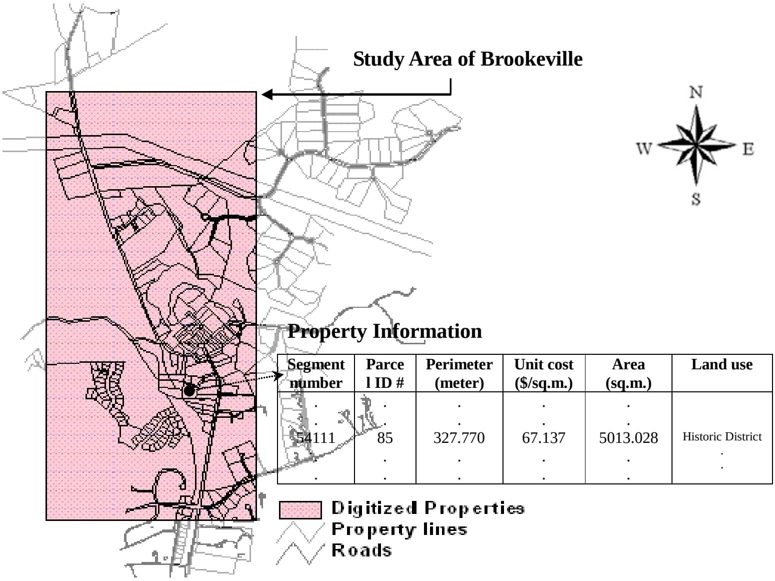
The purpose of horizontal map digitization is to reflect complex land uses in the study area on the GIS digitized map and to obtain detailed right-of-way costs for the proposed alignments. Horizontal map digitization mainly consists of two steps (see Figure 10.31). For this project, we first digitized properties of the study area and next imposed the associated property information to the previously digitized properties. After this step, the environmentally sensitive areas (such as wetlands and historic sites) were overlaid and redrawn onto the digitized map. Tradeoff values for the different land use characteristics were then superimposed.

10.5.4.5 Digitizing properties

For horizontal map digitization, we first digitized properties in the Brookeville study area using the Arc View GIS 3.2 software. In this step, each property was regarded as a polygon, which can retain property information as its attributes. Next, the property information, such as land value and land use characteristics were imposed on the digitized properties based on MDProperty View. Figure 10.31 shows a digitized map on which the real property information is assigned. The information assigned on the map includes parcel ID number, perimeter, unit cost, and area of each property. It is noted here that the unit cost is obtained simply by dividing the property value by its area. Among these attributes, unit cost ($/sq.m.) is mainly used for alignment evaluation. Right-of-way cost, length of alignment, and the area taken by the proposed alignments is computed based on the unit cost. As shown in Figure 10.31, we also imposed land use type and segment number, which is recorded on MDProperty View, to the digitized properties. While these attributes are not used in model computation; however, they may help in reducing other working times, such as in superimposing tradeoff values on critical areas and updating property information from the MDProperty View.

10.5.4.6 Handling environmental and socio-economical issues

In order to consider environmentally or socio-economically issued regions to the HAO model application, we overlaid and redrew the control areas on the previously digitized map. The existing land use in the study area of Brookeville is a combination of various land use types. Figure 10.32 presents various land use type of the study area in Brookeville. The land use type of the study area is represented as 10 different land use characteristics on the digitized map; structures (houses and other facilities), wetlands, residential areas, historic places, streams, park within historic district, parklands, floodplains, existing roads, and other properties.



Segment number	Parcel ID #	Perimeter (meter)	Unit cost ($/sq.m.)	Area (sq.m.)	Land use
.	.	.	.	.	
.	.	.	.	.	
54111	85	327.770	67.137	5013.028	Historic District
.	.	.	.	.	.
.	.	.	.	.	.

Figure 10.31: Digitized property cost map.

As shown in Figure 10.32, the study area comprises about 650 geographic entities (including land, structures, road etc.) with given start and end points of the proposed alignment. The search space (2.792 km^2) includes primarily residential areas (0.823 km^2), historic sites (0.297 km^2), parkland (0.273 km^2), and floodplains (0.125 km^2).

Figure 10.33 presents real property cost in the Brookeville study area. The unit property cost for land ranges from 0 to 151 \$/m^2 and structure costs (such as houses and public facilities costs) ranges from \$36,100 to \$1,162,200. The darker land parcels have higher unit costs.

10.5.4.7 Vertical map digitization

In the HAO model the earthwork cost of the proposed alignment was calculated based on an elevation matrix. Thus, preparation of the elevation matrix for the study area was required. We converted the Microstation contour map for Brookeville to a Digitized Elevation Map (DEM) using Arc View GIS 3.2. Figure 10.34 shows the ground elevation of the study area, which is represented as a matrix of 90 × 210 grids. Each grid size is 12 meter × 12 meter and the elevation range in the Brookeville area is 100 to 155 meter. As shown in Figure 10.34, the darker areas represent higher elevations. Floodplains and parklands exist in low elevation areas while Historic District is located in relatively high elevation sites (also see, Figure 10.32). It is noted that finer grid size results in the better precision of earthwork cost as will be seen by the sensitivity analysis presented later.

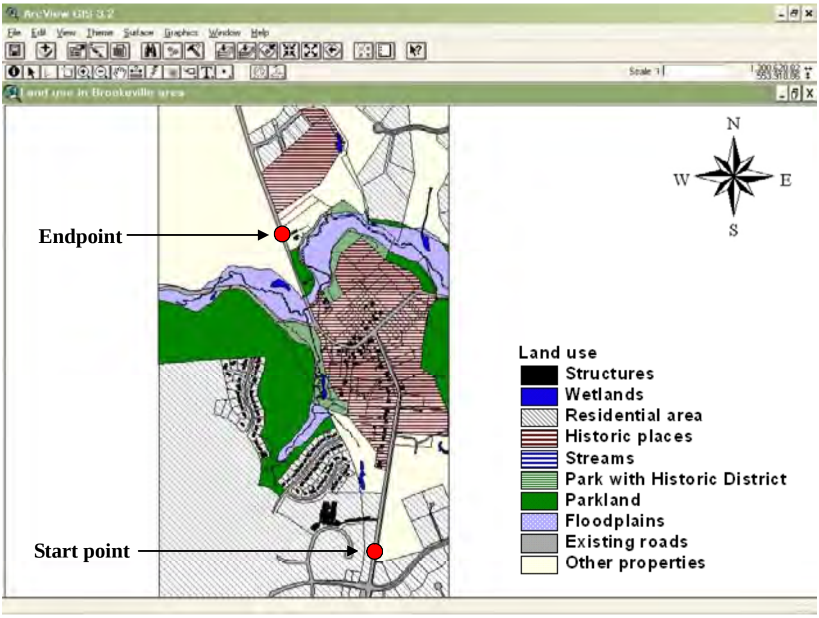


Figure 10.32: Land use of the study area in Brookeville.

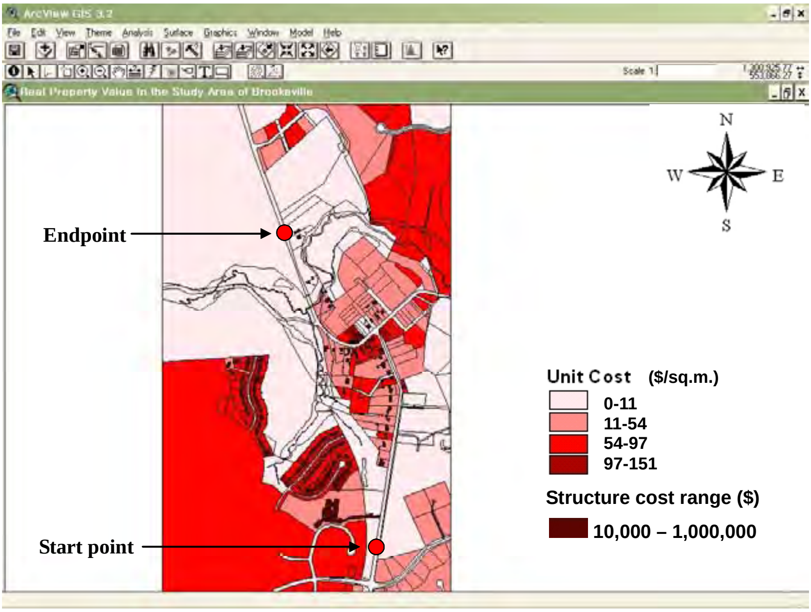
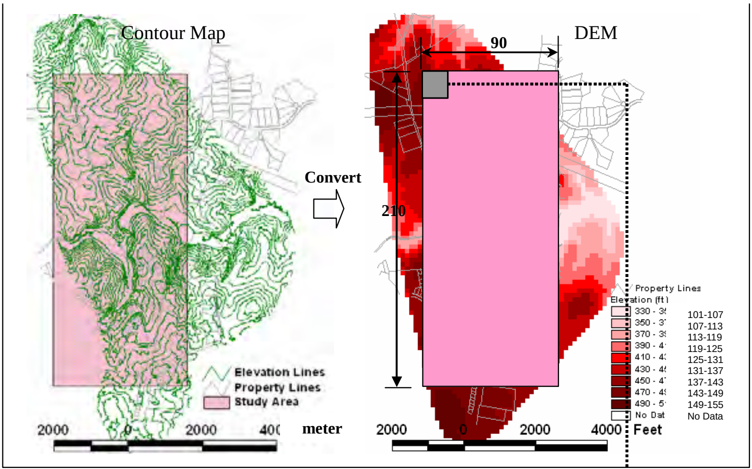


Figure 10.33: Real property value of the study area.

10.5.4.8 Tradeoffs in map representation for environmental issues

When considering roadway construction in a given project area, various geographically sensitive regions (such as historic sites, creeks, public facilities, etc.) may exist. These control areas should be avoided by the proposed alignment and to the extent, its impact to these regions should be minimized.



Sample Grid Evaluation for the Study Area (90 × 210 grids)

Grids	1	2	3	4	5	. . .	90
1	143.3	143.3	142.6	141.4	140.5	. . .	131.7
2	143.3	143.3	143.0	141.7	140.5	. . .	132.3
3	143.3	143.3	143.3	141.7	140.2	. . .	132.6
4	143.3	143.3	142.6	141.1	140.2	. . .	133.2
5	143.9	143.3	142.0	140.5	140.2	. . .	133.8
:	:	:	:	:	:	: : :	:
210	122.8	120.7	118.9	120.4	121.6	. . .	141.4

Figure 10.34: Ground elevation of the study area in Brookeville.

Based on the previous Brookeville transportation study by MSHA, we categorized residential properties, the Longwood Community Center, historic districts, and wetlands as environmentally sensitive primary areas that should not be taken by the new alignment if at all possible. In addition, parklands, floodplains, and streams were considered environmentally sensitive secondary areas, i.e., to the extent possibly their impact should be minimized next to the primarily sensitive area. This requires expressing different implicit cost levels for various environmental factors into the GIS based evaluations, practically. It should be noted that parklands, floodplains, and streams are located between the given start and end points; furthermore, these areas are unavoidably taken by the proposed alignment.

Table 10.5 shows two different types of control areas in the Brookeville study area with respect to their land use characteristics: (i) the control area that the proposed roadway alternatives can avoid, (ii) the area that the proposed alternatives cannot avoid. Type 1 area includes wetlands, historic places, residential areas, Community Center, and other structures. Type 2 area consists of streams, parklands and floodplains, which are unavoidably affected by the alignments.

Table 10.5: Types of control areas in the Brookeville study area.

Type	Control areas	Characteristics
Type 1	Wetlands Historic places Residential areas Site of Community center Structures (Houses, Public Facilities, etc.)	The control area that the proposed alignment can avoid
Type 2	Streams Floodplains Parklands	The area that the proposed alignment cannot avoid

To properly reflect these relevant environmental and socio-economic issues on the GIS map representation, careful tradeoff property values for the different land use types are required, since these values are significantly able to affect the resulting alignment. Thus, penalty costs for type 1 area should be much higher than that for type 2, since type 1 area has primary (i.e., stronger) environmental regions to be avoided whereas type 2 area contains only secondary regions.

Table 10.6 presents the order of magnitude of penalty costs for the various types of control areas. We developed a guideline for the penalty costs based on the maximum unit land cost[1] (151 $\$/m^2$). The idea is to eliminate impacts on type 1 areas and minimize those on type 2 areas, and to encourage the alignments to take other properties (e.g., Montgomery County's reserved areas and existing roads in this study area). For this purpose, we discriminated between type 1 and type 2 areas by assigning 1,510,000 $\$/m^2$ for type 1 areas and 15,100 $\$/m^2$ for type 2 areas (i.e., the penalty to type 1 area is 100 times higher than for type 2). In addition, we particularly differentiated wetlands among type 1 areas by assigning a considerably higher cost (15,100,000$\$/m^2$) since we assumed that wetlands are the most sensitive areas the proposed alignment must avoid. For the same reason, we distinguished streams from type 2 areas by assigning relatively high unit cost (151,000 $\$/m^2$). It is noted that the tradeoff values presented in Table 10.6 were successful in minimizing the control area taken by the proposed alignment.

Table 10.7 presents the list of unit costs, which were finally assigned to the properties for the HAO application in Brookeville bypass project. As stated

[1] Range of unit land cost for the study area is 0-151 $/ sq.m. (See, Figure 10.33)

earlier, these unit costs were mainly used to calculate right-of-way cost, length of alignment, and the area taken by the proposed alignment. Unit costs for group 1 and structure costs for group 7 are extracted directly from MDProperty View. On the other hand, unit costs for group 3 to 6 are the tradeoff values from Table 10.6. These costs were used to avoid taking the control areas, if possible, for the proposed alignments. It is noted here that we assumed the unit cost of the existing roads to be very small (0.27 $/m^2). Figure 10.35 shows a preferred search space of the study area with the tradeoff values presented in Table 10.7.

Table 10.6: Order of magnitude of penalty costs[2].

Type of Control Areas	**Level**	**Magnitude**	**Control Areas**	**Tradeoff Value ($/m^2)**
Type 2	1	100×X	Floodplains, Parklands, Park with Historic Districts	15,100
Type 2	2	1000×X	Streams	151,000
Type 1	3	10,000×X	Historic sites, Residential sites, Community center sites	1,510,000
Type 1	4	100,000× X	Wetlands	15,100,000

Table 10.7: Unit land cost finally assigned to the different land uses.

Group	**Land Use**	**Unit Cost ($/m^2)**	**Note**
1	Other properties	0-151	Real value
2	Existing roads	0.27	Assumed value
3	Floodplains, Parklands, Park with Historic Districts	15,100	Tradeoff value
4	Streams	151,000	Tradeoff value
5	Historic resources, Sites of Residential, and Community Center	1,510,000	Tradeoff value
6	Wetlands	15,100,000	Tradeoff value
7	Structures (Houses, Public facilities, etc.)	36,100-1,162,200 ($)	Real value

[2] X=151 $/ sq.m.: Maximum unit cost for land in the study area of Brookeville

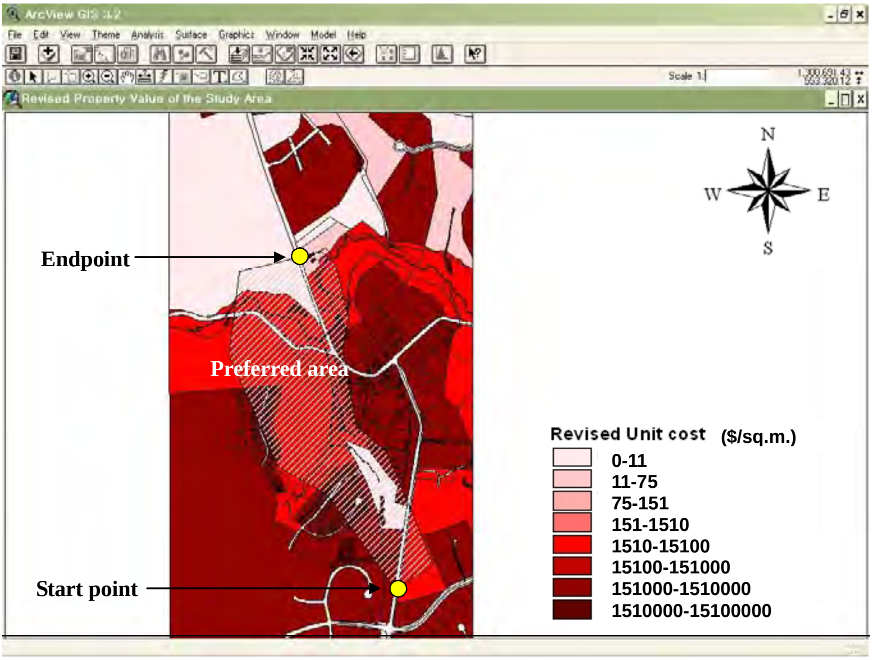


Figure 10.35: Preferred search space for the Brookeville bypass.

10.5.4.9 Application results

Input and output for optimized alignments

To conduct highway alignment optimization with the HAO model, users have to pre-specify some input values, such as proposed alignment width and design speed. Since the optimized alignment varies depending on these inputs, users should carefully determine the input variable values. We specified the start and end points of the proposed alignments to (1295645, 548735, 470) and (1294512, 552574, 407) as a default on the south and north sections of MD 97 in Brookeville, respectively (refer to Figure 10.35). The Euclidean distance between the start and end points is about 1.219 km. The design speed was set at 80 kph. The cross-section spacing, which are used as earthwork computation unit in the HAO formulation, is assumed to be 12 meter. The cross section of the proposed alignment is assumed to represent a 2 lane road with 12 meter width (3.35 meter for lanes and 2.74 meter for shoulders).

Grade separation was the only crossing type of the proposed alignment with the existing Brookeville Road, considered in this analysis. Various user specifiable input variables required in the highway alignment optimization process are described in left hand side of Table 10.8. The input values presented in Table 10.8 were used for analyzing sensitivity to the number of P_i's. These values were also used for sensitivity analyses to the other major key parameters as the baseline values.

Detailed results for the optimized alignments, such as costs breakdown of total earthwork cost per station, and coordinates of all evaluated alignments are provided as HAO model outputs. These results are automatically recorded in different files during program runs. In addition, environmental impacts for the optimized alignment can also be summarized using Arc View’s attribute table after program terminates. Available output results from the HAO model application presented in Table 10.9.

Table 10.8: Baseline inputs used in sensitivity analysis to # of P_i*’s*.

Input variables	**Value**
# of Intersection points (PI’s)	4 to 7
Proposed alignment width	12 meter, 2 lane road (3.35 for lane, 2.74 for shoulder)
Design speed	80 kph
Maximum super-elevation	0.06
Maximum allowable grade	5 %
Coefficient of side friction	0.16
Longitudinal friction coefficient	0.28
Location of start and end points (*X*,*Y*, *Z*)	Start: (1295645, 548735, 470), End: (1294512, 552574, 407)
Cross-section spacing	12 m
Fill slope	0.4
Cut slope	0.5
Earth shrinkage factor	0.9
Unit cut cost	45.5 $/m^3$
Unit fill cost	26 $/m^3$
Cost of moving earth from a borrow pit	2.6 $/m^3$
Cost of moving earth to a fill	3.9 $/m^3$
Unit length-dependent cost	656 $/m
Crossing type with the existing road	Grade separation
Terrain height ranges	100 to 155 m
Unit land value in the study area	0 to 151 $/m^2$
Unit cost of existing road	0.27 $/m^2$
Unit bridge cost	32,800 $/m

10.5.4.10 Description of optimized alignments

Four optimized alignments (Figs. 10.35 and 10.36) are produced here by using the HAO model with different numbers of P_i, s . It is assumed that all of the four alternatives have the same start and end points and cross the Brookeville Road with grade separation. They mainly dominate Montgomery County's reserved area and Reddy Branch Park without affecting any residential property and Brookeville Historic District. Optimized alignments A, B, C, D have 4, 5, 6, and 7 P_i*'s*, respectively. Figure 10.36 and Figure 10.37 show horizontal and vertical alignments of these four alternatives on the Brookeville property cost map. Various output details for optimized alignment B, such as cost breakdown for net total construction, coordinates, information of horizontal and vertical curvatures, and earthwork volume per station can be extensively presented as desired.

Table 10.9: Available output results.

Type of output	**Contents**	**Unit**
Costs breakdown for all searched alignments	Earthwork costs (Ethw cost)	$
	Length-dependent costs (Lnth cost)	$
	Right-of-way costs (Lctn cost)	$
	Penalty costs for gradient (Grad cost)	$
	Penalty for vertical curve (Lnvc cost)	$
	Structure cost (Bridge cost)	$
	Alignment length (Length)	m
Earthwork cost (per station)	Elevation of alignments (Zr)	m
	Cut volume (E_cutting)	m
	Fill volume (E_filling)	m^3
Detailed results for the optimized alignment	PI Index for horizontal and vertical alignment	
	Number of horizontal and vertical curves	No.
	Horizontal curve radius	m
	Length of vertical curves	m
Coordinate of the optimized alignments (*X*, *Y*)		*X*,*Y* coordinates
Environmental impact Summary	Residential relocations	No.
	Affected properties	No.
	Areas affected by the optimized alignment	m^2

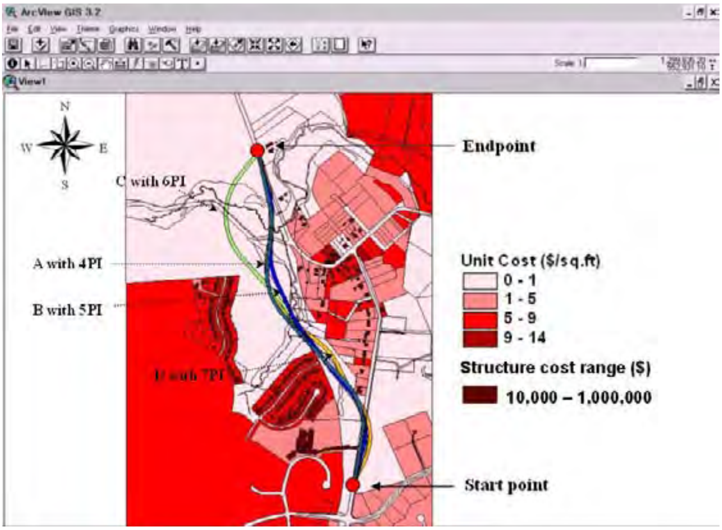


Figure 10.36: Optimized horizontal alignments for different PI's.

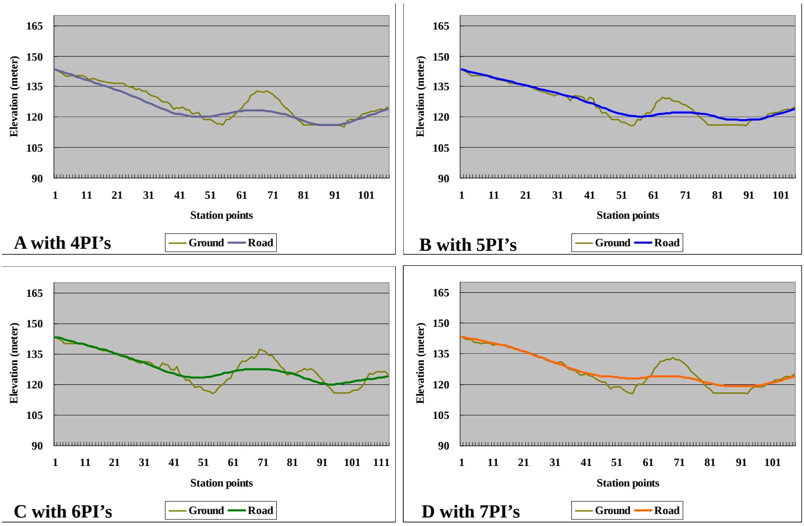


Figure 10.37: Optimized vertical alignments for different PI's.

Sensitivity of optimized alignments to the number of P_i's

Optimizing (roughly) the number of P_i's is quite desirable in applying the HAO model, mainly to reduce the number of curved sections. Moreover, the solution quality (such as the impact of the proposed alignment to the sensitive area and its right-of-way) and computation efficiency of the HAO model differ depending on this number. Therefore, a sensitivity analysis was conducted in this study to explore the preferable number of P_i’*s* between 4 and 7. More than 8 P_i’*s* were not considered in this analysis to avoid too many horizontal curves. Table 10.10 shows the result summary for the sensitivity analysis. Initial construction cost, environmental impacts, length, and model computation time for four different optimized alignments are presented here. The search was conducted over 300 generations, during which about 6,500 alignments were evaluated for each case. A desktop PC Pentium IV 3.0 GHZ with 512 MB RAM was used to run the model. It took a considerable time (about 4.5 to 6.5 hours) to run through 300 generations because the Brookeville study area is quite complex with many properties (about 650 geographical entities). As shown in Table 10.10, none of the four alternatives require any residential relocation and all have similar alignment lengths.

Table 10.10: Sensitivity to number of P_i’*s*.

Optimized alignment	# of PI’s	Initial construction costs ($)	Environmental impact				Length (m)	Computation time (hr)
			The control area taken by alignments (m^2)			Residential relocation (No.)		
			Type 1	Type 2	Sum			
A	4	5,148,404	42.56	6,563.16	6,605.73	0	1,295.97	4.41
B	5	4,629,708	0	5,853.32	5,853.32	0	1,278.33	4.68
C	6	5,956,983	0	7,616.55	7,616.55	0	1,371.37	4.95
D	7	5,220,679	0	5,988.80	5,988.80	0	1,315.18	5.01

Among the four alternatives, the initial construction cost is lowest for optimized alignment B ($4,629,708) and highest for optimized alignment C ($5,956,983). In terms of environmental impact, the sensitive areas taken by the alignment B (5,853 m^2 for total) are also the lowest although the differences are not too great among the four alignments. For type 1 areas, which were previously defined as environmentally primary sensitive regions, optimized alignment A with 4 PI’s affects relatively large amounts of type 1 areas compared to those of the other three alternatives. Alignment A affects 42.56 m^2 of type 1 area (28.41 m^2 for residential area and 14.15 m^2 for Longwood Community Center); on the other hand, the other three optimized alignments do not affect type 1 areas. A detailed environmental impact summary for optimized alignments A to D is presented in Table 10.11. In terms of computation efficiency, Table 10.10 shows that model computation time increases slightly when the number of P_i’*s* increases from 4 to 7. It seems that model computation time is not significantly affected by the number of P_i’*s*. However, it should be noted that computation time still increases with the number of P_i’*s* since additional P_i’*s* generate additional horizontal and vertical curved sections. For instance, the HAO model with 20 P_i’*s* requires over 10 hours computation time with the same inputs shown in Table 10.8. Thus, the HAO users should keep in mind that more P_i’*s* can increase computation burden significantly.

Table 10.11: Environmental impact summary for optimized alignments A to D.

Optimized alignments		**A**	**B**	**C**	**D**
Initial construction costs ($)		5,148,404	4,629,708	5,956,983	5,220,679
Length of the optimized alignment (m)		1295.97	1278.33	1371.37	1315.18
Socio-economic resources	Affected residential area (m^2)	28.41	0	0	0
	Residential relocations (no.)	0	0	0	0
	Affected Community Center (m^2)	14.15	0	0	0
	Affected properties in Historic Districts (m^2)	0	0	0	0
	Affected MC’s reserved area (m^2)	3890.34	4206.03	4205.11	4202.70
	Affected existing roads (m^2)	3635.54	2749.41	1582.06	2342.54
Natural resources	Affected wetlands (m^2)	0.00	0.00	0.00	0.00
	Affected floodplains (m^2)	2159.83	1602.74	1549.75	1382.03
	Affected streams (m^2)	64.12	72.21	58.95	56.71
	Affected parkland in Historic Districts (m^2)	1082.91	1867.34	857.23	1702.67
	Affected parkland (m^2)	3255.71	2310.52	5149.93	2846.87

It should be noted that the initial construction cost in Table 10.10 is systematically underestimated. This cost mainly consists of right-of-way, length-dependent, bridge, and earthwork cost; i.e., other costs required in road construction (such as drainage landscape architecture cost, traffic signal strain poles cost, etc.) and contingency cost are not included. It should also be noted that penalty costs (tradeoff values) for the control areas taken by optimized alignments are not included in the initial construction cost (i.e., the penalty costs are subtracted from the objective function value)[3].

Figure 10.38 shows changes in objective function value over successive generations for four different optimized alignments. It can be seen that most of the improvement is found in the early generations, i.e., there is no great improvement of the objective function after about 60 generations. This indicates that the HAO model can provide reliable (though not optimized results) results quite quickly. It is noted here that the objective function value of optimized alignment A is relatively higher than those of the others. This is because alignment A affects type 1 areas more than those of others, so that more penalties are added to its objective function.

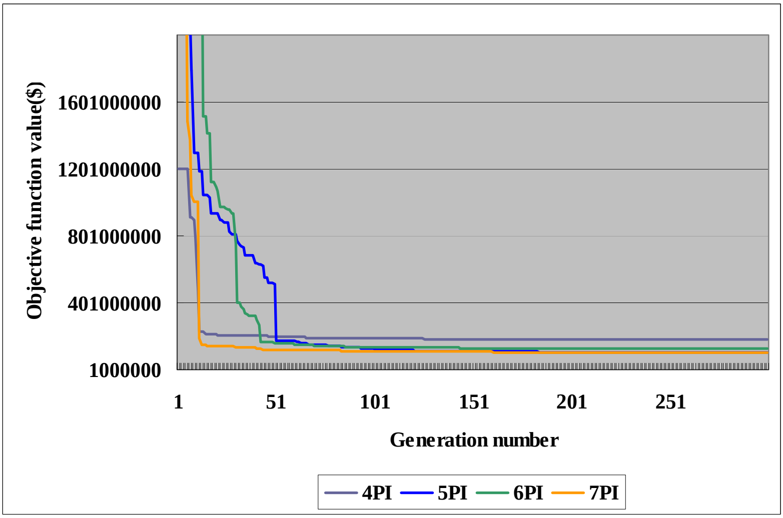


Figure 10.38: Changes in objective function value over successive generations.

10.5.5 Effects of map size

One of the main concerns about the proposed integrated model is its computational efficiency. The integration procedure with a Geographic Information System (GIS) is developed using ArcView GIS software, which is not intended for large-scale iterative applications. The graphical manipulations required for computing the right-of-way cost increase the computer time and memory requirements far beyond those of the genetic algorithm. A 450 MHz Pentium-II computer with 128 megabytes of random access memory (RAM)

[3] The initial objective function used in this study is $C_T = C_R + C_L + C_E + C_S$ and the estimated initial construction cost is $C_T = C_R + C_L + C_E + C_S - C_{penalty}$.

computes the right-of-way cost in about 3 seconds per candidate alignment. This is based on about 300 control points between the start and end points in the search space. About 32 *alignment solutions* per generation are explored. In genetic algorithms searching through an entire generation requires going through a number of candidate alignments whose number depends on the number of *genetic operators* (Jong and Schonfeld, 2003). Genetic operators are used to generate candidate alignments (also known as offspring). A total of 8 genetic operators are used in this example. At each generation after the initial population, each of the eight genetic operators generates four new offspring, for a total of 32.

Thus, with 300 points per alignment and 32 candidate alignments per generation about 1.6 minutes of computation time is required. With 3000 points per alignment the computation time increases to approximately 32 seconds per alignment. Such computation time does not seem to be a problem since optimized alignments do not have to be found in real-time applications. A well optimized solution can be obtained on personal computers (PCs) in a matter of hours.

To gain further insight into the computational efficiency, the following four map sizes were tested:

- a one square mile grid having 100 parcels, i.e., map density (parcels/m^2) = 100
- a one square mile grid having 900 parcels, i.e., map density (parcels/m^2) =900
- a one square mile grid having 10,000 parcels, i.e., map density (parcels/m^2) =10,000
- a one square mile grid having 1 million parcels, i.e., map density (parcels/m^2) =10^6

The test cases were designed so that the computation time with increasing map complexity could be investigated. The proposed optimization model was applied to the four maps (as described above) to investigate the following:

(1) computing time per generation vs. map density
(2) fluctuation in computing time over the number of generations
(3) number of candidate alignments over the number of generations

10.5.5.1 Computing time per generation vs. map density

Figure 10.39 shows the computing time versus map density. It shows that while the computing time generally increases with the map density, the rate of increase is not linear. The computing time is determined for a desktop computer with 450 MHz speed and 96 MB RAM. For much larger applications faster computer may be used.

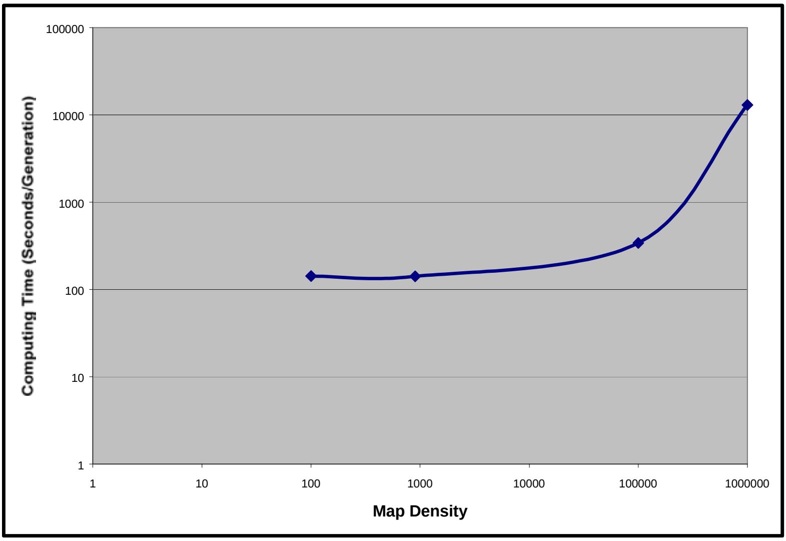


Figure 10.39: Computing time vs. map density.

10.5.5.2 Fluctuation in computing time over successive generations

The computing time over successive generations was tested primarily to see if it changes at later generations. Figure 10.40 shows computing time vs. number of generations for a map with 900 parcels/m^2. No significant changes in the computing time are observed in any part of the search process.

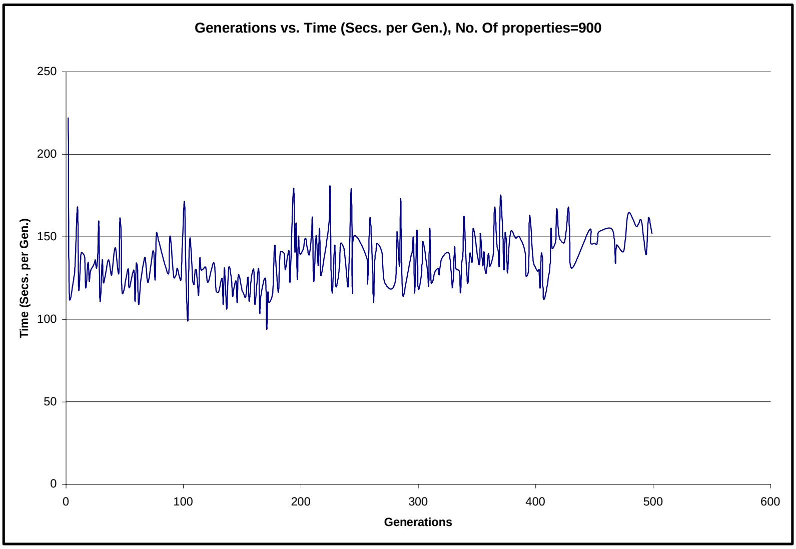


Figure 10.40: Computing time over successive generations for map density=900.

10.5.5.3 Number of candidate alignments at different generations

Here we explore the number of candidate alignments considered at each generation in the search. Figure 10.41 shows the number of candidate alignments over the number of generations for a map with 900 parcels/m^2. It is observed that the number of candidate alignments generally remains steady over successive generations.

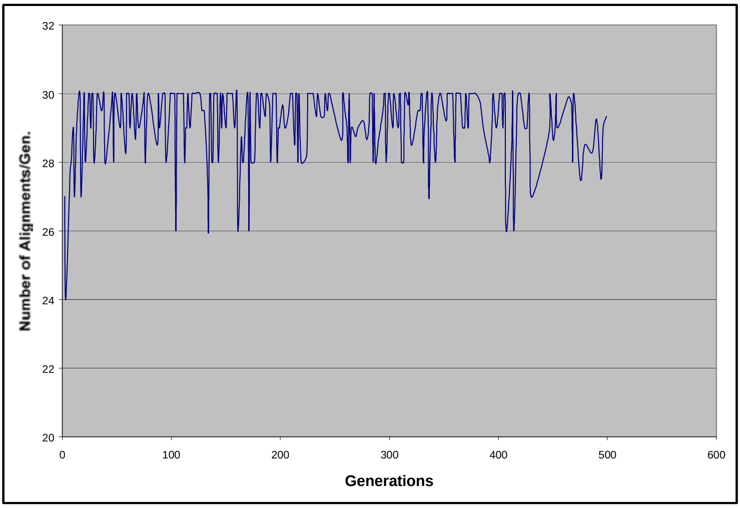


Figure 10.41: Candidate alignments vs. generation number for a map with 900 parcels/m^2.

10.5.6 Significantly different alignments

The first step in selecting significantly different alignments is to perform the optimal search using genetic algorithms until an optimized solution is found. The optimized solution and solutions at intermediate generations (including solutions at intermediate iterations within a generation) are saved for the entire search. The lengths of candidate alignments during the optimal search are also saved. A reference alignment is chosen with respect to which alignments being significantly different are explored. The optimized alignment found using the optimal search is treated as the reference alignment. The following significance functions are computed:

$$SL_i = \frac{|C_i - C_R|}{|L_i - L_R|}, \qquad (10.12)$$

$$SA_i = \frac{|C_i - C_R|}{A_{iR}}, \qquad (10.13)$$

where i = 1,…,n = index of alignments evaluated
SL_i = length-based significance function for the i^{th} alignment
C_i = cost of the i^{th} alignment; C_R = cost of the reference alignment
L = length of the i^{th} alignment
L_R = length of the reference alignment
SA_i = intervening area-based significance function for the i^{th} alignment
A_{ir} = intervening area between the i^{th} alignment and the reference alignment.

In general, lower values of SL_i and SA_i indicate a greater difference between the i^{th} alignment and the reference alignment. However, it is possible to find two alignments of equal length but different intervening areas in which case (Figure 10.42) the alignment having the larger intervening area would indicate greater difference with the reference alignment. Therefore, the alignments are first ranked in the increasing order of *SL* values. The alignments with lower *SL* values are then ranked in the increasing order of *SA* values. The alignments ranking high in this list (i.e., with lower *SA* values) are considered to be significantly different than the reference alignment.

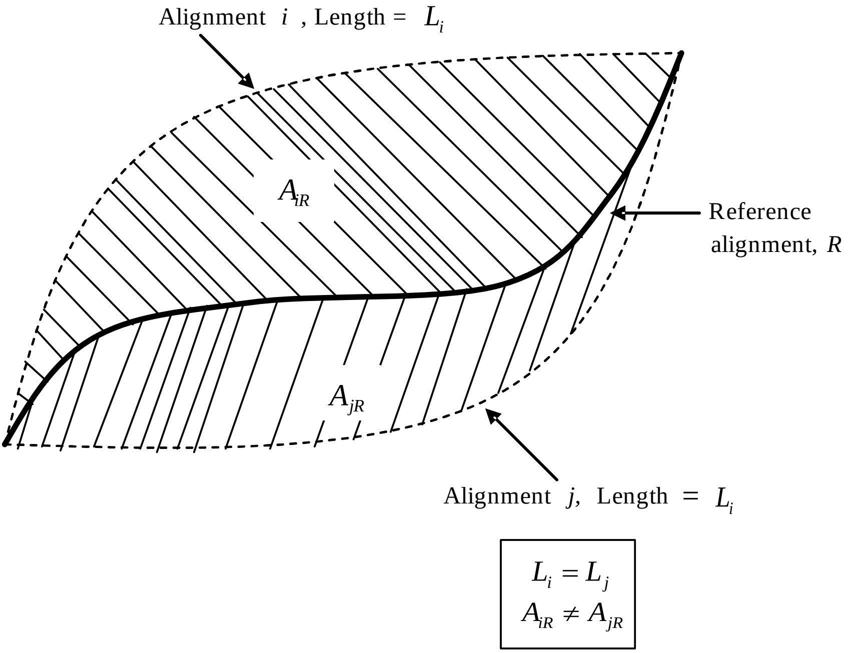


Figure 10.42: Equal length alignments with different intervening areas.

10.5.6.1 An example for identifying significantly different alignments

In order to demonstrate the application of the proposed decision support system a test case using real geographic maps and databases from Maryland is constructed. It is desired to search for a best alignment alternative connecting existing Interstate Rt. 97 and Reece Rd. (Maryland Rt. 174) between specified start and end points from a study section in Anne Arundel County, Maryland. Existing major highways and environmental factors such as floodplains and wetlands falling in the study section are also shown. The shaded portions represent wetlands. The area of the study section is about 8 km^2 and the Euclidean distance between the start and end points is 2.68 km. The terrain height in the study section ranges from 40 to 55 meters.

We choose to perform the optimal search based only on operator cost (i.e., right-of-way, earthwork, and construction costs), primarily to investigate the sensitivity of the proposed alignment to operator cost. Using genetic algorithms the search for an optimized alignment is first performed and the objective function values and lengths for intermediate alignments during the search are saved. The search is carried out for 100 generations during which a total of 3,040 alignments are explored. The optimized alignment obtained at the 100th generation costs $11.005 millions. The length of the optimized alignment is 2.89 km. It crosses floodplain boundaries at locations marked as 1, 2, and 4 in Fig. 10.41, affecting a total floodplain area of 635.16 m^2. It also intersects with Maryland Rt. 170 (marked as 3 in Figure 10.43) requiring an overpass construction. No impact to wetlands is noticed.

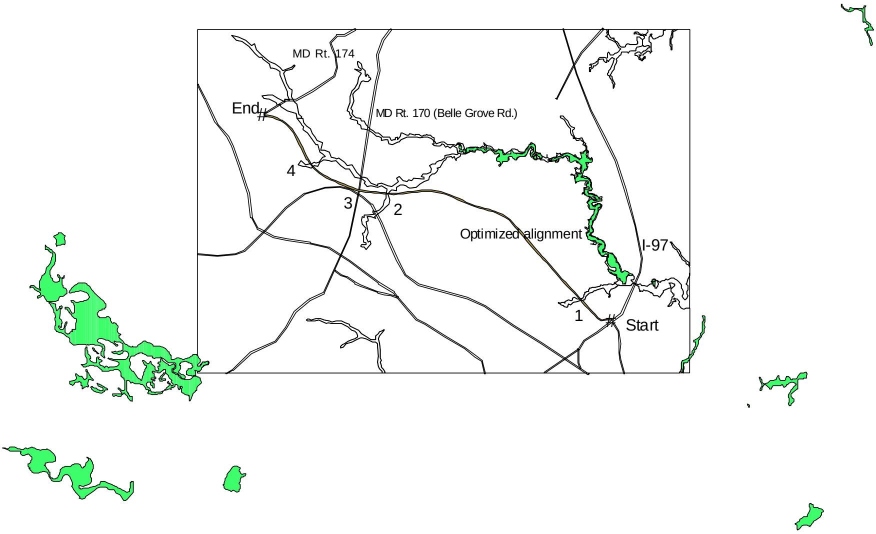


Figure 10.43: Optimized alignment at the 100th generation.

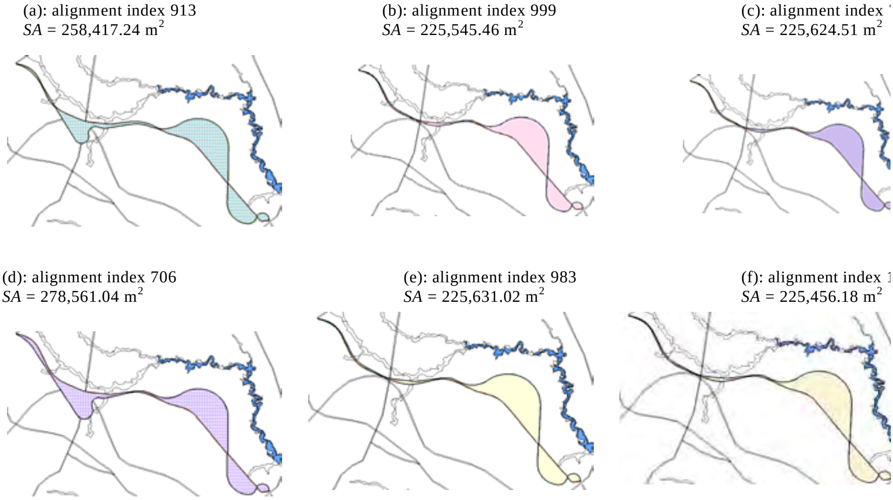
10.5.6.2 Significantly different alignments

In order to explore significantly different alignments the length-based significance function, *SL* is computed for the entire population. The alignments are then ranked in the increasing order of *SL* value. Alignments with lowest 5% *SL* value are then selected for further evaluation since lower *SL* values would indicate greater difference with the reference alignment. Next, identical alignments with nearly equal *SL* values are grouped together to avoid redundancies. This procedure results in 6 alignment groups. A representative alignment from each group is chosen for further analysis. At this stage the intervening area-based significance function, *SA* is computed for the representative alignments. These alignments are then ranked in the increasing order of *SA* value. The configuration of these alignments and their intervening areas are shown in Figure 10.44. An examination of the *SA* values indicate that alignments 999, 771, 983, and 1010 have nearly equal *SA* values and can be grouped together. It is also obvious by looking at Figure 10.44 that the configurations and intervening areas of these alignments are nearly identical. Alignment 999 is selected as the representative alignment of this group. Thus, alignments 706, 913, and 999 are considered to be significantly different than the optimized alignment obtained in the 100th generation during the optical search. It is interesting to note that these alignments (706, 913, and 999) are encountered much earlier (in the 20th, 27th, and 30th generation respectively) in the search.

The final set of alignments with their lengths and total costs are shown in Table 10.12. The corresponding area of floodplain impacts and number of intersections with existing highways are also shown in the table. No impacts to wetlands are noted for either of these alignments. It can be seen that while the optimized alignment has shortest length, minimum cost, and only results in one highway intersection its area of floodplain impact is second lowest. Alignment 999 while 19.4% longer and 22.2% costlier than the optimized alignment further reduces floodplain impact area by 22%. The analysis is quite significant for environmental impact assessment and can be presented to community groups and environmental agencies to work out the tradeoffs among alignment circuity, cost, environmental impact areas, and impacts on existing highways. In Figure 10.45 all four alignments used in the analysis are plotted together, which provides a visual representation of the comparisons among various alternatives. While a simple case with few environmental factors is presented in this analysis the model is able to handle more complex scenarios.

Table 10.12: Final set of alignments.

Alignment Index	Generation Number	Length (m)	Objective Function ($)	Area of Floodplain Impacted (m^2)	Number of Highway Intersections
Optimized alignment	100	2,885.22	11,005,233.92	635.15	1
706	20	3,729.83	14,715,329.37	1,258.51	3
913	27	3,665.93	14,303,501.75	567.03	3
999	30	3,445.63	13,453,250.89	495.79	1


(a): alignment index 913
SA = 258,417.24 m^2

(b): alignment index 999
SA = 225,545.46 m^2

(c): alignment index
SA = 225,624.51 m^2

(d): alignment index 706
SA = 278,561.04 m^2

(e): alignment index 983
SA = 225,631.02 m^2

(f): alignment index 1
SA = 225,456.18 m^2

Figure 10.44: Intervening area of significantly different alignments.

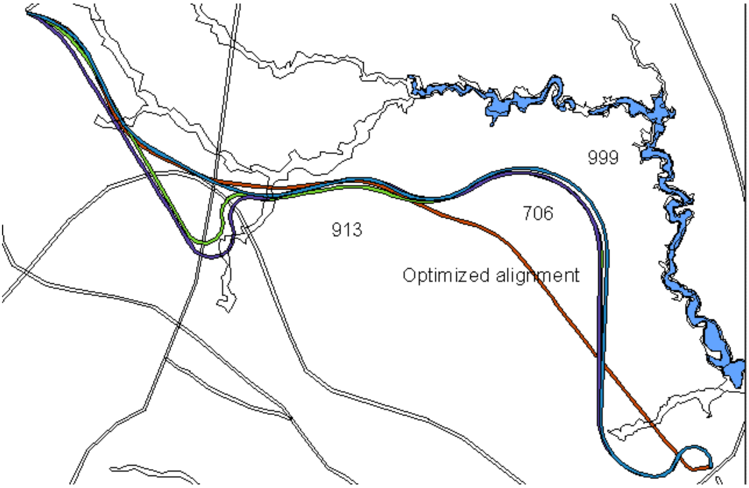


Figure 10.45: Final set of alignments.

This page intentionally left blank

Part C

Intelligent intersection and road structure design

This page intentionally left blank

Chapter 11

Modeling intersections and road structures

Highway alignment selection processes should consider many factors that increase the complexity of this problem. The factors may include structures, topography, socio-economics, ecology, geology, soil types, land use patterns, environment and even community concerns and politics. They are considered with different emphasis and levels of detail at different stages in the alignment selection processes. Traditionally, these processes have consumed much time and efforts of agencies, planners, engineers and residents. Moreover, many highway construction projects have experienced significant changes in project costs and scope due to unpredictable legislated factors (exogenous inputs) during the construction processes (MDOT, 1999). This might be avoided by using optimization algorithms which can minimize the unexpected factors of a plan.

Since an alignment is made subject to a set of design constraints and operational requirements, even slight changes in a particular part of an alignment may eventually influence its whole configuration, thus considerably changing the total costs. Such slight changes may be considered, for example, in minimum curvature, minimum grade and design speed. In addition to these, what if we allow an alignment to incorporate intersections, tunnels, bridges and other structures? And what if we allow the existing roads to be re-optimized for better crossings with the new alignment? We can easily imagine how much these considerations affect the final configuration of an alignment. Unfortunately, structures and re-optimization of existing roads have not previously been incorporated into any highway alignment optimization process.

Recently in the USA, new highways are typically needed to reduce traffic congestion by providing bypass routes or enhancing accessibility between transportation demand generators while minimizing environmental impacts and community concerns. This trend means that a new highway is more likely to pass through a complex environment and require many structures. Indeed, the outputs from alignment optimization models able to handle structures could differ considerably from those without such capabilities.

11.1 Road structures in highway engineering

The scope of this book covers a relatively broad range of structures most likely encountered when optimizing highway alignments. The representative structures on a highway alignment might be intersections, interchanges, bridges and tunnels. Among them, interchanges, bridges and tunnels are the subjects of their own vast research areas. Moreover, in many cases bridges and tunnels dominate a highway planning and construction process in terms of costs and locations. This book is not intended to deal with those completely dominating cases. Therefore, large interchanges, bridges and tunnels are excluded. However, relatively small interchanges and small-scale overpass and underpass structures are considered.

Intersection performance and costs significantly affect highway alignments. According to AASHTO (2001), an intersection is an important part of a highway because, to a great extent, the highway's efficiency, safety, speed, cost of operation and capacity depend on its design. Intersections really influence the configuration of alignments since there is a design constraint for a crossing angle (AASHTO, 2001). Therefore, intersections should be included in highway alignment optimization processes.

Among interchanges, only three small-scale types (e.g., clover, diamond and trumpet types) will be considered. Since an interchange type is normally set in the early stages of alignment optimization, the type will be treated as an exogenous input whenever we should consider interchanges along alignments.

This book focuses on two-lane rural highway alignments. However, the cost functions and alignment optimization models being developed can be extended to other kinds of highways with moderate changes. In summary, the structures covered in this book are as follows:

(1) Intersections
(2) Small bridges
(3) Small tunnels
(4) Small grade separation structures (underpasses and overpasses without ramps)
(5) Small scale interchanges (diamond, clover and trumpet types)

The structures' cost components that are sensitive to alignments will be formulated in detail. These are the superstructure and substructure costs, right-of-way costs, earthwork costs, pavement costs, accident costs, vehicle operating costs, travel time costs, user delay costs and environmental costs. Spatial analysis tools (GIS) and search algorithms (GAs) will be used to incorporate the developed cost functions.

The structures' costs are dominating and sensitive factors. For instance, according to OECD (1973), bridge costs account for about 20% of the total construction costs. Bridge costs are also sensitive. Garber and Hoel (1996) explain principles of bridge location: "The basic principle for locating highway bridges is that the highway location should determine the bridge location, not the reverse. When the bridge is located first, in most cases the resulting highway alignment is not the best. The general procedure for most highways, therefore, is to first determine the best highway location and thus determine the bridge site. In some cases, this will result in skewed bridges, which are more expensive to construct, or in locations where foundation problems exist. When serious problems of this nature occur, all factors such as highway alignments, construction costs of the bridge deck and its foundation, and construction cost of bridge approaches should be considered in order to determine a compromise route alignment that will give a suitable bridge site".

Tunnels are also dominating and sensitive factors in highway alignments in the same way discussed for bridge location. Often, highway tunnels dominate entire alignments, since tunnel construction costs can be very high, as shown in Table 11.1 (Ponnuswamy and Victor, 1996).

Table 11.1: Costs of selected old highway tunnels.

Location and period of construction	**Length (km)**	**Shape**	**Width/height (m)**	**Lining**	**Rock material**	**Cost/m at time of construction (US $)**
Pennsylvania turnpike, 1939-40	10.60	Semi-circle vault	6.90/4.30	RC	Marl, slate, sandstone	1,165
Holland N.Y., 1920-27	5.08	Circle	6.00/3.95	Cast iron	Silt mixed with rock debris	9,500
Mersey, 1925-34	3.18	Circle	19.00/5.70	Cast iron	Fissured sandstone	11,100
Lincoln N.Y., 1934-35	4.68	Circle	6.45/4.00	Cast Iron	Silt mixed with rock debris	10,000
Memorial turnpike, 1954	0.54	Semi-circle vault	7.20/4.30	RC	Sandstone and slate	6,200
Baltimore, 1954-57	2×3.77	Double circle	6.60/4.20	Steel sheet RC lining	Silt, sand and clay	6,650

Ponnuswamy and Victor (1996) state: "A length of road/track through a tunnel may cost up to even 10 times the cost of the road/track on plain land and 4 to 6 times that of the same in a cutting open to sky in hilly areas. Hence tunnels can be justified only by the compensating savings in distance, time and operating cost for the volume of traffic to be handled". A highway alignment will be affected by the small changes in tunnel configurations along it because tunnels have their own special operational, structural, constructional and maintenance requirements.

When designing overpass or underpass structures along a highway, we must provide sufficient vertical and lateral clearances. The need for overpass or underpass structures occurs when a highway meets other roads or railways and there is insufficient reason to provide intersections or interchanges. Providing sufficient vertical and lateral clearances means other parts of vertical profile may be significantly changed since additional design constraints are imposed.

Interchange construction costs are also significant enough to influence entire highway projects. First of all, construction costs can reach hundreds of millions of dollars (Holzmann and Marek, 1993). AASHTO (2001) states: "There are several basic interchange forms or geometric patterns of ramps for turning movements at a grade separation. Their application at a particular site is determined by the number of intersection legs, the expected volumes of through and turning movements, topography, culture, design controls, proper signing, and the designer's initiative". This explains why there are several types of interchanges and several design constraints to satisfy for each type. Indeed, the selection of an interchange type determines how the connected highways should be designed.

11.2 Importance of incorporating road structures into highway alignment optimization

One illustrative example is presented to show how structures possibly affect highway alignments. Previously, this book developed highway alignment optimization models using genetic algorithms without considering structures. The developed algorithms worked on artificial study areas divided into small rectangular cells. By assuming the lake in the middle part of the study area to be untouchable, the developed model found the optimized alignment shown in Figure 9.34, which is repeated in Figure 11.1.

Suppose we can consider a bridge over the lake by relaxing the untouchable limitation (in most real application cases, this relaxation is better than assuming the lake untouchable). Then a better solution is imaginable, as shown in Figure 11.2.

Again suppose a search model is developed to consider tunnels and bridges together. Then a solution for this case may be less circuitous than in Figure 11.1. Providing tunnel structures may lead to a better solution, as shown in Figure 11.3.

Meanwhile, if other roads exist within the study area and connections with them are needed, the final solution may be significantly changed from the original solution, as shown in Figure 11.4.

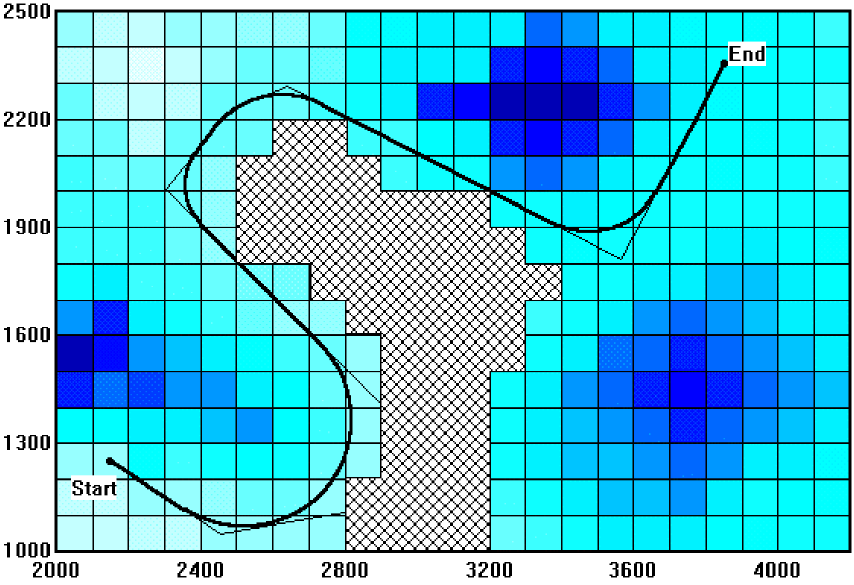


Figure 11.1: Best alignment based on an artificial study area by the developed algorithm.

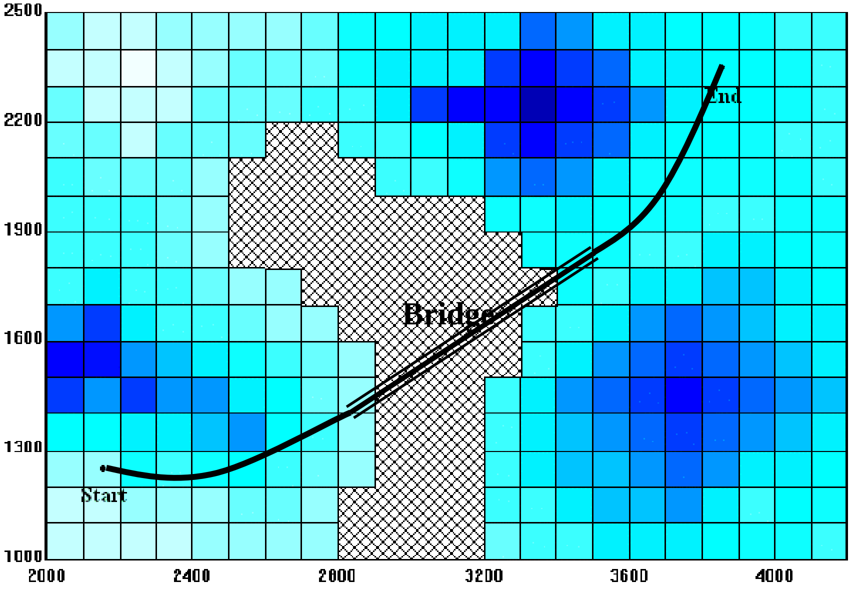


Figure 11.2: Possible solution incorporating bridges into optimization.

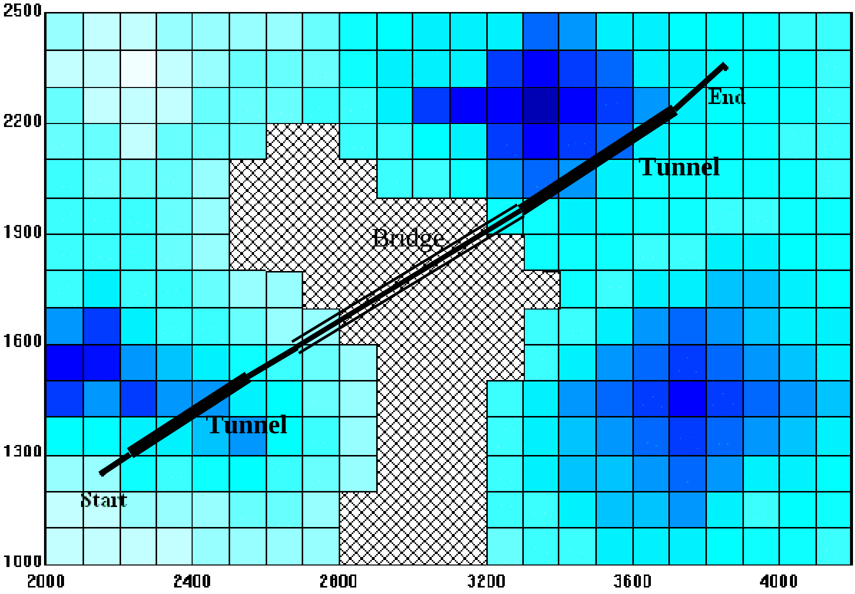


Figure 11.3: Another possible solution incorporating bridges and tunnels.

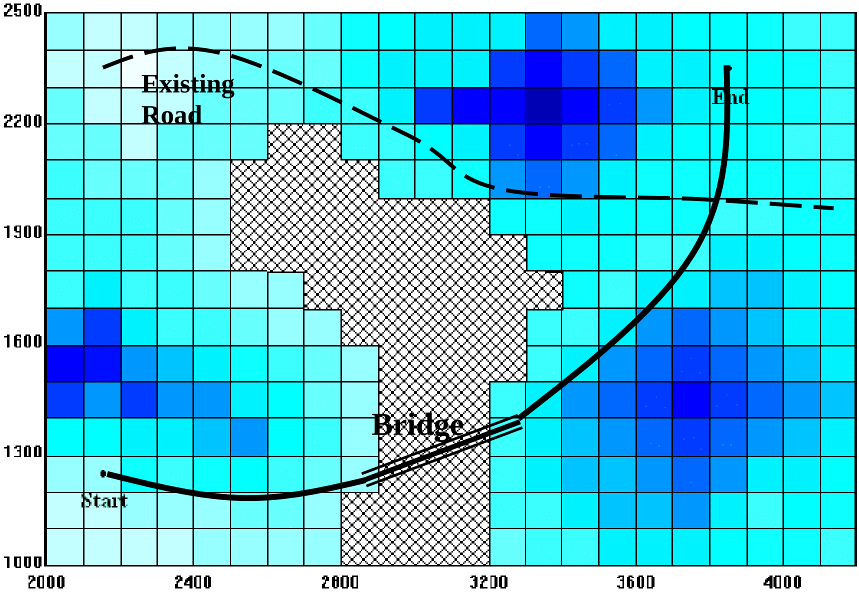


Figure 11.4: Another possible solution with an existing road.

11.3 Characteristics of road structures on highways

In order to efficiently incorporate the characteristics of structures into highway alignment optimization processes, this section discusses important design constraints and operational requirements of structures. For each structure type, there are many factors to consider. Some of them affect alignment configurations while the others affect smaller design details. Among them, only factors sensitive to alignments at highway planning levels are reviewed. The characteristics of each structure discussed in the following sections provide the fundamental concepts for formulating each structure's cost function in chapter 12.

11.3.1 At-grade intersection characteristics

AASHTO (2001) introduces four basic elements that enter into design considerations of at-grade intersections. They are human factors, traffic considerations, physical elements and economic factors, as listed in Table 11.2.

Among the factors in Table 11.2, those mainly affecting alignments are vehicle speeds (design speed), vertical alignments at the intersection (differences between an existing road and a new road), angle of the intersection, geometric features (topography of the site and cross sections), design hourly turning movements (additional lane need for turning volumes) and sight distances.

The design speed is one of the most important factors influencing alignment configurations. Especially in curved sections, the design speed determines the lengths of radii and the associated superelevation. Vertical alignments at intersections very significantly affect earthwork cost estimates. Greater elevation differences between an existing road and a new road imply increasingly higher earthwork costs. The intersection angle is also an important factor. Too acute or oblique crossings should be avoided. Walker (1993) described a number of problems arising as the angle deviates from 90 degrees: (1) the area of conflict increases, (2) visibility is limited, (3) larger turning roadway areas are required, (4) the exposure time through the intersection is increased and (5) the potential of accidents increases.

Intersection geometric features, including cross sections and topography of the site, are also affecting earthwork costs, drainage costs and pavement costs. Design-hour turning movements are the factors determining whether additional lanes are needed for smoothness and safety. They also affect accident frequency. Proper sight distances are important for avoiding potential vehicle conflicts at intersections. Sight distances can be used for finding what obstructions around the intersection should be removed. Hence, sight distance information can be employed to estimate right-of-way costs of the intersection.

Many factors are left out of Table 11.2. They should be considered at the detailed intersection design stage. Control types of intersections and land acquisition related factors (that is, right-of-way cost factors) are further discussed and formulated in chapter 12 since they are sensitive to alignments.

Table 11.2: Intersection design considerations.

Main items	Subcomponents
Human factors	1. Driving habits 2. Ability to make decisions 3. Driver expectancy 4. Decision and reaction time 5. Conformance to natural paths of movements 6. Pedestrian use and habits 7. Bicycle traffic use and habits
Traffic considerations	1. Design and actual capacities 2. Design-hour turning movements 3. Size and operating characteristics of vehicle 4. Variety of movements (diverging, merging, weaving, and crossing) 5. Vehicle speeds 6. Transit involvement 7. Accident experience 8. Bicycle movements
Physical elements	1. Character and use of abutting property 2. Vertical alignments at the intersection 3. Sight distance 4. Angle of the intersection 5. Conflict area 6. Speed-change lanes 7. Geometric features 8. Traffic control devices 9. Lighting equipment 10. Safety features 11. Bicycle traffic 12. Environmental factors
Economic factors	1. Cost of improvements 2. Effects of controlling or limiting right-of-way on abutting residential or commercial properties where channelization restricts or prohibits vehicular movements 3. Energy consumption

11.3.2 Small bridge characteristics

When bridges are dominating the associated alignments, the following arguments by Barker and Puckett (1997) are generally true: “a bridge is the key element in a transportation system for three reasons, (1) it controls the capacity of the system, (2) it is the highest cost per mile of the system and (3) if the bridge fails, the system fails”. Even for the much smaller bridges considered in this book, the last argument is partially true. In bridge engineering, there are several forms of bridges (Barker and Puckett, 1997) such as stone arch bridges, wooden bridges, metal truss bridges, suspension bridges, metal arch bridges, reinforced concrete bridges and girder bridges. Among them, steel and concrete composite girder bridges account for most highway bridges in the USA. This book focuses on the steel and concrete composite girder bridge type.

To extract highway bridge characteristics affecting alignments, we need to take earthwork volumes into account. Earthwork costs, especially fill volumes, are directly associated with constructing bridges. There should be an economical break-even point between fills and building bridges depending on various site-specific characteristics. In this book, the break-even point is determined and used to control alignment alternatives. Of course, bridges are definitely selected when they are the only possible options, e.g., for crossing creeks while considering high water level constraint (ε_w) in case of flooding.

The other important bridge characteristics affecting highway alignments include radii of bridges, span lengths, number of spans, number of piers and heights of piers. Since this book is not limited to straight bridges, those having horizontal curvatures are considered within alignments. This could significantly increase bridge costs. However, the total alignment costs may be smaller. Bridge costs can be separated into superstructure costs and substructure costs (Xanthakos, 1994; O'Connor, 1971). Span lengths, number of spans, numbers of piers and heights of piers are very crucial for estimating bridge costs. The bridge cost formulation is discussed in detail in chapter 12.

11.3.3 Characteristics of grade separated structures (overpass and underpass)

New highway alignments may cross many existing roads, using either underpasses or overpasses, depending on the alignment profile. These grade separation structures have similar characteristics to bridges discussed earlier in section 11.3.2.

There are two additional factors affecting alignments: vertical and lateral clearances. In principle, the minimum lateral clearance from the edge of the traveled way to the face of the protective barrier should be the normal shoulder width (AASHTO, 2001). The required vertical clearance should also be provided. Although vertical clearances of 4.1 m to 4.4 m have been adopted by several U.S. states, additional clearance is desirable to compensate for resurfacings, snow, ice accumulation and an occasional slightly overheight load. The recommended minimum clearance is 4.4 m, and the desirable clearance is 5.0 m (AASHTO, 2001).

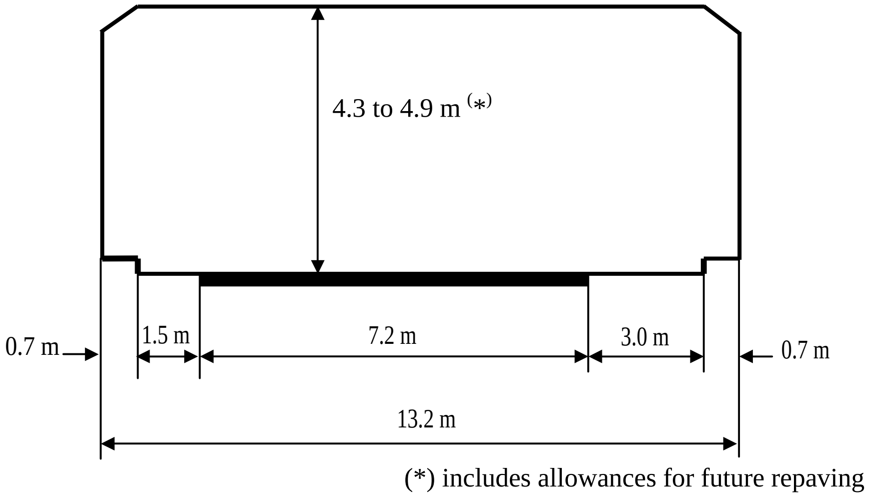
11.3.4 Small tunnel characteristics

A tunnel is well described by King and Kuesel (1996): "A tunnel serves any of myriad functions - highway, railroad, or rapid transit artery; pedestrian passageway; fresh water conveyance, cooling water supply, wastewater collector or transport; hydropower generator; or utility corridor… Its length can vary from less than 100 ft to more than 30 miles. A tunnel can be located in any of a variety of places – under mountains, cities, rivers, lakes, sea estuaries, straits, or bays. Finally, a tunnel is constructed in one of innumerable media – soft ground, mixed face, rock, uniform, jumbled, layered, dry, wet, stable, flowing and squeezing".

Among many tunnel elements, the most important factors for highway tunnels are ventilation for pollutants and consequent adjustment of the air supply and exhaust, lighting for safety and ensuring maximum appropriate speeds, fire life safety provisions for providing refuge from a raging fire or deadly smoke, elaborate traffic surveillance and control systems coordinated with the other system for protected egress of motorists in the event of a fire and access for fire-fighting personnel, and soil types for earthwork and construction process (King and Kuesel, 1996). These elements are functions of several characteristics of a tunnel. Many tunnel characteristics affect their costs. Among them, those characteristics affecting highway alignments include cross sections, clearances, horizontal alignments and grades.

Figure 11.5 illustrates a typical AASHTO desirable cross section and clearances for a two-lane tunnel with two 3.6 m lanes, a 3.0 m right shoulder, a 1.5 m left shoulder, and a 0.7 m curb or sidewalk on each side. The roadway width may be distributed to either side in a different way if necessary to better fit the dimensions of the tunnel approaches. The vertical clearance for the desirable section is 4.9 m for freeways and 4.3 m for other highways.

If possible, the tunnel alignment should be straight. If curves are required, the minimum radius is determined by stopping sight distances and acceptable superelevation in relation to design speed. Where shoulders are narrow, horizontal sight distance may be restricted by the proximity of the tunnel sidewall. Usually, passing distances do not apply.



(*) includes allowances for future repaving

Figure 11.5: AASHTO typical desirable cross section and clearances for a two-lane tunnel.

Upgrades in tunnels carrying heavy traffic are preferably limited to 3.5% in order to reduce ventilation requirements. For long two-lane tunnels with two-way traffic, a maximum grade of 3% is desirable to maintain reasonable truck speed. For downgrade traffic, 4% or more is acceptable. For lighter traffic volumes, grades up to 5% or even 6% have been used for economy's sake (King and Kuesel, 1996).

11.3.5 Characteristics of interchanges

Interchanges provide exclusive movement of traffic among two or more roadways and are systems of interconnected roadways using grade separation. General design considerations for interchanges are type determination, including the number of structures involved, horizontal and vertical alignments, cross sections and sight distances.

An interchange could be either a system interchange or a service interchange, depending on its role. Generally, system interchanges provide for freeway-to-freeway connections, whereas service interchanges connect one level of the highway system to a lower service level facility (Holzmann and Marek, 1993; AASHTO, 2001). Interchanges can be categorized into several basic types including three-leg interchanges (trumpet, directional T or Y), diamond, cloverleaf, partial cloverleaf, directional with loops and all directional (Leisch, 1993). AASHTO (2001) states that the final configuration of an interchange may be determined by the need for route continuity, uniformity of exit patterns, single exit in advance of the separation structure, elimination of weaving on the main facility, signing potential, and availability for right-of-way.

The general controls for horizontal and vertical alignment and their combination should be adhered to closely. In particular, any relatively sharp horizontal and vertical curves should be avoided (AASHTO, 2001). The alignments and cross sections of the approaches to a grade separation without ramps involve no special problems except where a change in width is made to include a middle pier or where the median is narrowed for structure economy. With ramps, changes in alignment and cross section may be required to ensure proper operation and to develop the necessary capacity at the ramp terminals, particularly where there is not a full complement of ramps and where some left turns at grade are necessary (AASHTO, 2001). AASHTO also recommends that the sight distance on the highways through an interchange should be at least as long as that required for stopping, and preferably longer. Generally, design considerations for interchanges address the same requirements as basic highway segments, even if there are some exceptional cases. Therefore, except for some extreme cases such as ramps, all components of interchanges are treated as one part of basic highway segments when developing cost functions.

This page intentionally left blank

Chapter 12

Cost functions of intersections and road structures for highway alignment optimization

This chapter develops cost functions for structures based on their characteristics, which affect highway alignments, as discussed in chapter 11. The cost functions developed here may not suitable for final detailed design. However, they are sufficiently sensitive to allow structures to be incorporated into highway alignment optimization. This chapter starts with improvements to earthwork cost estimation methods embedded in previously developed alignment optimization models. Cost functions for each structure follow.

12.1 Estimating highway earthwork cross sectional areas

Earthwork volume estimation is one of the most important components in estimating highway construction costs. According to studies by OECD (1973) and Chew at al (1989), earthwork costs reach up to about 25% of all construction costs. Therefore, a good method for correctly estimating earthwork volume is essential, although precision requirements for earthwork volume estimation are lower in planning applications than in detail design of highways. Thorough surveying is needed for detailed design but agencies usually cannot afford it for all alternatives considered at the planning stage and try, if possible, to use existing data sets.

Among a number of methods available, the Average End Area Method and the Prismoidal Method have been most frequently employed to estimate earthwork volumes (Jha, 2000). The precision of both methods depends on the precision of cross section area estimates. Since detailed surveying results are usually unavailable in preliminary planning, we must usually rely on existing data sets with lower resolution. Although technologies for finding and storing positions (or locations) into data files, such as global positioning systems (GPS) and geographic information systems (GIS) are advancing in promising ways

(ESRI, 1999), they still provide insufficient resolutions in many planning applications.

12.1.1 Current methods for estimating cross sectional areas

The Average End Area Method is the most commonly employed method for estimating earthwork volume (Wright, 1996; Garber and Hoel, 1996). It assumes that the earthwork volume between two consecutive cross sections is the average of their areas multiplied by the distance between them:

$$V = L(\frac{A_1 + A_2}{2}) , \qquad (12.1)$$

where V = volume (m^3)
A_1 and A_2 = end areas (m^2)
L = distance (m).

The Prismoidal Method applies Simpson's rule, and is formulated as follows:

$$V = \frac{L}{6}(A_1 + 4A_m + A_2) , \qquad (12.2)$$

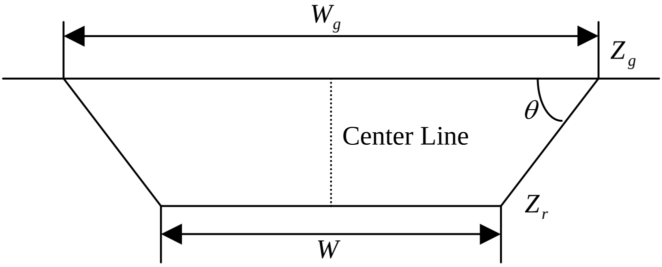
where V = volume (m^3)
A_1 and A_2 = end areas (m^2)
A_m = middle area (m^2).

Eqns (12.1) and (12.2) indicate that the accuracy of V really depends on the accuracy of each end area. They also indicate that the approximation of V improves as L decreases. Figure 12.1 shows a typical cut cross section whose area (A_{c1}, m^2) can be computed using eqn (12.3).

$$A_{c1} = \frac{1}{2}(W + W_g)(z_g - z_r) . \qquad (12.3)$$

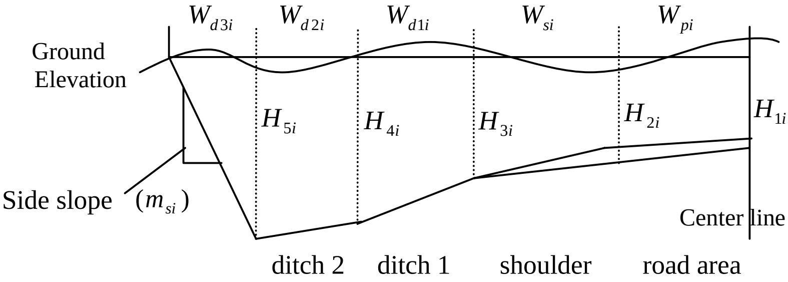
In applying eqn (12.3), two questions arise: "How can we find the ground elevation of center line?" and "What if the terrain is not flat?" The first question implies a need for accurate databases while the second requires a reasonable method for estimating cross section areas.

Moavenzadeh *et al* (1973) developed one method to handle irregular terrain. They introduced an analytical way to compute cross section area (A_{c2}, m^2) by dividing it into several pieces based on the design standard. Figure 12.2 and eqn (12.4) show the basic concept, based on a typical cut section. However, the method provides no way for finding the coordinates of many referencing points such as the edges of the traveled portion and shoulders. Without the coordinates of these points, corresponding ground elevations cannot be found and the first question above is left unanswered. This weakness prevents full automation of earthwork cost estimates in highway optimization processes.



W_g = ground width of cutting (m)
W = road width (m)
z_g = ground elevation of center line (m)
z_r = road elevation of center line (m)
θ = side slope (degree)

Figure 12.1: Typical cut cross section.



$H_{\bullet i}$ = height at each referencing point (m)
$W_{\bullet i}$ = width of each segments (m)
m_{si} = side slope (decimal)

Figure 12.2: More detailed cut cross section.

$$A_{c2} = 2\left[\begin{array}{l}(\frac{H_{1i}+H_{2i}}{2}\times W_{Pi})+(\frac{H_{2i}+H_{3i}}{2}\times W_{Si})+(\frac{H_{3i}+H_{4i}}{2}\times W_{d1i})+\\(\frac{H_{4i}+H_{5i}}{2}\times W_{d2i})-(\frac{H_{5i}{}^{2}}{2m_{3i}})\end{array}\right] \qquad (12.4)$$

In an automated optimization process, the ground elevation of each referencing point of a road cross section should also be found automatically. Then we must develop a method to estimate ground elevation precisely enough, and construct a good data set with sufficiently fine resolution for our study area. At the highway planning level, GIS database tools are relatively effective for constructing databases. With GIS, one can store each parcel's attributes (information, including elevation) through a so called "theme table". In many applications, GIS databases are constructed based only on the ownership of properties, with only one elevation value in each parcel. Unless a parcel is subdivided into several pieces (which is unlikely, based on ownership), calculations within the processes introduced by Jha (2000) or Jong (1998) may be insufficiently precise. Therefore, it is worthwhile to develop a good method for ensuring sufficiently accurate estimation of earthwork.

12.1.2 Methodology for estimating cross section areas

When obtaining an alignment alternative through an optimization process, the expected output is a centerline having three dimensional coordinates. From that, it is easy to find such cross section referencing points as the edge of traveled way, the edge of shoulder, the edge of ditch, and the side slope, based on the design standards of the road being built. To describe center lines, parametric representation is useful (Swokowski, 1979; Mortenson, 1997; Lovell, 1999) and an alignment can be defined as in section 3.1.

Based on the definition and vector representation of a space curve, it is simple to describe highway alignments using vector notation. Suppose the i^{th} station has 7 referencing points being considered (as in Figure 12.2). Then the center line representation produced by optimization techniques will be

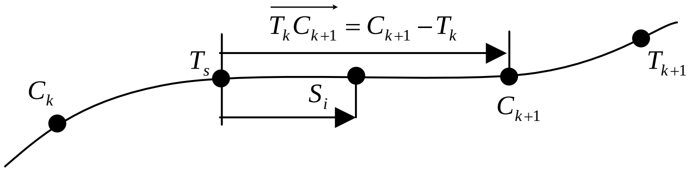
$$\mathbf{S}_i = \mathbf{X}_{Si}^C = [x_{Si}^C, y_{Si}^C, z_{Si}^C]^T , \tag{12.5}$$

where $\mathbf{X}_{Si}^C$ = vector notation of three dimensional (3D) coordinates of i^{th} station's center point.

To obtain the *x* and *y* road coordinates for other referencing points in the travel direction, those should be investigated separately, since $\mathbf{S}_i$ could be located either on a tangent section or on a circular curve. (Spiral transition curves may be neglected at the planning stage.) This can be done on a two dimensional (2D) basis.

12.1.2.1 Tangent section

Figure 12.3 illustrates a typical tangent section being investigated.



$C_k =$ k^{th} circular point

$T_k =$ k^{th} tangent point

$S_i =$ i^{th} station point

Figure 12.3: Typical tangent section.

To find the coordinates of referencing points, “The Gram-Schmidt Process” (Lay, 1997) is useful, which is a simple and quick method for generating an orthogonal or orthonormal basis. The following procedure specifies how the orthogonal basis and the needed coordinates are determined:

(1) Let $\mathbf{V}_1 = \overrightarrow{T_kC_{k+1}}$

(2) Find projection $\mathbf{P}$ of $\mathbf{S}_i$ onto $\mathbf{V}_1$; then, the orthogonal component of $\mathbf{S}_i$ to $\mathbf{V}_1$ is $\mathbf{S}_i$ - $\mathbf{P}$

(3) Then, $\mathbf{V}_2 = \mathbf{S}_i - \mathbf{P} = \mathbf{S}_i - \left(\dfrac{\mathbf{S}_i \cdot \overrightarrow{T_kC_{k+1}}}{\overrightarrow{T_kC_{k+1}} \cdot \overrightarrow{T_kC_{k+1}}} \right) \overrightarrow{T_kC_{k+1}}$, where “·” is dot product.

(4) Therefore, the coordinates of the edge of traveled way ($\mathbf{X}_{Si}^{et}$), edge of shoulder ($\mathbf{X}_{Si}^{es}$) and edge of ditch ($\mathbf{X}_{Si}^{ed}$) are

$$\mathbf{X}_{Si}^{et} = \begin{bmatrix} x_{Si}^{et} \\ y_{Si}^{et} \\ z_{Si}^{et} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_i + (-\dfrac{w}{2})\dfrac{\mathbf{V}_2}{\|\mathbf{V}_2\|} \\ \\ z_{Si}^{C} + ts(-\dfrac{w}{2}) \end{bmatrix}, \qquad (12.6)$$

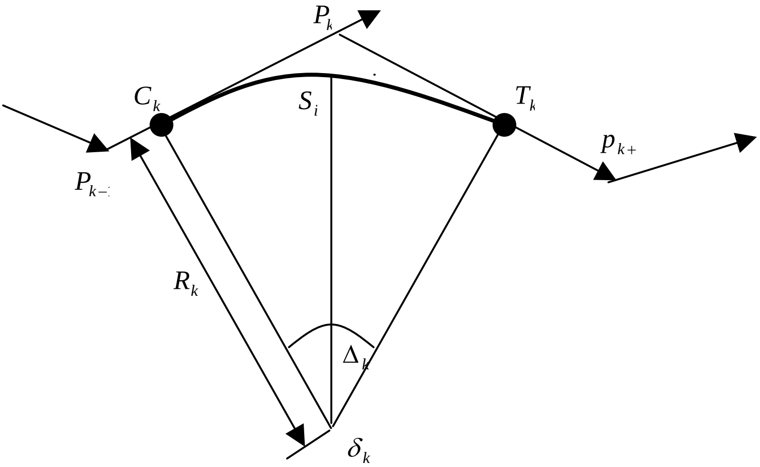
$$\mathbf{X}_{Si}^{es} = \begin{bmatrix} x_{Si}^{es} \\ y_{Si}^{es} \\ z_{Si}^{es} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_i + (-\dfrac{w}{2} - ws)\dfrac{\mathbf{V}_2}{\|\mathbf{V}_2\|} \\ \\ z_{Si}^{et} + ss(-ws) \end{bmatrix}, \qquad (12.7)$$

$$\mathbf{X}_{Si}^{ed} = \begin{bmatrix} x_{Si}^{ed} \\ y_{Si}^{ed} \\ z_{Si}^{ed} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_i + (-\dfrac{w}{2} - ws - wd)\dfrac{\mathbf{V}_2}{\|\mathbf{V}_2\|} \\ \\ z_{Si}^{es} + sd(-wd) \end{bmatrix}, \qquad (12.8)$$

where w = width of traveled way
ts = slope of traveled way
ss = slope of shoulder
sd = slope of ditch
ws = width of shoulder
wd = width of ditch.

12.1.2.2 Circular Curve

Figure 12.4 shows one typical circular section on an alignment.



where δ_k = center of k^{th} circular point
P_k = k^{th} point of intersection
R_k = radius of k^{th} circular curve
Δ_k = angle of k^{th} circular curve

Figure 12.4: Typical circular curve section.

Then, the coordinates of the referencing points can be determined using vector manipulation, as shown below:

$$\mathbf{X}^{et}_{S_i} = \begin{bmatrix} x^{et}_{S_i} \\ y^{et}_{S_i} \\ z^{et}_{S_i} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_i + (-\frac{w}{2})\dfrac{\overrightarrow{\delta_k S_i}}{\left\| \overrightarrow{\delta_k S_i} \right\|} \\ \\ z^{C}_{S_i} + e(-\frac{w}{2}) \end{bmatrix}, \qquad (12.9)$$

$$\mathbf{X}_{S_i}^{es}=\begin{bmatrix}x_{S_i}^{es}\\ y_{S_i}^{es}\\ z_{S_i}^{es}\end{bmatrix}=\begin{bmatrix}\mathbf{S}_i+(-\frac{w}{2}-ws)\frac{\overrightarrow{\delta_k S_i}}{\left\|\overrightarrow{\delta_k S_i}\right\|}\\ \\ z_{S_i}^{et}+e(-ws)\end{bmatrix}, \tag{12.10}$$

$$\mathbf{X}_{S_i}^{ed}=\begin{bmatrix}x_{S_i}^{ed}\\ y_{S_i}^{ed}\\ z_{S_i}^{ed}\end{bmatrix}=\begin{bmatrix}\mathbf{S}_i+(-\frac{w}{2}-ws-wd)\frac{\overrightarrow{\delta_k S_i}}{\left\|\overrightarrow{\delta_k S_i}\right\|}\\ \\ z_{S_i}^{es}+e(-wd)\end{bmatrix}, \tag{12.11}$$

where e = superelevation.

12.1.3 Developing new methods for finding ground elevations

By now, all *x* and *y* coordinates of the referencing points are obtained. The remaining task is to find the ground elevations (*z* coordinates) of those points.

At the highway planning level, GIS databases might be used to obtain those. As discussed earlier GIS databases may only supply each parcel's elevation, which provides insufficient resolution.

To overcome this problem, we introduce in this work "Proportionally Weighted Interpolation" and "Planar Interpolation", depending on the cases considered below in Figure 12.5. These two approximation methods are inspired by the Limit Theorem, one of the most important theorems in calculus, explained by Swokowski (1979) as follows:

"In calculus and its applications we are often interested in the values of *f*(*x*) of a function *f* when *x* is very close to a number *a*, but not necessarily equal to *a*. As a matter of fact, in many instances the number *a* is not in the domain of *f*; that is, *f*(*a*) is undefined. Roughly speaking, we ask the following question: As *x* gets closer and closer to *a* (but $x \neq a$), does *f*(*x*) get closer and closer to some number *L*? If the answer is yes, we say that the limit of *f*(*x*), as *x* approaches *a*, equals *L*, and we write $\lim_{x\to a} f(x) = L$ "

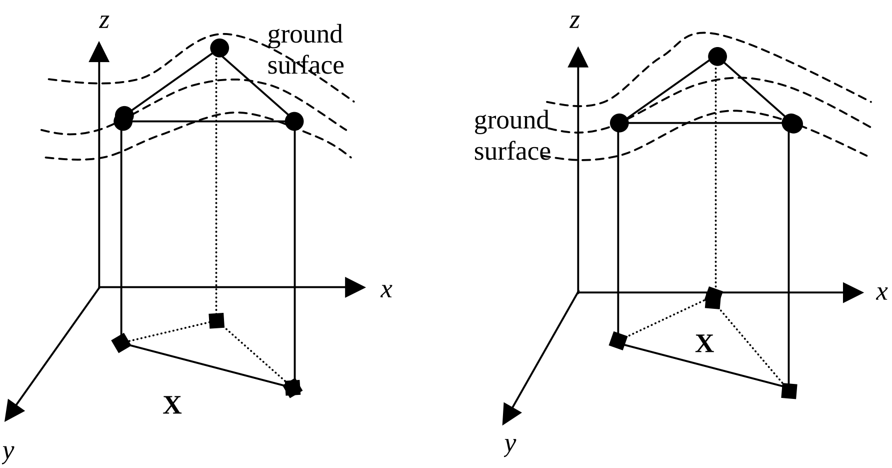
For an application to the problem, this book focuses on the case where the ground elevation of center line of some particular station point $\mathbf{S}_i$ must be obtained. Again its coordinate is $\mathbf{X}_{S_i}^{C}=[x_{S_i}^{C}, y_{S_i}^{C}, z_{S_i}^{C}]^T$. Suppose that the ground elevations of a few points above the station point can be obtained from surveying or GIS databases. Two possible cases shown in Figure 12.5 can be imagined.

In case 1, the road elevation is not covered by the orthogonal projection of a triangular plane made by three closest surveying points in terms of 2D distances. In case 2, the road elevation is covered. For the case 1, Proportionally Weighted Interpolation will be developed while case 2 employs Planar Interpolation.

12.1.3.1 Case 1 – Proportionally weighted interpolation

First, 3 points based on the 2D closest criterion should be found. The reason for not using 3D distances is that 3D distances are more likely to mislead than those of 2D. The 2D distance between $\mathbf{X}_{Si}^C$ and any surveying point $\mathbf{S}_n^s = [x_n^s, y_n^s, z_n^s]^T$ is

$$\left\| \mathbf{X}_{Si}^C - \mathbf{S}_n^s \right\|_{2D} = \sqrt{(x_{Si}^C - x_n^s)^2 + (y_{Si}^C - y_n^s)^2} \ . \qquad (12.12)$$



Case 1

Case 2

X = road elevation points
● = surveying points
■ = projected points

Figure 12.5: Two possible cases when projecting ground elevations onto *xy* plane.

Suppose that three points $\mathbf{S}_1^s$, $\mathbf{S}_2^s$, and $\mathbf{S}_3^s$ which are closest to $\mathbf{X}_{Si}^C$ in order, are obtained and let $A = \left\| \mathbf{X}_{Si}^C - \mathbf{S}_1^s \right\|_{2D}$, $B = \left\| \mathbf{X}_{Si}^C - \mathbf{S}_2^s \right\|_{2D}$, and $C = \left\| \mathbf{X}_{Si}^C - \mathbf{S}_3^s \right\|_{2D}$, respectively, where, $A \le B \le C$. Also, suppose that three vectors using three points, say $\mathbf{S}_2^s - \mathbf{S}_1^s$, $\mathbf{S}_3^s - \mathbf{S}_1^s$ and $\mathbf{S}_3^s - \mathbf{S}_2^s$ are constructed. Then, the *z* coordinate changes z_1^s up to z_2^s along with $\mathbf{S}_2^s - \mathbf{S}_1^s$. If $\mathbf{X}_{Si}^C$'s ground elevation is assumed proportional to those distances of $A = \left\| \mathbf{X}_{Si}^C - \mathbf{S}_1^s \right\|_{2D}$, and $B = \left\| \mathbf{X}_{Si}^C - \mathbf{S}_2^s \right\|_{2D}$, the interpolated value, $[z_1^s + (z_2^s - z_1^s)A/B]$ (*alternative 1*), may be considered as one alternative approximation of ground elevation of $\mathbf{X}_{Si}^{C}$. Likewise, two more alternatives using z_3^s and z_1^s, and z_3^s and z_2^s can be obtained, and they will be $[z_1^s + (z_3^s - z_1^s)A/C]$ (*alternative 2*), and $[z_1^s + (z_3^s - z_2^s)B/C]$ (*alternative 3*), respectively.

For an even better approximation value of the ground elevation of $\mathbf{X}_{Si}^{C}$, the proportionally weighted interpolation method is preferable:

$$\mathbf{X}_{Si}^{C} = \left[z_1^s + (z_2^s - z_1^s)\frac{A}{B} \right] \times a + \left[z_1^s + (z_3^s - z_1^s)\frac{A}{C} \right] \times b + \left[z_2^s + (z_3^s - z_2^s)\frac{B}{C} \right] \times c, \tag{12.13}$$

where $0 < a < 1$, $0 < b < 1$, $0 < c < 1$, $a > b > c$, and $a+b+c = 1$.

12.1.3.2 Case 2 – Planar interpolation

For case 2, where the road elevation is within the orthogonal projection of a triangular plane made by three closest points, the planar interpolation is employed. To apply this method, an equation representing the plane (a plane equation is $ax+by+cz+d=0$) should be developed. Then, it is easy to obtain the z coordinate of a road point by substitution.

To make this possible, this research introduces a *vector product* that can produce a *normal vector*. Since two vectors ($\mathbf{S}_2^s - \mathbf{S}_1^s$, and $\mathbf{S}_3^s - \mathbf{S}_1^s$) are easily obtained, we can produce another vector, which is normal and also orthogonal to both $\mathbf{S}_2^s - \mathbf{S}_1^s$ and $\mathbf{S}_3^s - \mathbf{S}_1^s$, can be produced using the vector product. After rewriting them (i.e., $\mathbf{S}_2^s - \mathbf{S}_1^s$ and $\mathbf{S}_3^s - \mathbf{S}_1^s$) into $[x_2^s - x_1^s, y_2^s - y_1^s, z_2^s - z_1^s]^T$ and $[x_3^s - x_1^s, y_3^s - y_1^s, z_3^s - z_1^s]^T$, respectively, their vector product is

$$(\mathbf{S}_2^s - \mathbf{S}_1^s) \times (\mathbf{S}_3^s - \mathbf{S}_1^s) = \left[\begin{vmatrix} y_2^s - y_1^s & z_2^s - z_1^s \\ y_3^s - y_1^s & z_3^s - z_1^s \end{vmatrix}, \begin{vmatrix} x_2^s - x_1^s & z_2^s - z_1^s \\ x_3^s - x_1^s & z_3^s - z_1^s \end{vmatrix}, \begin{vmatrix} x_2^s - x_1^s & y_2^s - y_1^s \\ x_3^s - x_1^s & y_3^s - y_1^s \end{vmatrix} \right]^T, \tag{12.14}$$

where $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad\text{-}bc$, determinant of order 2

$\times$ = symbol for vector product.

Now, since one point, $\mathbf{S}_1^s = [x_1^s, y_1^s, z_1^s]$, and the normal vector, $(\mathbf{S}_2^s - \mathbf{S}_1^s) \times (\mathbf{S}_3^s - \mathbf{S}_1^s)$, are known, the needed plane equation can be constructed as follows (Swokowski, 1979):

$$\begin{vmatrix} y_2^s - y_1^s & z_2^s - z_1^s \\ y_3^s - y_1^s & z_3^s - z_1^s \end{vmatrix}(x - x_1^s) + \begin{vmatrix} x_2^s - x_1^s & z_2^s - z_1^s \\ x_3^s - x_1^s & z_3^s - z_1^s \end{vmatrix}(y - y_1^s) + \begin{vmatrix} x_2^s - x_1^s & y_2^s - y_1^s \\ x_3^s - x_1^s & y_3^s - y_1^s \end{vmatrix}(z - z_1^s) = 0. \quad (12.15)$$

12.1.3.3 Parametric representation of other points

Finally, to find the ground elevations of other referencing points, a parametric representation can be used for the entire road cross section at each station point. That will be a vector-valued function:

$$\mathbf{X}_{S_i}(t) = \left[x_{S_i}(t), y_{S_i}(t), z_{S_i}(t)\right]^T, \quad (12.16)$$

where $t \in [t^{\min},\ldots,t_{-n,}t_{-n+1},\ldots,t_{-1},t_0,t_1,\ldots,t_{n-1},t_n,\ldots,t^{\max}]$

t_0 = parameter representing center line

t_n, t_{-n} = parameters representing any referencing points investigated

$t^{\max}, t^{\min}$ = ends of referencing points of a road cross section.

This cross section will be traced in such a way that the parameter *t* represents distance measured through the cross section in a discrete manner. If we wish to use the set of consecutive (continuous) centerlines for more precise earthwork volume estimation instead of using stations, the already developed parametric representation method can also be employed (Lovell, 1999).

12.1.4 Example study

To illustrate the application of the proposed methods, we design an artificial study area as shown in Figure 12.6. Each rectangle is 10 m × 10 m. A 50 m wide highway (30 m for the road, 20 m for both shoulders) crosses the study area.

There are two stations A and B, 100 m apart, and one middle point is located between them. Figure 12.7 illustrates the topography of the study area. It should be remembered that each square cell is represented only by its center's elevation.

To apply the Average End Area Method, two road elevations (30m for station A, 32 m for station B) are given and two ground elevations are obtained (50 m for station A, 40 m for station B).

The side slope angle is set at 45 degrees (see Figure 12.1). Consequently, the cross sectional areas are 1,400 m^2 for station A and 464 m^2 for station B. The total earthwork volume is then 93,200 m^3.

To use the Prismoidal Method, information about the middle point is needed. The road and ground elevations are 31 m and 35 m, respectively. The resulting cross sectional area at the middle point is 216 m^2.

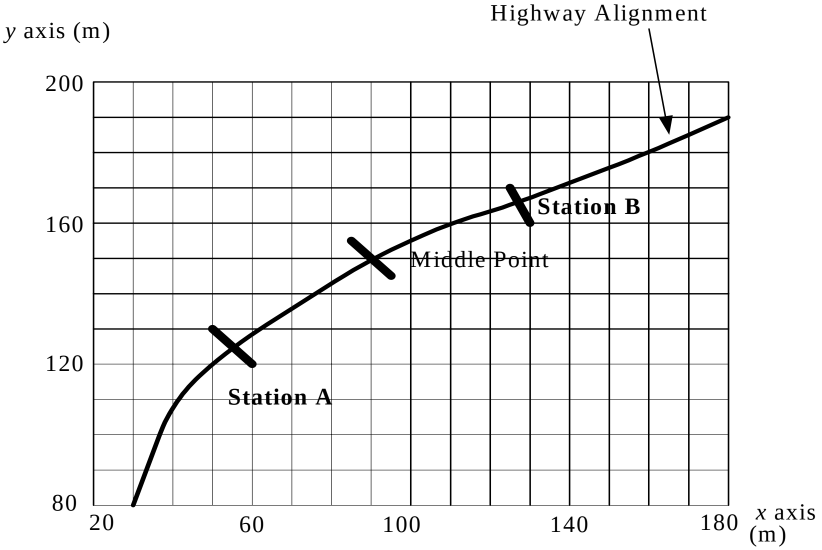


Figure 12.6: An example study area.

Using eqn (12.2), the total earthwork volume of 45,467 m^3 is found, which is about half of that (93,200 m^3) found with the Average End Area Method.

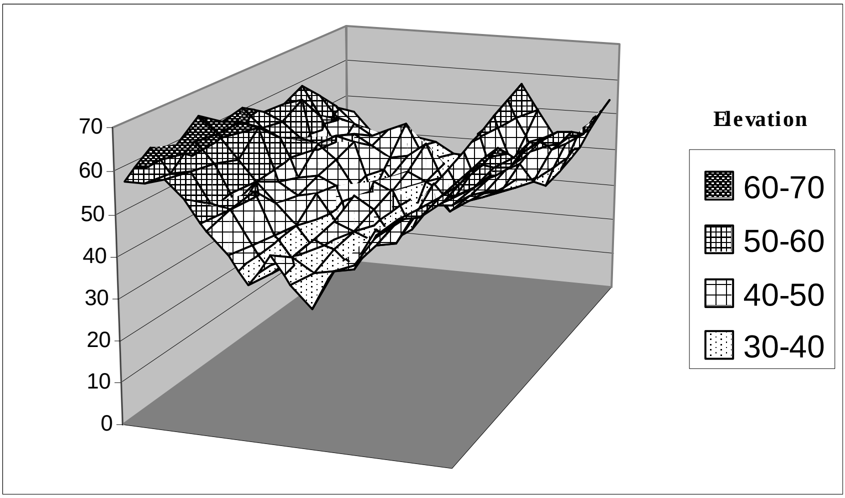


Figure 12.7: Topography of the study area.

For comparison, Moavenzadeh's method is also applied. The resulting cross section areas at stations A and B and at the middle point are 1,256 m^2, 590 m^2 and 304 m^2, respectively. The total earthwork volume using the Simpson's rule is then 51,209 m^3. This result is better than the 45,467 m^3 obtained with the original Prismoidal Method because Moavenzadeh's method divides the cross sections into several segments while 45,467 m^3 is based only on the center point elevation differences. The proposed method, planar interpolation, is finally applied. To demonstrate it, one particular case for finding the ground elevation of the center point at station B is illustrated in Figure 12.8. Recall that the previously applied methods use the cell's center point elevation (i.e., 40 m) as the ground elevation of station B. If the rectangle is relatively large and irregular (which is normal in real situations), it is very undesirable to use 40 m as the ground elevation at station B.

Since $\mathbf{S}_2 - \mathbf{S}_1 = [10,0,7]^T$ and $\mathbf{S}_3 - \mathbf{S}_1 = [10,10,5]^T$, the vector product of those, $(\mathbf{S}_2 - \mathbf{S}_1) \times (\mathbf{S}_3 - \mathbf{S}_1) = \left[\begin{vmatrix} 0 & 7 \\ 10 & 5 \end{vmatrix}, \begin{vmatrix} 10 & 7 \\ 10 & 5 \end{vmatrix}, \begin{vmatrix} 10 & 0 \\ 10 & 10 \end{vmatrix} \right]^T = [-70,-20,100]^T$. Hence, the plane equation we seek is: -70(x-115)-20(y-155)+100(z-35)=0.

The resulting ground elevation is 45.9 m. This process continues for other referencing points, determining the areas at stations A, B and the middle point to be 1,199 m^2, 670 m^2 and 452 m^2, respectively. The total earthwork volume using the proposed (planar interpolation) method is 61,283 m^3.

Table 12.1 shows the results obtained with four different methods. By comparing the figures in parentheses with the proposed planar interpolation method, significant differences from using different methods are found. Thus, the existing methods over or under estimate quite significantly.

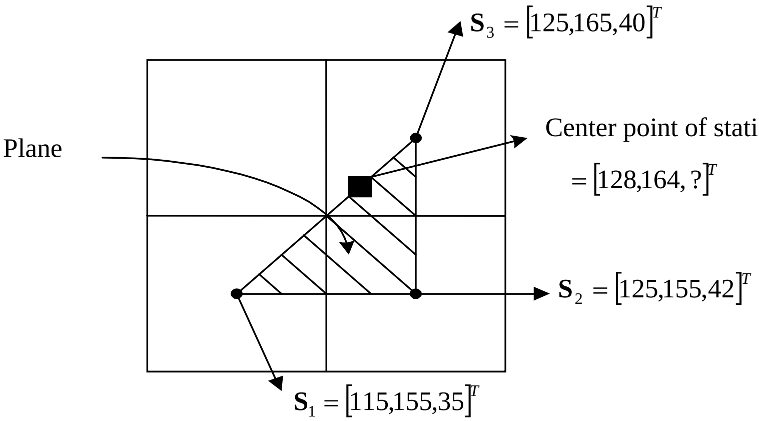

$\mathbf{S}_3 = [125,165,40]^T$
Plane
Center point of station
$= [128,164,?]^T$
$\mathbf{S}_2 = [125,155,42]^T$
$\mathbf{S}_1 = [115,155,35]^T$

Figure 12.8: Surrounding vectors of station B's center point.

Conceptually, since the Average End Area Method uses only two cross sections and one ground elevation difference per cross section, it provides less precision than the Prismoidal Method where one more cross section and Simpson's rule are added. Moavenzadeh's Method should improve the precision of cross section estimates by subdividing those into several pieces. Among the four methods only the newly proposed planar interpolation method determines precise ground elevations from existing databases for calculating cross section areas at each referencing point. Its accuracy should significantly exceed that of the previous three methods in most cases, especially if terrain is far from flat.

Table 12.1: Comparison of the results obtained with different methods.

	Cross section areas (m^2)			**Earthwork volume (m^3)**	**Use of Simpson's Rule**
	Station A	**Middle point**	**Station B**		
Average End Area Method	1,400 (+16.8%)	-	464 (−30.7%)	93,200 (+52.1%)	No
Prismoidal Method	1,400 (+16.8%)	216 (−52.2%)	464 (−30.7%)	45,467 (−25.8%)	Yes
Moavenzadeh Method	1256 (+4.8%)	304 (−32.7%)	590 (−11.9%)	51,209 (−16.4%)	Yes
Proposed Planar Interpolation Method	1,199	452	670	61,283	Yes

Even the proposed method relies on approximation. However, by finely slicing a cross section using the parametric representation, the approximation becomes increasingly precise, and sufficient for any planning applications. Finely slicing a cross section requires more calculation than existing methods, but that is a diminishing problem with fast computers that also avoid human errors. If a study area is relatively flat and we have enough station points, the Average End Area Method or the Prismoidal Method suffice. However, for mountainous areas and relatively wide highways, the proposed methods are more appropriate. Best of all, the proposed methods can automatically find any referencing point along a cross section using vector manipulation within automated processes. Developed methods work with some information, i.e., surveying points. Such information is available on-line from the web. The U.S. Geological Survey (USGS) provides topographic maps using brown contours to show the shape and elevation of the terrain.

12.2 Modeling intersection cost functions sensitive to alignments

The performance and costs of intersections significantly affect those of the entire transportation system and the regions they serve. Therefore, when developing highway alignment optimization processes, intersection cost functions should be included in them. However, those functions have been neglected since accommodating them requires very detailed cost formulations and extensive data analysis. This deficiency clearly reduces the usefulness of the developed models for highways that have intersections. This section pursues a detailed enough evaluation of intersection costs to provide cost functions with sufficient accuracy and sensitivity.

Table 12.2 shows the full range of intersection cost items. This section presents a method for formulating intersection cost functions including construction components (earthwork costs, right-of-way costs and pavement costs) and operating costs (accident costs, delay costs and fuel costs). Environment costs (noise costs and pollution (emission) costs) and drainage costs are left for future studies.

Table 12.2: Intersection cost components specification.

Main specification	Sub specification	Elements	Sensitivity to alignment
Construction costs	Earthwork costs	Physical elements	High
	Right-of-way costs	Physical elements Economic factors	High
	Pavement costs	Physical elements	High
Operational costs	User delay costs	Traffic considerations Human factors	Medium
	Accident costs	Human factors Physical elements	Medium
	Fuel costs	Traffic considerations Physical elements	Low
Environmental costs	Noise costs	Traffic considerations Physical elements	Low
	Pollution (emission) costs	Traffic considerations Physical elements	Low
Drainage costs		Traffic considerations Physical elements Economic factors	Low

12.2.1 Methodology for intersection construction cost modeling

Suppose there is a set of points of intersection (PI) defining centerlines of two intersecting roads shown in Figure 12.9 as the dotted lines. After applying additional procedures developed and explained in previous chapters for filling in vertical curves, horizontal curves and road cross section characteristics based on design standards, the intersecting alignments are sufficiently specified for evaluating their costs. The final crossing point will most likely deviate from its original PI location.

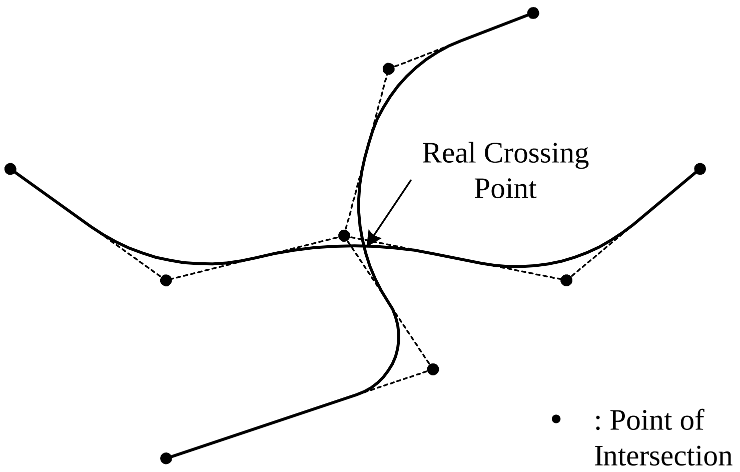


Figure 12.9: Points of intersections (PI's) and final crossing point deviating from its original PI.

Suppose there are two roads labeled I and II crossing at one particular point. First, the two dimensional (2D) coordinates of that point should be found so any other interesting points can be traced. There could be three different types of crossing:

- Type 1 : Tangent section of one alignment intersecting tangent section of the other alignment.
- Type 2 : Tangent section of one alignment intersecting circular section of the other alignment.
- Type 3 : Circular section of one alignment intersecting circular section of the other alignment.

Since type 3 is the most difficult and quite likely to actually occur, we focus on it. To find the coordinates of the intersection point, the equations of the circular curves should be obtained. Then we set those equal to each other and solve for the coordinates. To do this, the coordinates of the center point of two circles, and each alignment's points of tangency (PT) and points of curvature (PC) should be obtained. Ways of finding these points are already discussed in many sources (Jong, 1998; Jha, 2000; Wright, 1996; Hickerson, 1964). Figure 12.10 displays the characteristics of a typical circular section on an alignment.

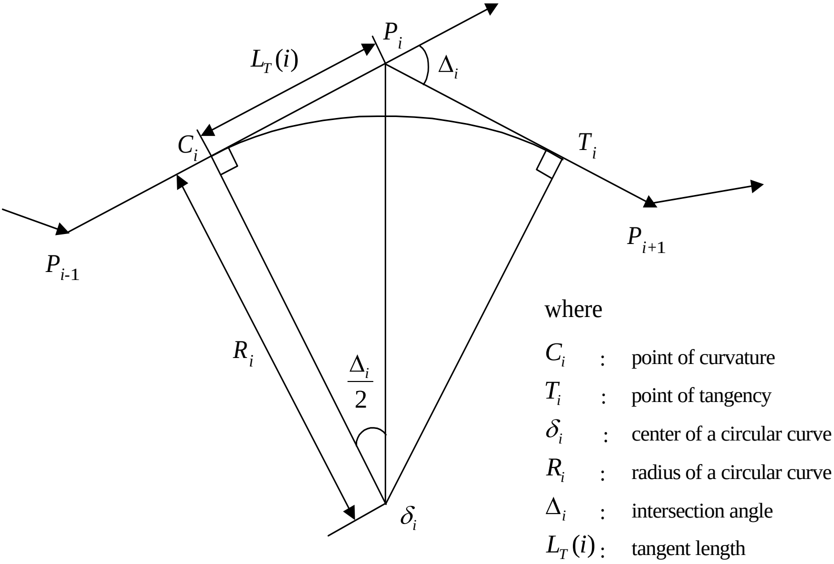

P_i
$L_T(i)$
Δ_i
C_i
T_i
P_{i+1}
$P_{i\text{-}1}$
R_i
$\frac{\Delta_i}{2}$
δ_i
where
C_i : point of curvature
T_i : point of tangency
δ_i : center of a circular curve
R_i : radius of a circular curve
Δ_i : intersection angle
$L_T(i)$: tangent length

Figure 12.10: Geometric specification of a circular curve.

Using vector analysis and trigonometry, the intersection angle (Δ_i), the coordinates of the point of curvature (C_i) and the point of tangency (T_i) are obtained as below:

$$\Delta_i = \cos^{-1}\left(\frac{(\mathbf{P}_i - \mathbf{P}_{i-1})\cdot(\mathbf{P}_{i+1} - \mathbf{P}_i)}{\|\mathbf{P}_i - \mathbf{P}_{i-1}\|\|\mathbf{P}_{i+1} - \mathbf{P}_i\|}\right), \tag{12.17}$$

$$\mathbf{C}_i = \begin{bmatrix} x_{C_i} \\ y_{C_i} \end{bmatrix} = \mathbf{P}_i + L_T(i)\frac{\mathbf{P}_{i-1} - \mathbf{P}_i}{\|\mathbf{P}_{i-1} - \mathbf{P}_i\|}, \tag{12.18}$$

$$\mathbf{T}_i = \begin{bmatrix} x_{T_i} \\ y_{T_i} \end{bmatrix} = \mathbf{P}_i + L_T(i)\frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{\|\mathbf{P}_{i+1} - \mathbf{P}_i\|}. \tag{12.19}$$

To find the coordinates of a circle's center point, let M_i be the middle point of the line segment between C_i and T_i, as shown in Figure 12.11.

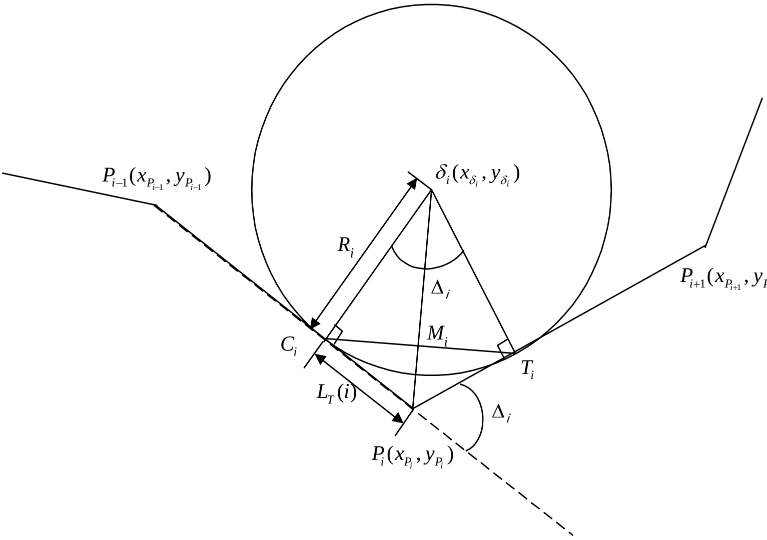

$P_{i-1}(x_{P_{i-1}}, y_{P_{i-1}})$
$\delta_i(x_{\delta_i}, y_{\delta_i})$
R_i
Δ_i
M_i
C_i
T_i
$L_T(i)$
Δ_i
$P_i(x_{P_i}, y_{P_i})$
$P_{i+1}(x_{P_{i+1}}, y_{P_{i+1}})$

Figure 12.11: Geometric relations among C_i, T_i, δ_i, and M_i.

It can be shown that M_i is located on the line connecting P_i and δ_i. Therefore, by vector analysis,

$$\boldsymbol{\delta}_i = \begin{bmatrix} x_{\delta_i} \\ y_{\delta_i} \end{bmatrix} = \mathbf{P}_i + R_i \sec\frac{\Delta_i}{2}\frac{\mathbf{M}_i - \mathbf{P}_i}{\left\|\mathbf{M}_i - \mathbf{P}_i\right\|} . \tag{12.20}$$

Now, the two circle equations crossing each other are

$$(x^I - x^I_{\delta_i})^2 + (y^I - y^I_{\delta_i})^2 = R_i^{I\,2} , \tag{12.21}$$

$$(x^{II} - x^{II}_{\delta_j})^2 + (y^{II} - y^{II}_{\delta_j})^2 = R_j^{II\,2} . \tag{12.22}$$

By setting these two equations to be equal, one single equation is obtained:

$$(x^I - x^I_{\delta_i})^2 + (y^I - y^I_{\delta_i})^2 - R_i^{I\,2} = (x^{II} - x^{II}_{\delta_j})^2 + (y^{II} - y^{II}_{\delta_j})^2 - R_j^{II\,2} . \tag{12.23}$$

A linear equation is obtained by solving eqn (12.23), since the square terms of x^2 and y^2 will be deleted. Substituting that linear equation into either of the two circle equations gives us the coordinates of the intersecting point. One of the two solutions is not feasible and will be automatically discarded within a computer program.

12.2.2 Pavement cost estimation

Estimating pavement costs is relatively simple. AASHTO (2001) design standards supply geometric specifications of additional flared areas providing paths for turning movements. To illustrate the pavement cost calculation, a typical four-leg flared intersection is shown in Figure 12.12.

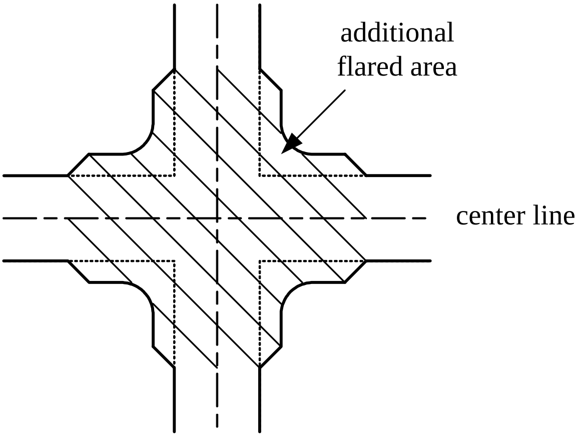


Figure 12.12: A typical four-legged intersection.

Although there could be numerous variations for providing traffic islands and divisional islands, all shaded areas shown in Figure 12.13 should be included in estimating pavement costs. Shaded areas can generally be divided into flared areas and remaining areas.

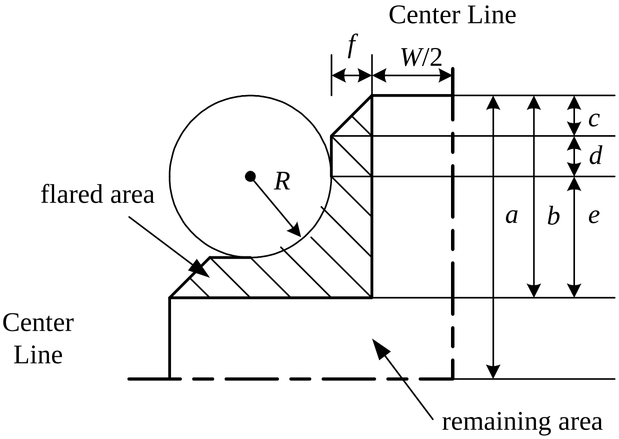


Figure 12.13: Flared and remaining areas of a typical four-legged intersection.

Based on Figure 12.13 and design standards for a two-lane rural highway intersection, the flared area (A_f) and remaining area (A_r) can be determined as follows:

$$A_r = \left(\frac{W}{2}\right)^2 + 2b\left(\frac{W}{2}\right) = \frac{W^2}{4} + bW\ , \tag{12.24}$$

$$A_f = 2\left(\frac{1}{2}\right)cf + 2df + \left(e^2 - \frac{\pi R^2}{4}\right) = f\left(c + 2d\right) + e^2 - \frac{\pi R^2}{4}\ . \tag{12.25}$$

Thus, the total pavement areas (A_p) is

$$\begin{aligned} A_p = 4\left(\frac{W^2}{4} + bW\right) + 4\left[f(c + 2d) + e^2 - \frac{\pi R^2}{4}\right] = \\ W^2 + 4bW + 4f(c + 2d) + 4e^2 - \pi R^2. \end{aligned} \tag{12.26}$$

Finally, total pavement costs (C_P) can be obtained by introducing a unit cost, K_P:

$$C_P = K_P[W^2 + 4bW + 4f(c + 2d) + 4e^2 - \pi R^2]\ , \tag{12.27}$$

where K_P = pavement cost per unit area (\$/m^2).

12.2.3 Earthwork boundaries and cost estimation

First, we must examine where earthwork boundaries should be placed. Boundary setting insures that costs for approach segments and intersections are not double-counted. To describe how earthwork boundaries are set and earthwork volumes are estimated, a typical fill intersection is introduced. For explanatory convenience, this book only uses one quadrant of a fill intersection. Following design standards, a constructed fill intersection may look like Figure 12.14.

Although estimating earthwork volumes based on Figure 12.14 is possible, just for the evaluation purposes, a simplified fill intersection is redesigned as shown in Figure 12.15. This simplified fill intersection may yield a slightly less accurate estimation but helps us develop a more understandable formulation.

Figure 12.15 shows where earthwork boundaries lie. Clearly, boundaries depend on the location of the center point of flared parts. Either the Average End Area Method or the Prismoidal Method can be used for approaching segments.

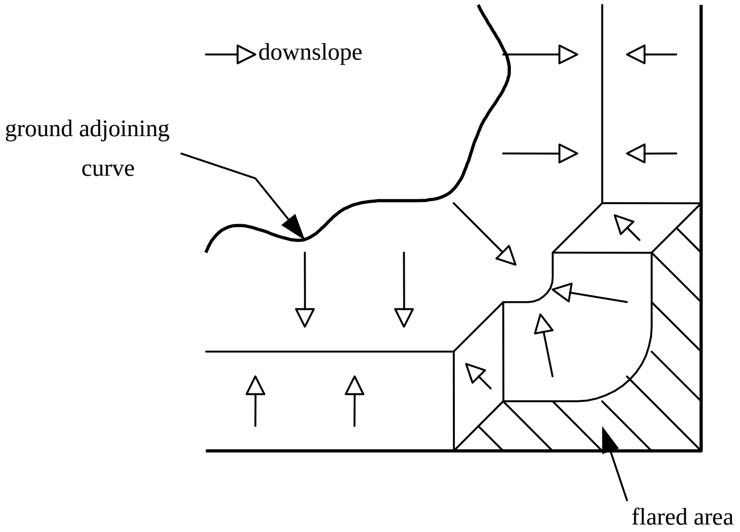


Figure 12.14: Quadrant of a typical fill intersection.

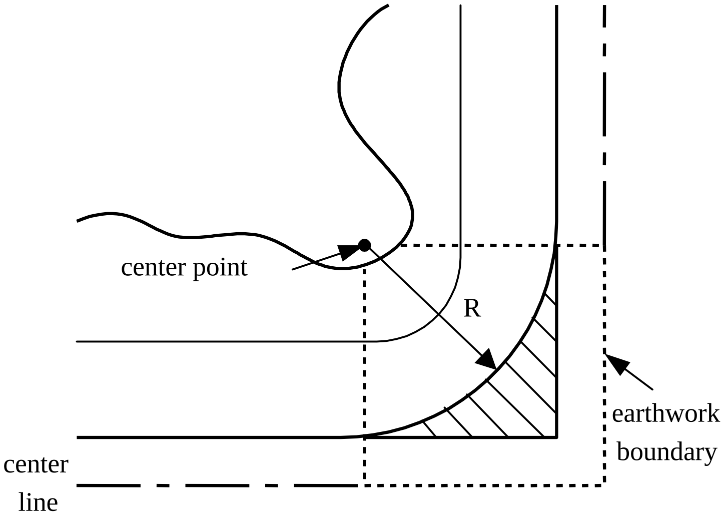


Figure 12.15: A simplified quadrant of a typical fill intersection.

12.2.3.1 Earthwork volume estimation

In estimating earthwork volumes and costs, the basic idea is to find the coordinates of the points in Figure 12.16. Afterwards, the average road elevations of each slice can be approximated using the coordinates of surrounding points.

An example shows how the coordinates of important points in Figure 12.16 (*A*, *B*, *D*, *E*, *F*, *G*, *H*, *I*, *J*, *K*, N_m, g_1, g_2, g_3 and g_4) are found. If it is possible to find the coordinates of all these points, subdividing the area into arbitrarily many slices can be done easily. Among many points in Figure 12.16, it should be noted that the coordinates of $I(x, y)$ are already found from the previous section, and the coordinates of $O(x_o, y_o)$ and $K(x_k, y_k)$ are taken from design standards. Based on this information, the other points can also be obtained.

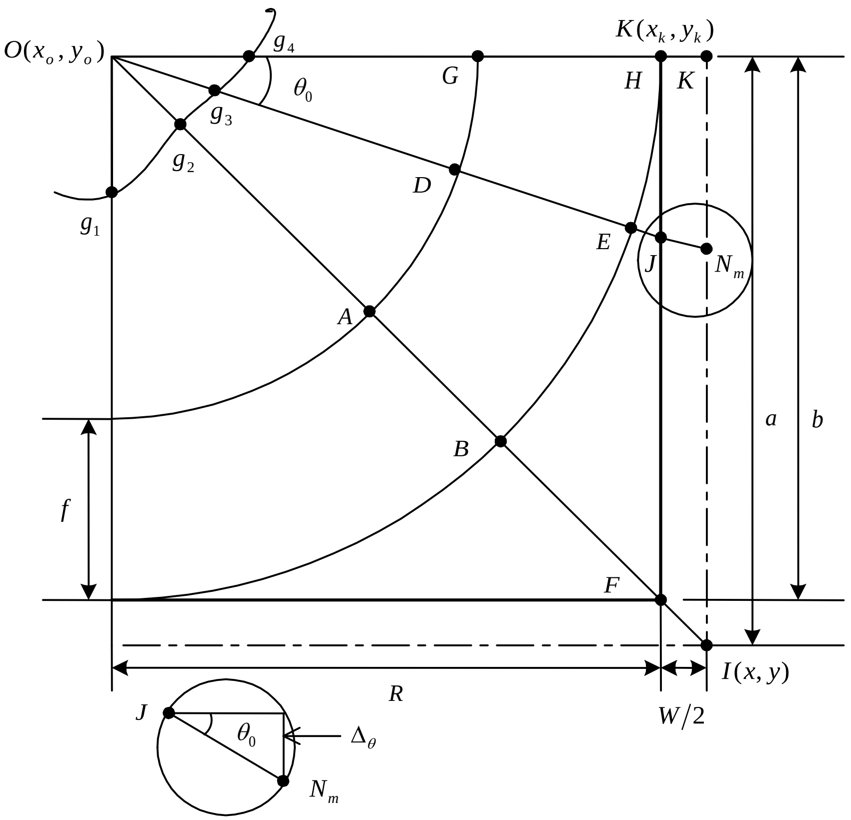


Figure 12.16: Important points for determining coordinates.

The coordinates of points *A*, *B* and *F* lying on the line segment between the intersecting point ($I(x,y)$) and the center point ($O(x_o, y_o)$) for the flared area, can be found simply by using vector operations:

$$\mathbf{A} = \mathbf{O} + (R - f)\frac{\mathbf{I}-\mathbf{O}}{\|\mathbf{I}-\mathbf{O}\|}, \tag{12.28}$$

$$\mathbf{B} = \mathbf{O} + R\frac{\mathbf{I}-\mathbf{O}}{\|\mathbf{I}-\mathbf{O}\|}, \tag{12.29}$$

$$\mathbf{F} = \mathbf{O} + \left[\|\mathbf{I}-\mathbf{O}\| - \frac{W}{\sqrt{2}}\right]\frac{(\mathbf{I}-\mathbf{O})}{\|\mathbf{I}-\mathbf{O}\|}, \tag{12.30}$$

where $\| \ \|$ represents the norm(or length) of a vector.

The above vector manipulation can also be used to find the coordinates of *G* and *H*:

$$\mathbf{G} = \mathbf{O} + (R - f)\frac{\mathbf{K}-\mathbf{O}}{\|\mathbf{K}-\mathbf{O}\|}, \tag{12.31}$$

$$\mathbf{H} = \mathbf{O} + R\frac{\mathbf{K}-\mathbf{O}}{\|\mathbf{K}-\mathbf{O}\|}. \tag{12.32}$$

For more general cases such as points *D, E, J* and N_m, introducing a small value is needed, Δy defined by $\Delta y = b/n$, where *n* is a user selected value. Next, let *m* be any multiple number of Δy $(\Delta y \le m \le b)$ and N_m be the point located $m \times \Delta y$ away from $I(x,y)$. Then, the coordinates of *D, E* and N_m are

$$\mathbf{D} = \mathbf{O} + (R - f)\frac{\mathbf{N_m}-\mathbf{O}}{\|\mathbf{N_m}-\mathbf{O}\|}, \tag{12.33}$$

$$\mathbf{E} = \mathbf{O} + R\frac{\mathbf{N_m}-\mathbf{O}}{\|\mathbf{N_m}-\mathbf{O}\|}, \tag{12.34}$$

$$\mathbf{N_m} = \mathbf{I} + m\frac{\mathbf{K}-\mathbf{I}}{\|\mathbf{K}-\mathbf{I}\|}. \tag{12.35}$$

Now, the only remaining point needed is *J*. Finding *J*’s coordinates requires finding the angle θ_o between two vectors, $\mathbf{N_m} - \mathbf{O}$ and $\mathbf{K} - \mathbf{O}$.

$$\theta_{\mathbf{o}} = \cos^{-1}\left(\frac{(\mathbf{N_m}-\mathbf{O})\cdot(\mathbf{K}-\mathbf{O})}{\|\mathbf{N_m}-\mathbf{O}\|\|\mathbf{K}-\mathbf{O}\|}\right), \text{ where } \cdot = \text{inner (dot) product.} \tag{12.36}$$

Therefore,

$$\Delta_\theta = \frac{W}{2}\tan\theta_{\mathbf{o}} = \frac{W}{2}\tan\left[\cos^{-1}\frac{(\mathbf{N_m}-\mathbf{O})\cdot(\mathbf{K}-\mathbf{O})}{\|\mathbf{N_m}-\mathbf{O}\|\|\mathbf{K}-\mathbf{O}\|}\right], \tag{12.37}$$

and the size of vector $\mathbf{N_m}-\mathbf{J}$ is

$$\|\mathbf{N_m}-\mathbf{J}\| = \sqrt{\frac{W^2}{4}+\left(\frac{W}{2}\tan\theta_o\right)^2}\ . \tag{12.38}$$

Finally, the coordinates of point **J** are

$$\mathbf{J} = \mathbf{O} + \left[\|\mathbf{N_m}-\mathbf{O}\| - \sqrt{\frac{W^2}{4}+\left(\frac{W}{2}\tan\theta_o\right)^2}\right]\frac{(\mathbf{N_m}-\mathbf{O})}{\|\mathbf{N_m}-\mathbf{O}\|}\ . \tag{12.39}$$

The next important task is to find the coordinates of adjoining ground points, g_i. As an example, finding the coordinates of g_2 in Figure 12.17 is illustrated. Figure 12.17 illustrates the vertical profile between g_2 and *B*.

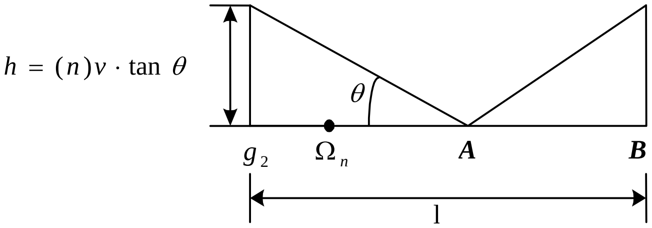


Figure 12.17: Vertical profile between g_2 and ***B***.

Suppose *l* is cut into several segments using a small unit distance, *v*. Let $\mathbf{\Omega}_n$ s be the consecutive coordinates specified by increasing *v*, such as (1)*v*, (2)*v*,…..,(*n*)*v*. Then those coordinates can be found using information already obtained above:

$$\mathbf{\Omega}_n = \begin{bmatrix} x_{\Omega_n} \\ y_{\Omega_n} \end{bmatrix} = \mathbf{A} + (n(v))\frac{\mathbf{O}-\mathbf{A}}{\|\mathbf{O}-\mathbf{A}\|}\ . \tag{12.40}$$

This process continues until the ground elevation of $\mathbf{\Omega}_n$, ($Z^g_{\Omega_n}$), is no less than the height *h*, i.e.,

$$Z^g_{\Omega_n} \geq h = (n)v\tan\theta\ . \tag{12.41}$$

Using eqns (12.40) and (12.41) iteratively, g_2 can be found. Meanwhile, in some cases, it is possible that ground adjoining curves might be located beyond the center point of the flared area, as shown in Figure 12.18.

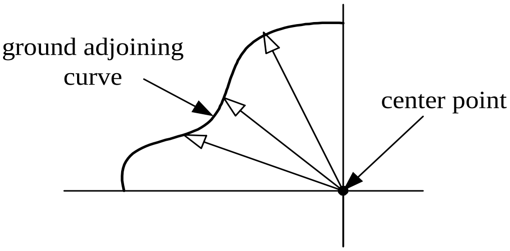


Figure 12.18: Ground meeting curves located beyond the center point.

In these cases, the coordinates of ground adjoining curves could also be obtained by extending the line segments from the center point.

12.2.3.2 Earthwork cost estimation

Given ground elevation databases, costs can be estimated by calculating the base areas of the relevant cells using previously found coordinates. For instance, to determine the base area surrounded by points *A*, *B*, *D* and *E* in the left part of Figure 12.19, that shape needs to be approximated into a simplified form, as shown in the right part of Figure 12.19, for easy subdivision into triangles.

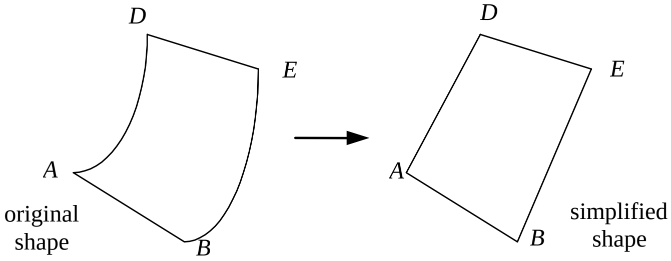


Figure 12.19: Simplified shape of the area *ABED*.

Based on the simplified shape, two triangles, *ADE* and *ABE*, can be created (alternatively *BAD* and *BDE* would be possible). Then, the base areas (A_b, m^2) are obtained as follows:

$$A_b = \left(\frac{1}{2}\right)\left[\left\|(\mathbf{B}-\mathbf{A})\times(\mathbf{E}-\mathbf{A})\right\| + \left\|(\mathbf{E}-\mathbf{A})\times(\mathbf{D}-\mathbf{A})\right\|\right]. \qquad (12.42)$$

This vector method can be used for any shapes that are separable into triangles. To find the earthwork volumes, two elevations are needed: (1) base elevation and (2) ground elevation. Here we simply average surrounding points' elevations (e.g., *A, B, D* and *E* in Figure 12.19) for finding the base elevation of a parcel. The corresponding ground elevation can be similarly obtained from the surrounding points' ground elevations available from existing databases. Although simple, this averaging method becomes quite precise if Δy is set to be sufficiently small.

Suppose there is a total of *T* parcels in the intersection. Then, the total earthwork (fill) volumes (E_V) are

$$E_V = \sum_{i=1}^{T} A_i^b \left(Z_{b_i}^{ave} - Z_{g_i}^{ave}\right), \qquad (12.43)$$

where A_i^b = base area of cell *i*
$Z_{g_i}^{ave}$ = average ground elevation of cell *i*
$Z_{b_i}^{ave}$ = average base elevation of cell *i*.

Therefore, total earthwork costs (C_E) is

$$C_E = K_F E_V, \qquad (12.44)$$

where K_F = filling cost per cubic meter (\$/m^3).

12.2.4 Right-of-way boundaries and cost estimation

By adding intersections to alignments, it is expected that the alignment right-of-way costs may increase. The coordinates of additional boundaries can be found in previous sections. Ground adjoining curves (g_1 through g_4 of Figure 12.16) already give us ways of finding right-of-way cost calculation boundaries.

Given newly found boundary information, a method is needed to estimate right-of-way costs by identifying the properties affected by the new intersection design. To do this, Jha's (2000) method is employed is this book. Jha divided right-of-way costs into three sub items: (1) temporary easement costs, which are defined as the partial taking of a property during the construction, (2) just compensation costs combining damage, site improvements and cost of the fraction of property taken by the alignment, and (3) appraisal fees. That is,

$$C_{RW} = \sum_{i=1}^{n} C_{RW_i} = \sum_{i=1}^{n} (C_{TE_i} + C_{JC_i} + C_{AF_I}), \qquad (12.45)$$

where C_{TE_i} = cost of the fraction of property i taken for temporary easement
C_{JC_i} = just compensation paid for property i
C_{AF_i} = appraisal fees for property I, and,

$$C_{JC_i} = C_{DP_i} + C_{DS_i} + C_{SI_i} + C_{F_i}, \qquad (12.46)$$

where C_{DP_i} = cost of damage to the value of property i
C_{DS_i} = cost of damage to structures on property i
C_{SI_i} = cost associated with site improvements of property i
C_{F_i} = cost of the fraction of property i taken for the alignment or intersection.

Generally, the computation takes into account the residual values of properties and pieces of properties left when a given alignment or an intersection is implemented. These values are affected by the size, shape and relative isolation of properties. The estimation procedures largely automate and computerize the existing appraisal process of the Maryland State Highway Administration Office of Real Estate.

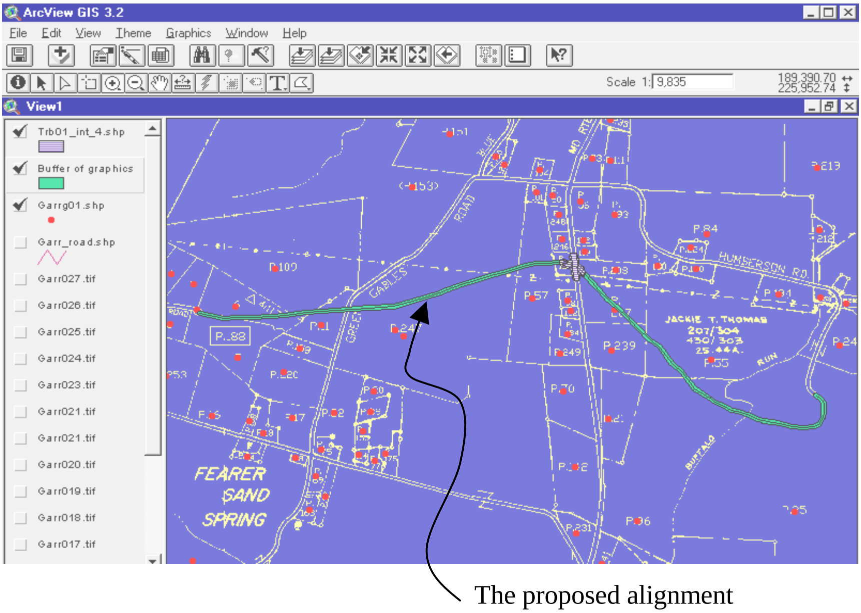
12.2.5 An example study for right-of-way cost estimation

In order to illustrate the application of the proposed highway alignment optimization method, an example from Kim, Jha and Schonfeld (2004), is introduced.

A new two-lane highway to connect two existing roads is considered (Hollis-Beaulieu Rd and Humberson Rd. in Garrett County, Maryland) to reduce congestion along MD Rt. 42. An optimized alignment is obtained by running the previously developed highway design optimization model and shown in Figure 12.20. It is observed that the optimized alignment intersects with Green Gables Rd. and MD Rt. 42.

The proposed intersection affects five properties, namely # 57, 208, 119, 136, and 207. In these, the land uses are either agricultural or residential and their unit land values range from \$0.038/m^2 to \$3.089/m^2 (Table 12.3). Some of the properties have structures on them whose values must be considered in right-of-way cost computation. Costs due to temporary occupation of surrounding lands during construction (also known as temporary easement) should also be considered.

Using the proposed method, the coordinates of points defining the boundary of the proposed intersection (including additional flared area) are obtained and input to Jha’s algorithm. The total right-of-way cost thus obtained by Jha’s method is $117,209.09. The flared areas cause the total right-of-way to increase due to affected structures nearby. The cost of additional right-of-way to be acquired due to the flared areas accounts for 55% of the total land cost.



The proposed alignment

Figure 12.20: Configuration of the proposed alignment.

The cost breakdown is shown in Table 12.3. The total intersection area (including the flared area) is 2,144.28 m^2. Such computations become very useful when extended from a single intersection to an automated highway alignment optimization model where many intersections may have to be analyzed for each candidate.

Table 12.3: Property and cost data for the example study.

Property #	Unit cost ($/m^2)	Land use	Structure cost ($)	Area (m^2)	Land cost ($)	Easement cost ($)	Total cost ($)
57	0.038	Agricultural	0	30.65	1.17	0.12	1.29
208	0.744	Agricultural	0	402.71	299.64	29.96	329.61
119	3.089	Residential	7,830	23.05	71.20	7.12	7,908.32
136	2.808	Residential	65,260	736.42	2,068.15	206.82	67,534.97
207	0.728	Residential	41,350	103.50	75.38	7.54	41,432.91
Total cost = $117,207.09							

12.2.6 Intersection accident costs

When intersections are added to an alignment, accidents tend to increase since intersections are more hazardous than basic highway segments. Hence, accident costs should be appropriately estimated. No new accident models are developed here; this book adopts the most suitable models from other studies.

Many different models have been developed to predict frequencies of accidents based on different intersection configurations (Lau and May, 1988; Vogt and Bared, 1998; Sayed and Rodriguez; 1999; Khan *et al*, 1999). Table 12.4 shows possible explanatory variables for intersection accident prediction models found through the literature review in various studies, including geometrics, traffic characteristics, control types, location, operation and maintenance, weather, surrounding conditions and driver characteristics (human factors).

Table 12.4: Independent variables for intersection accident prediction models.

Characteristics	**Factors**	**Elements**
Geometrics	1. Horizontal curvature	Major road, minor road or the average of two roads
	2. Vertical profile	Sag or crest. Grade for each road for each type
	3. Crossing angle	
	4. Terrain	Mountainous, rolling or flat
	5. Road width	Major and minor road
	6. Number of lanes	Major and minor road
	7. Lane width	Major and minor road. Average lane width for each road
	8. Channelization	No support, Right turn lanes (one or two), Left turn lanes (one or two)
	9. Sight distances	Absolute values for each road for each direction, or Average values for each road
	10. Shoulder	Width, surface condition
Traffic characteristics	1. *ADT*	Major and minor road
	2. Volume distribution over the specified duration	Major and minor road
	3. Turning Volumes	Major to minor and minor to major

Table 12.4 Independent variables for intersection accident prediction models (continued).

Characteristics	Factors	Elements
Traffic characteristics		
	4. % of heavy vehicles	Major and minor road
	5. % of buses	Major and minor road
	6. Design speed	Major and minor road
	7. Posted speed	Major and minor road
	8. Pedestrian volumes	Total volumes
	9. Bicycle volumes	Total volumes
Control types	1. Pre-timed	Two or multiple phases
	2. Semi-actuated	Two or multiple phases
	3. Full-actuated	Two or multiple phases
	4. Stop controlled	On minor road
	5. All way stop controlled	
Location	1. Rural/suburban/urban	
Operation and maintenance	1. Lighting	Excellent, normal or poor
	2. Lane marking	Excellent, normal or poor
	3. Pavement condition	Excellent, normal or poor
	4. Road sign	Visibility, reflectivity, luminosity
Weather	1. No. of dry/wet days (or % of time)	
	2. No. of snow/slush days (or % of time)	
	3. No. of ice/packsnow days (or % of time)	
Surrounding conditions	1. Roadside hazard rating	1 to 7 (Zegeer, 1986)
	2. Number of driveways	
	3. Local income level	High, middle or low
Driver characteristics (human factors)	1. Gender	
	2. Age	
	3. Occupation	
	4. Driving (safety) records	

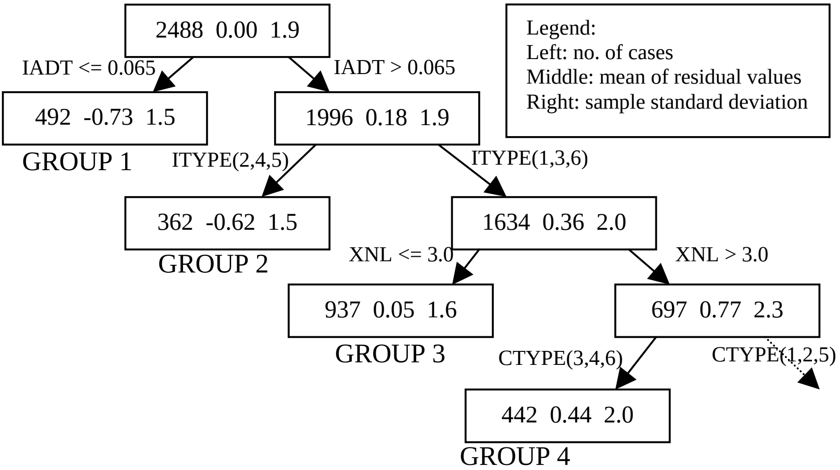
Based on thorough reviews of the safety literature, this book employs two different methods for two representative intersection types on two-lane highways. Lau and May's model (1988) is used for signalized intersections while Vogt and Bared's model (1998a&b) is adopted for two-way stop controlled (TWSC, on minor road) intersections. All-way stop controlled (AWSC) types are excluded since those are less likely to be employed for two-way rural highways.

Lau and May developed an accident prediction model for signalized intersections using the Traffic Surveillance and Analysis System (TASAS) in California. They derived macroscopic-type models for only injury accident models per year but also argued that the models can be used for fatal and property damage only (PDO) accidents based on a grouping and classifying technique called Classification and Regression Trees (CART). First, they developed the base model using traffic intensity characteristics:

$$\text{FIACCYR} = 0.61856 + 0.16911 \times \text{MVYR}, \tag{12.47}$$

where FIACCYR = number of expected injury accident per year
MVYR = millions of vehicles entering an intersection.

The residuals of the base model are analyzed with CART, which generates a nine fold cross-validation tree dividing observed intersections (totaling 2,488) into nine groups with similar accident characteristics, as shown in Figure 12.21.



IADT = ratio of cross street ADT to total ADT
ITYPE: 1=four-legged 2=multilegged, 3=offset, 4=T, 5=Y and 6=other
XNL=cross street no. of lanes
CTYPE:1=pretimed two-phase, 2=pretimed multiphase, 3=semi-traffic actuated two-phase, 4= semi-traffic actuated multiphase, 5=full traffic actuated two-phase and 6=full traffic actuated multiphase

Note:
(1) adopted from Lau and May (1988)
(2) only groups 1 through 4 are shown here
(3) 9-fold cross validation tree s.e.=0.3, r(d)=0.90

Figure 12.21: Grouping of intersections by CART.

Among those nine groups, three are selected for this study. Group 2 is selected for a three-leg intersection and group 3 is adopted for a four-leg with a cross (minor) street having not more than three lanes while group 4 is for a four-leg with a cross street having more than three lanes. To get each group's final estimate, the mean of residual values is added to the result of the base model.

Recently, Vogt and Bared (1998) developed two accident prediction models, which used several geometric factors as independent variables, for three-leg and four-leg intersections that are stop-controlled on the minor leg. These used Minnesota databases obtained from Highway Safety Information System (HSIS) files for the period 1985 to 1989. The estimated models are as follows:

Minnesota three-legged model:

$$\hat{y} = \exp\begin{bmatrix} -11.48 + 0.82LOG(ADT1) + 0.51LOG(ADT2) + 0.26VI \\ + 0.036HI + 0.027SPDI + 0.18HRI + 0.24RT \end{bmatrix}, \qquad (12.48)$$

Minnesota four-legged model:

$$\hat{y} = \exp\begin{bmatrix} -7.74 + 0.64LOG(ADT1) + 0.58LOG(ADT2) + 0.33VI \\ + 0.053DELT + 0.11ND \end{bmatrix}, \qquad (12.49)$$

where $ADT1$ = average daily traffic on mainline, vehicles per day.

$ADT2$ = average daily traffic on minor road, vehicles per day.

V = crest curve grade rate, $(1/m)\sum V\{j\}$ in percent per 100 ft, taken into account if any portion of the crest curve falls within 250 ft of the intersection on the major road.

HI = degree of curve for horizontal curves, $(1/n)\sum DEG\{i\}$ in degrees per 100 ft, averaged over all horizontal curves, any portion of which lies within 250 ft of the intersection on the major road.

$SPDI$ = posted speed on the main road, averaged as necessary, in miles per hour.

HRI = roadside hazard rating (1 to 7) within 250 ft of the intersection on the main road.

RT = if there is exactly one right-turn lane on the main road for three-legged intersections, this variable is 1; otherwise, it is 0.

$DELT$ = adjusted intersection angle, $= (\alpha - 15)^2/100$, in degrees squared.

ND = number of driveways within 250 ft of the intersection on the main road.

After obtaining accident frequencies for intersections with the above models, accident costs are calculated by multiplying accident frequencies with the unit costs, U_c, per accident.

12.2.7 Intersection delay costs

Intersections inherently generate additional delays while providing better access for highway users. These delays should also be incorporated into intersection cost function formulation. For delay cost estimation, this book adopts already developed models since this subject has been studied for a long time.

Webster (1958) developed the following model for an isolated signalized intersection:

$$d = \frac{c(1-\lambda)^2}{2(1-\lambda x)} + \frac{x^2}{2q(1-x)} - 0.65\left(\frac{c}{q^2}\right)^{\frac{1}{3}} x^{(2+5\lambda)}, \tag{12.50}$$

where d =average delay per vehicle on the particular approach of the intersection
c = cycle time
λ = proportion of the cycle which is effectively green for the phase under consideration (i.e., g/c)
q = flow (veh/sec)
s = saturation flow (veh/sec)
x = degree of saturation (i.e., $q/\lambda s$) .

The first two terms of Equation 12.50 have forms based on queuing theory while the last is an empirical correction term.

The TRANSYT methods (Robertson, 1969; Wallace *et al*, 1991; Wallace *et al*, 1998) also estimate delays at intersections when optimizing signal timings for isolated intersections, arterials and networks based on the minimization of a performance index combining stops and delays (Park, 1998). Eqns (12.51) and (12.52) show how delays are estimated. (TRANSYT version 7.2 cannot deal with saturated (queue blocking) conditions, while TRANSYT VERSION 8.1 can, by providing several strategies which are not shown in eqn 12.52.)

TRANSYT-7F version 7.2:

$$D_u + D_{rs} = \left[\frac{\sum_t^N q_t}{N}\right] + \left[900Tx^2\left\{(x-1) + \sqrt{(x-1)^2 + \left(\frac{4x}{cT}\right)}\right\}\left(\frac{v}{3600}\right)\right], \tag{12.51}$$

TRANSYT-7F version 8.1:

$$D_u + D_{rs} = \left[\frac{\sum_t^N q_t}{N}\right] + \left[900Tx^2\left\{(x-1) + \sqrt{(x-1)^2 + \left(\frac{mx}{cT}\right)}\right\}\left(\frac{v}{3600}\right)\right], \tag{12.52}$$

where D_u = uniform delay in veh-hr/hr
q_t = queue length in vehicles during step *t*.
N = number of steps in cycle
D_{rs} = random-plus-saturation delay in veh-hr/hr
T = period length, normally 60 minutes for undersaturated conditions
x = degree of saturation
c = capacity of phase in vph
v = demand volume in vph
m = an incremental delay calibration term representing the effect of arrival type and degree of platooning.

In TRANSYT-7F version 7.2, *m* is a constant equal to 4 on the basis of 1985 HCM (Highway Capacity Manual). In the 1994 update of the HCM, *m* is a function of the arrival type (Park, 1998). The value of *m* was introduced to accommodate lower values of random delay on approaches in coordinated systems due to the metering effect of a coordinated system (Wallace *et al*, 1998). The Highway Capacity Manual (HCM, 2001) also provided a model for estimating the delays of signalized intersections which can deal with oversaturated conditions:

$$\begin{aligned} d &= d_1(DF) + d_2 \\ &= 0.38C\frac{[1-(g/C)]^2}{[1-(g/C)X]}(DF) + 173X^2\left[(X-1)+\sqrt{(X-1)^2+\frac{mX}{c}}\right], \end{aligned} \qquad (12.53)$$

where d_1 = uniform delay, sec/veh
d_2 = incremental delay, sec/veh
g = effective green phase for lane group
C = cycle length
c = capacity of lane group, vph
X = v/c ratio for lane group
m = adjustment for early or late platoon arrivals
DF = delay adjustment factor (for progression).

This book adopts Webster's model for estimating the delays at signalized intersections since it focuses on two-way rural intersections where oversaturated conditions are rare. However, the HCM method will be employed when oversaturated conditions occur. For unsignalized intersections, eqn (12.54) of HCM is used:

$$D = \frac{3600}{C_x} + 900T\left[\frac{V_x}{C_x} - 1 + \sqrt{\left(\frac{V_x}{C_x} - 1\right)^2 + \frac{\left(\frac{3600}{C_x}\right)\left(\frac{V_x}{C_x}\right)}{450T}}\right], \tag{12.54}$$

where D = average total delay for a movement x (sec/veh)
V_x = volume for movement x, expressed as an hourly flow rate
C_x = capacity of movement x, expressed as an hourly flow rate
T = analysis period (usually, 15 minutes, T=0.25).

Eqn (12.54) only accounts for one particular movement. 12 movements for a four-leg intersection and 6 movements for a three-leg intersection should be considered for a full analysis. Intersection delay costs are finally obtained by multiplying the delays and the unit delay costs, U_s.

12.2.8 Intersection vehicle fuel costs

Another intersection cost sensitive to alignments is the vehicle fuel cost. The related costs to the fuel cost include oil consumption costs, tire wear costs and vehicle depreciation costs. However, those are not dominating. Vehicle fuel costs for basic highway segments are already well developed and employed by Jong (1998) based on the average running speed. However, additional fuel costs caused by a new intersection are neglected along with any other costs attributable to intersections.

There are four types of fuel cost models: (1) Instantaneous models, (2) Delay type models, (3) Speed type models and (4) Analytical Models. Instantaneous models need individual vehicle data, record direct measurements and use several vehicle types and various road conditions (Akcelik *et al*, 1983; Bowyer, 1986; Biggs, 1988). Delay type models are basically macroscopic simulation model using traffic measures of effectiveness such as delay and stops for signalized corridors or an isolated signalized intersection (FHWA, 1984; Bauer 1975; Courage and Parapar, 1975). Speed type models are usually regression models using networkwide variables such as travel speed, travel time and travel distance for streets (Evans *et al*, 1976; Herman and Ardekani, 1985). Lastly, analytical models use mathematical representations for vehicle movement approaches. They are basically aggregated models using fuel consumption rates for an isolated signalized intersection (Liao and Machemehl, 1998)

In this book, speed-type and delay-type models are used to develop a new model incorporating Jong’s approach. Jong developed the following fuel consumption model for a basic highway segment using multiple regression approach:

$$F = \alpha_0 + \alpha_1\overline{G} + \alpha_2\overline{V} + \alpha_3\overline{V}^2, \tag{12.55}$$

where F = fuel consumption (gallons/1000 miles)
$\overline{G}$ = grade of road section (%)
$\overline{V}$ = vehicle average running speed (mph).

Then, he included total traveled miles when calculating actual fuel consumption costs. Eqn (12.55) is well developed to cope with traffic and geometric characteristics. However it does not account for intersections' effects such as delays and stops.

The FHWA (1984) introduced the following delay type model when developing TRANSYT-7F traffic macroscopic simulation model:

$$F_{TRANSYT} = \beta_1 TT + \beta_2 D + \beta_3 S\ , \tag{12.56}$$

where $F_{TRANSYT}$ = fuel consumption in gallons per hour
TT = total travel in vehicle-miles per hour
D = total delay in vehicle-hours per hour
S = total stops per hour

This model predicts fuel consumption based on the measures of effectiveness (MOE's) produced by the simulation. Those MOE's are: (1) vehicle mile traveled, (2) total delays and total stops.

FHWA (1984) further developed a figure providing reduction of stops as a function of delay.

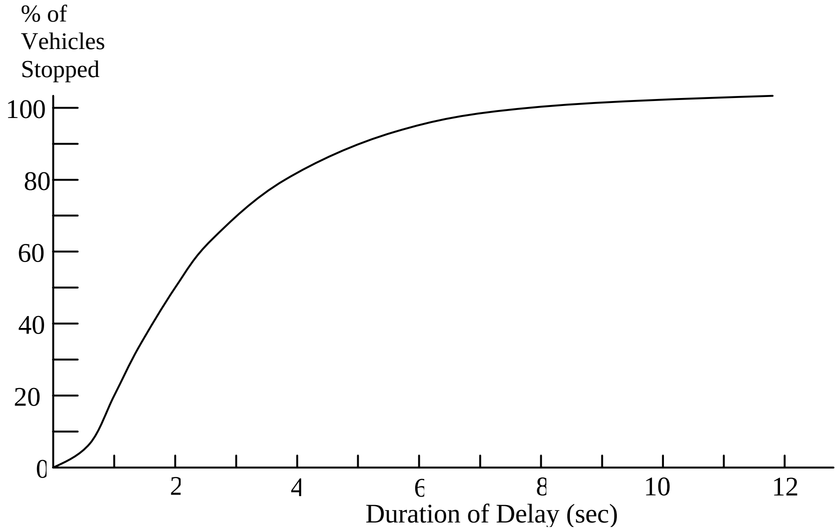


Figure 12.22: Reduction of stops as a function of delay.

Using this Figure 12.22 and eqns (12.50) and (12.54), the associated number of stops can be obtained. Then, the remaining problem is to find β_2 and β_3 in eqn. (12.53). Fortunately, there are average fuel consumption rates reported by Liao and Machemehl (1998). They estimated fuel consumption rates for each case from vehicle speed and acceleration/deceleration profile models and corresponding EPA fuel consumption data. Average fuel consumption rates for vehicles changing from speed V_i to speed V_j are calibrated and given in a table. Table 12.5 shows only two interesting values from that original table.

Table 12.5: Average fuel consumption rate from speed V_i to speed V_j.

Definition	**Fuel consumption rate**	
	Grams/sec	**Gallons (10^{-5})/sec**
Idle fuel consumption rate	0.3310	13.00
Change speed from desired speed to stop	0.6000	23.56

In summary, the fuel consumption costs for alignment alternatives are initially computed with eqn (12.55). Then, additional costs caused by intersections are added by incorporating delays (from eqns (12.50) and (12.54)) and stops (from Figure 12.22) with the associated average fuel consumption rates shown in Table 12.5. The fuel consumption model is not derived from reliable data sets and theoretical approaches. Rather, it is developed from very elementary and intuitive analysis. Although its reliability must be tested, the model is simple and easy to incorporate in preliminary highway planning.

12.3 Development of bridge cost functions

According to bridge engineering conventions, bridge costs are composed of superstructure costs and substructure costs. Those two cost components are functions of number of spans, span lengths, types, materials, pier heights and topography. There is no simple formula for estimating the bridge costs based on all these variables. Only theoretical linear functions for superstructure and substructure costs based on one variable, namely span lengths, are available from the literature (O’Connor, 1971). Those linear functions are based on simply supported composite (steel and concrete) girder bridges. Among many bridge types, since girder bridges have been most commonly employed as highway bridges, they are considered here to be most representative type.

Once the type, material and length of a bridge are set, the most important factors for bridge cost estimation are to determine (1) how many piers (substructures) should be selected and (2) where those piers should be located. In optimizing bridges, these two factors are interrelated. Also, the pier spacings inherently determine span lengths.

In bridge engineering, continuity is considered a very important factor in optimizing pier locations or span lengths. Continuity considerations favor equal spans or at least gradually varying spans. This book develops a simple approach for estimating bridge costs based on these continuity considerations.

The following example shows how the number of piers or span lengths for bridge can be optimized. Suppose that we consider a highway bridge which is 300 feet long, as shown in Figure 12.23.

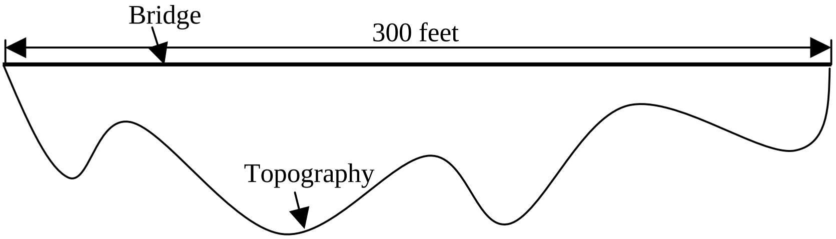


Figure 12.23: An example bridge for construction.

To maintain continuity (that is, equal spans), we should divide the 300-foot length by some number, say K_b. Suppose we choose a 30-foot, a 50-foot and a 100-foot for K_b's. Then, we have different number of piers and equally spaced spans for each K_b, as shown in Figures 12.24.

Based on configurations in Figure 12.24, we can compute the total bridge costs for each case based on the given number of piers and span lengths. (The details will be explained in section 12.3.1.)

This approach is pursued by numerically searching through different values of K_b until we obtain the optimal span as shown in Figure 12.25. (There is no guarantee that the bridge cost function is convex. Local optima may exist when a bridge is relatively long.)

The computation time depends on how may times we try using different values of K_b. Since this book is not intended to fully optimize bridges, some limited but acceptable number of trials is envisioned.

In the above method, it is notable that the search space has just one dimension, the span. The problem could be treated as two-dimensional by adding one more variable, the location of the first pier, but that is left to future extensions.

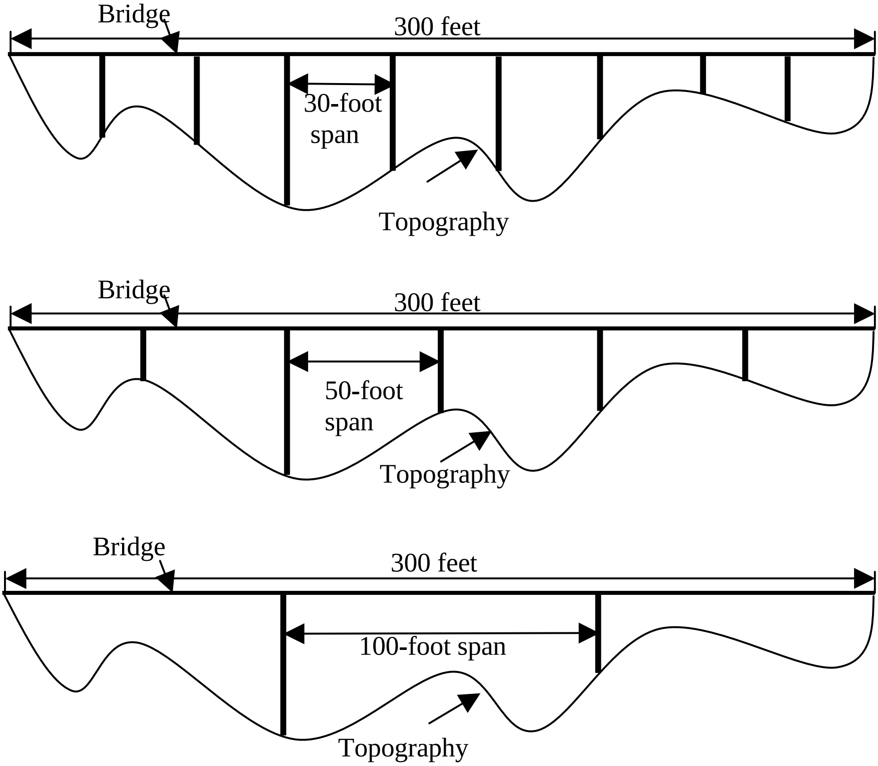


Figure 12.24: Equally spaced piers with varying spans.

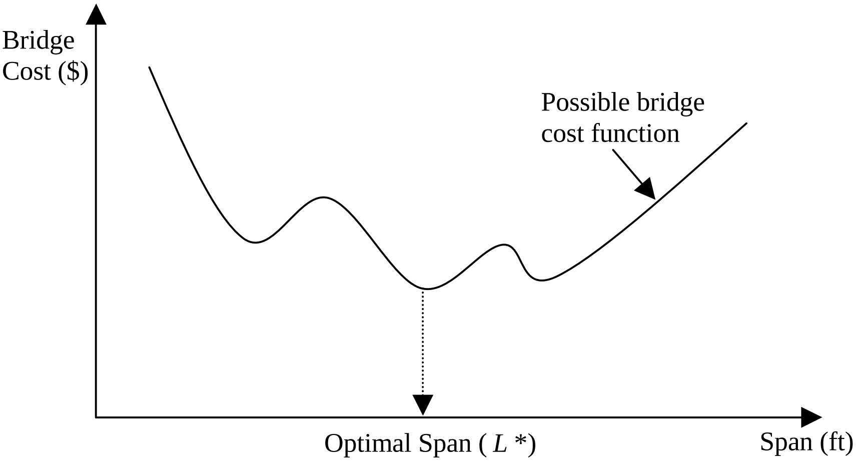


Figure 12.25: Graphical representation for getting optimal span of bridges.

To estimate bridge costs, this book adopts linear cost functions for superstructure and substructure already developed by O'Connor (1971). For the superstructure costs, Figure 12.26 can be used.

Based on Figure 12.26, O'Connor developed the linear cost function for the superstructure:

$$C_U^B = (a_1 + a_2 L), \tag{12.57}$$

where L = span length and the coefficients are differentiated by a girder spacing.

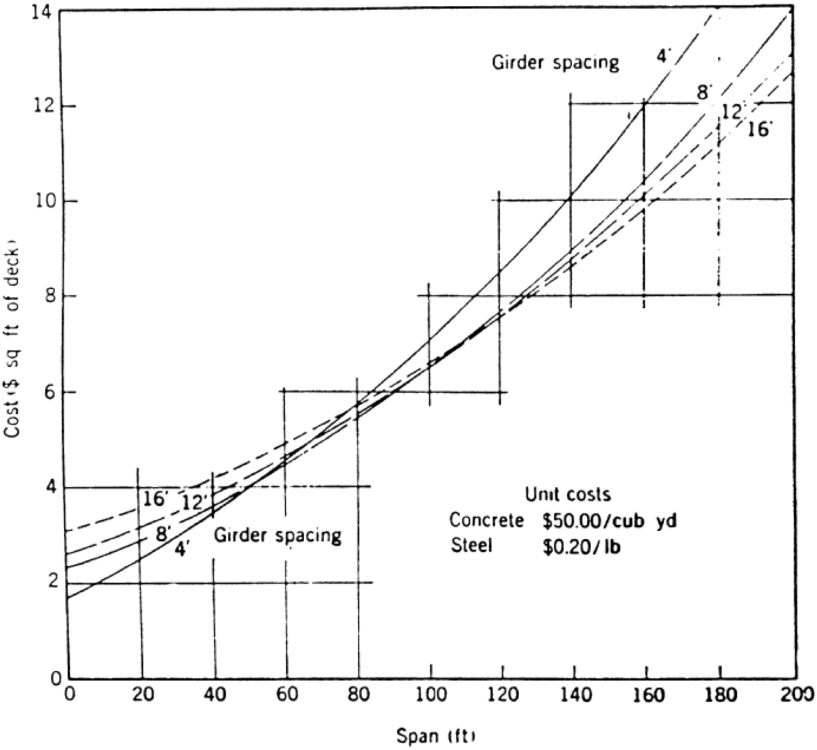


Figure 12.26: Theoretical superstructure costs of the composite girder bridge.

Figure 12.27 shows the substructure costs for simply supported girder bridges. It shows that costs per foot-width of substructure for variable pier heights increase linearly with span:

$$C_L^B = (a_3 + a_4 L). \tag{12.58}$$

The coefficients in eqn (12.58) differ for different pier heights.

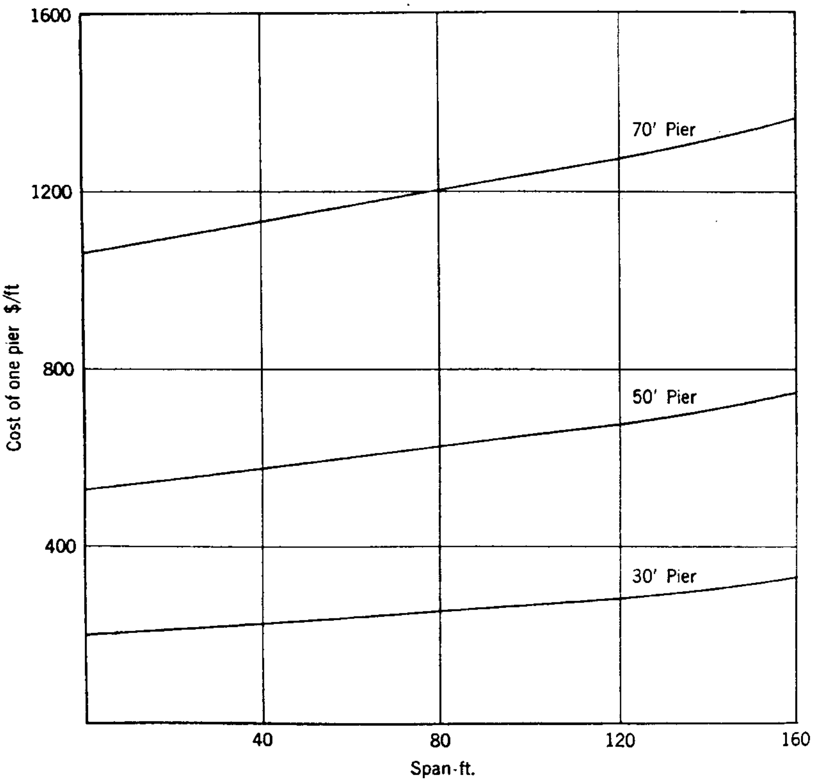


Figure 12.27: Foundation costs for bridges with various span lengths (1969 dollars).

12.4 Cost functions for grade separated structures (underpass and overpass)

Grade separation structures might be considered as small bridges. Therefore, bridge cost functions developed in section 12.3 are adopted just for the construction (both superstructure and substructure) costs.

If the elevation difference (Δh) between the ground and a new road satisfies the needed vertical clearance, all cost functions used for a basic highway segment can be employed except for bridge costs. If the elevation difference is not satisfied, additional earthwork and right-of-way costs for providing sufficient vertical and lateral clearances should be estimated. For the additional earthwork costs, let Δh, Δh^+ and Δh^- be elevation differences between a ground elevation and a new road, a needed overpass vertical clearance and a needed underpass vertical clearance, respectively.

Suppose that the vertical clearance is not satisfied (i.e., $\Delta h^- < \Delta h < \Delta h^+$). Then, additional earthwork costs are a function of distances between stations (Δs) and $\left|\Delta h^+ - \Delta h\right|$ (for overpass case) or $\left|\Delta h^- - \Delta h\right|$ (for underpass case).

Figure 12.28 shows a typical fill cross section where the road elevation should be raised by Δh^+.

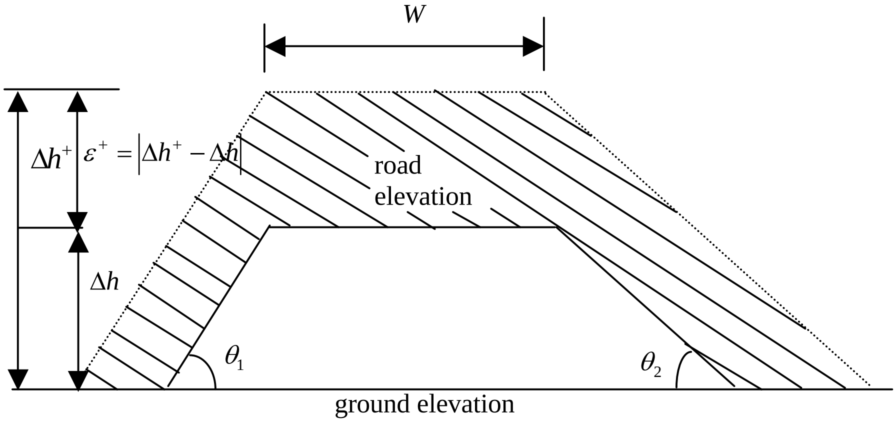


Figure 12.28: Typical cross section with insufficient vertical clearances.

Then, the additional cross sectional area (ΔA) is

$$\begin{aligned}\Delta A &= W\varepsilon^+ + \left(\Delta h^+ \cot\theta_1 - \Delta h \cot\theta_1\right) + \left(\Delta h^+ \cot\theta_2 - \Delta h \cot\theta_2\right) \\ &= \varepsilon^+\left(W + \cot\theta_1 + \cot\theta_2\right).\end{aligned} \tag{12.59}$$

Therefore, the additional earthwork volumes (ΔV) and costs (ΔC_E) are

$$\Delta V = \varepsilon^+\left(W + \cot\theta_1 + \cot\theta_2\right)\Delta s, \tag{12.60}$$

$$\Delta C_E = K_F \varepsilon^+\left(W + \cot\theta_1 + \cot\theta_2\right)\Delta s, \tag{12.61}$$

where K_F = unit filling cost per cubic yard.

Likewise, an additional right-of-way cost function might be formulated as a function of $\left|\Delta h^+ - \Delta h\right|$ for an overpass or $\left|\Delta h^- - \Delta h\right|$ for an underpass.

Intuitively, assuming relatively flat topography, the area affected by raising the road elevation by ε^+ is a quadratic function of ε^+.

$$\Delta C_R = \alpha_1^R \left(\varepsilon^+\right)^2 + \alpha_2^R \varepsilon^+ + \alpha_3^R \, . \qquad (12.62)$$

In eqn (12.62), the values of each coefficient depend on topography, associated side slope (θ), land use, sizes and shapes of associated parcels and other factors.

12.5 Interchange cost functions

As specified in the book scope, cost functions are formulated here for three types of interchanges: (1) diamond, (2) clover and (3) trumpet types. Diamond and clover types are for four-leg interchanges while trumpet types are for three-leg interchanges.

Interchange cost functions are basically compilations of already developed cost functions for other structures. For instance, Figure 12.29 shows centerlines of the associated roads and ramps for a diamond interchange and two major design criteria to consider: (1) major interchange leg length, l_d^b and (2) minor interchange leg length, l_d^s.

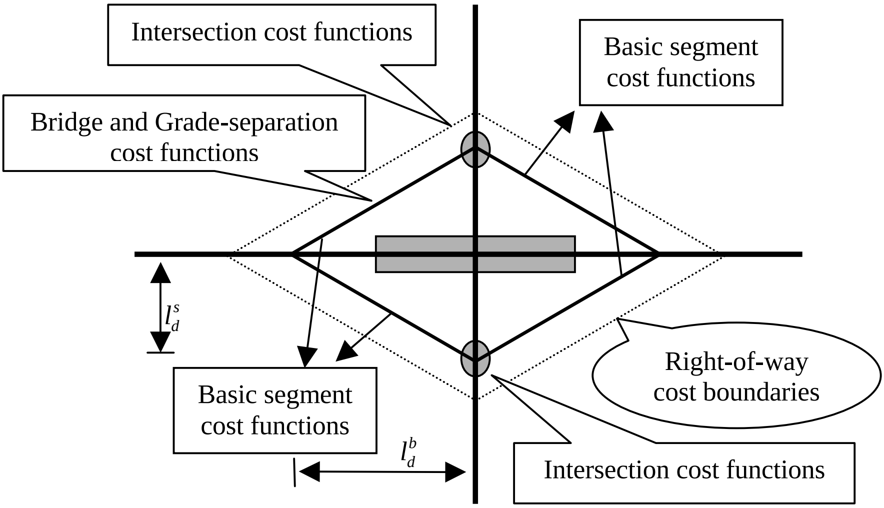

Intersection cost functions
Basic segment cost functions
Bridge and Grade-separation cost functions
l_d^s
Basic segment cost functions
Right-of-way cost boundaries
l_d^b
Intersection cost functions

Figure 12.29: Centerlines of a typical diamond interchange, major design criteria and separated parts for each cost function.

The diamond interchange is divided into several parts to be evaluated using already developed formulations for intersections, basic segments, bridges and grade separations.

Similarly, clover interchange costs can be obtained. Figure 12.30 shows the centerlines of a clover interchange. Three major design criteria are important in evaluating the total costs: (1) interchange leg length, l_c , (2) outside turning ramp radius (r_c^o) and (3) inside turning ramp radius (r_c^i). The total costs can be evaluated by employing the necessary cost functions for each structure.

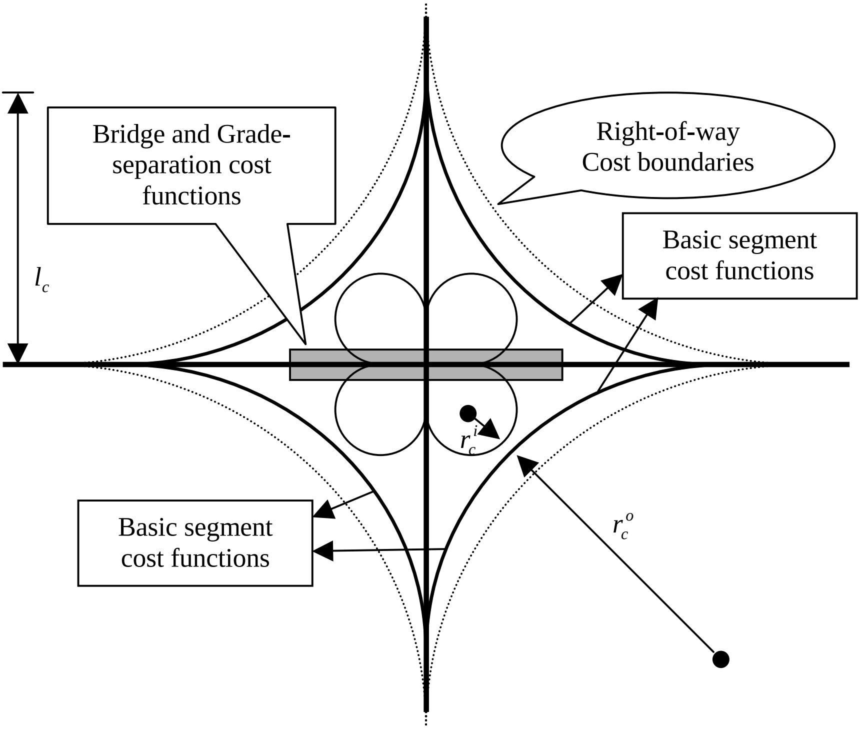

Bridge and Grade-separation cost functions
Right-of-way Cost boundaries
Basic segment cost functions
l_c
r_c^i
Basic segment cost functions
r_c^o

Figure 12.30: Centerlines of a typical clover interchange, major design criteria and separated parts for each cost function.

For trumpet interchanges, the same procedure is also adopted. In it two major design criteria should be considered: (1) outside turning ramp radius (r_t^o) and (2) inside turning ramp radius (r_t^i).

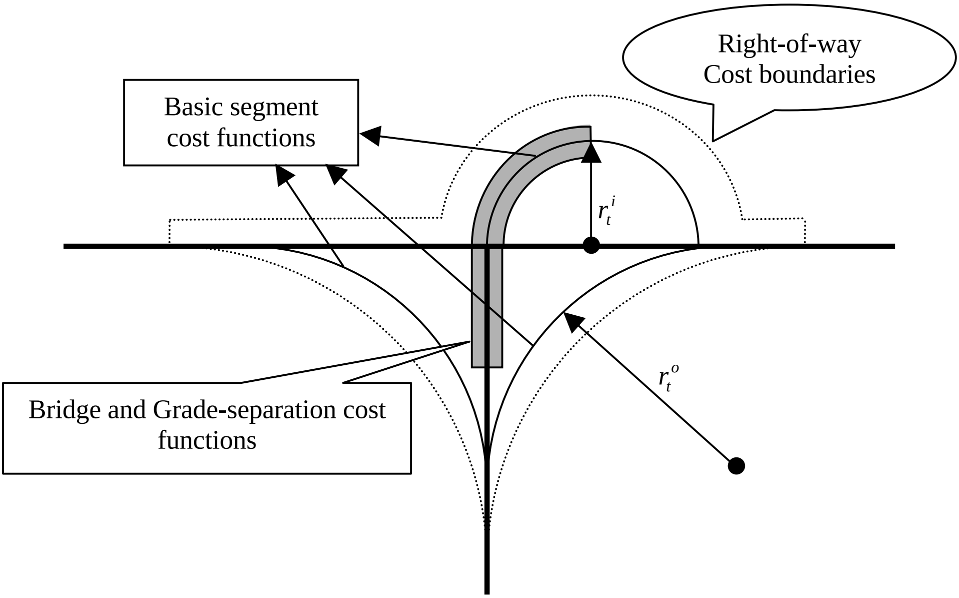


Figure 12.31: Centerlines of a typical trumpet interchange, major design criteria and separated parts for each cost function.

12.6 Cost functions for short tunnels

In section 11.3, the characteristics affecting small tunnel costs are found. They are lengths, cross sections, clearances, horizontal alignments and grades. Also, ventilation for pollutants, lighting for safety, fire life safety provisions for refuge, surveillance and control systems for traffic management and soil types for earthwork are considered as important cost factors. Based on these characteristics and factors, this book formulates (1) earthwork (i.e., excavation) costs and (2) additional costs for small tunnels. Earthwork costs account for lengths, cross sections and clearances, while additional costs account for remaining characteristics and factors. For estimating earthwork costs, typical desirable cross sections and clearances for a two-lane highway tunnel configuration shown in Figure 11.5 are considered. This book refers to two parameters of Figure 11.5, the height (h_T, 4.3 m) and the width of the tunnel (w_T, 13.2 m) and only uses the tunnel radius (r_T, m) since tunnels are usually excavated with circular cross-sections.

Therefore, the tunnel earthwork costs, C_E^T, are

$$C_E^T = K_T L_T \pi (r_T)^2 , \qquad (12.62)$$

where K_T = tunnel earthwork unit cost per cubic meter (\$/m^3)
L_T = tunnel length (m).

The tunnel earthwork cost function, shown in eqn (12.63), is a linear function of tunnel length. However, the other costs for ventilation, lighting, fire safety, surveillance and traffic controls may not be captured with a linear function of tunnel length.

For estimating additional tunnel costs, it is desirable to find functional forms developed from previous studies. Unfortunately, the cost functions for those items are hard to obtain. We may then rely on reliable databases for each affecting factor, from which some functional forms can be developed.

This book does not develop the cost functions for those factors. Here, a quadratic function of tunnel length is simply introduced for preliminary analysis. We can easily envision that some factors such as ventilation, lighting, fire life safety provisions and traffic control systems have a more than linear effect on tunnel costs.

It is important at this point to remember that the optimization processes for highway alignments should involve both dominating and sensitive cost items. For the preliminary alignment optimization, the effects of tunnel excavation costs may dominate the other cost items such as ventilation, fire safety and surveillance. Also, these additional costs for rural two-lane highways might not be sensitive to highway alignments compared to tunnel excavation costs.

Moreover, for normal operations, naturally ventilated and traffic-induced ventilation systems are considered adequate for relatively short tunnels (less than 600 ft (180 m)) with low traffic volumes (Bendelius, 1996). Also, in most cases, a lighting system is not required inside short tunnels (of less than 150 ft) (Mowczan, 1996), and fire safety provisions and traffic control systems may not be deployed for short two-way rural highway tunnels.

In summary, this book assumes that tunnel excavation costs mainly represent tunnel construction costs and additional tunnel costs in eqn (12.63) are only applied where needed:

$$C_a^T = \alpha_1^T \left(L^T\right)^2 + \alpha_2^T \left(L^T\right) + \alpha_3^T , \qquad (12.63)$$

where C_a^T = additional tunnel costs (\$).

This page intentionally left blank

Chapter 13

Incorporating the developed cost functions for intersections and road structures into genetic algorithms

This chapter describes several methods for incorporating the cost functions developed in chapter 12 into genetic algorithms. Section 13.1 develops an algorithm for obtaining ground elevations using planar interpolation to precisely estimate earthwork costs.

Methods for incorporating the cost functions of bridges and tunnels in genetic algorithms are developed in section 13.2. In section 13.3, several methods are discussed for combining intersections, grade separations and interchanges with genetic algorithms. With these developed methods, intersections and other structures can be sufficiently modeled for highway alignment optimization processes.

13.1 Algorithm for obtaining ground elevations using planar interpolation

In section 12.1 two methods, planar interpolation and proportionally weighted interpolation were developed to precisely obtain road ground elevations. This section develops an algorithm for computerizing those methods. Since planar interpolation can be applied instead of the proportionally weighted interpolation to facilitate automation and consistency, this section only discusses the planar interpolation at the station points of new alignments. To apply the planar interpolation, three survey points are needed. Figure 13.1 shows a concept for general cases. Whenever a new alignment crosses one cell (cell 1) and the station point lies in it, this algorithm chooses two adjacent cells as survey points: the next right cell (cell 2) and the cell below (cell 3). It should be remembered that three survey points have three dimensional (3D) coordinates. The coordinates represent the center of each cell. The ground elevation throughout each cell is assumed to be uniform.

There could be some extreme cases where the general case is not applicable. Figure 13.2 shows how we can select two other adjacent cells for those extreme cases.

After obtaining these cells' three dimensional coordinates, the algorithm 13.1 below can be used to find a precise ground elevation for a particular road station point.

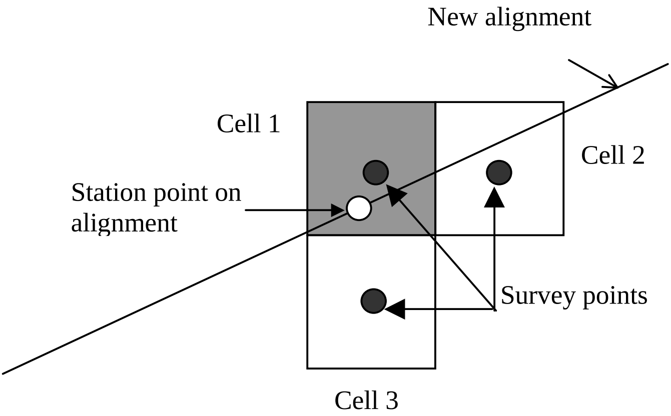


Figure 13.1: A general case for obtaining ground elevations.

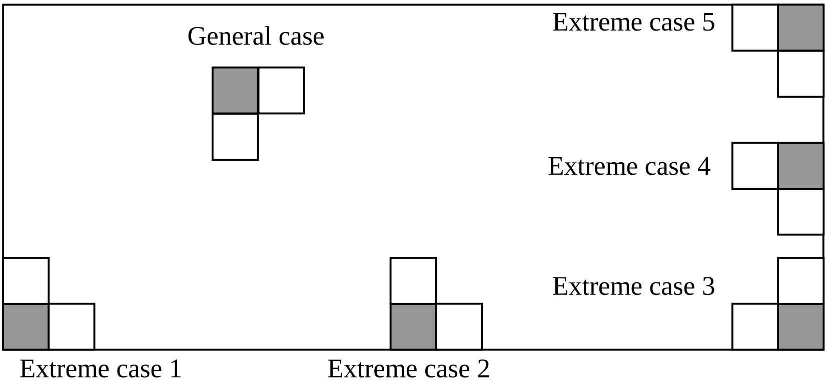


Figure 13.2: Extreme cases when choosing two adjacent cells for planar Interpolation.

Algorithm 13.1 : obtaining ground elevations using planar interpolation

Step 1: Place as many stations on a new alignment as possible. (In this work, 20 ft stations are used.)

Step 2: Obtain three ground elevation points according to a road station's location using the ways developed in Figures 13.1 and 13.2.

Step 3: Find *x*, *y* and *z* coordinates of the obtained three points for each station.

Step 4: Construct two intermediate vectors using three points.

Step 5: Obtain a normal vector using vector product of two intermediate vectors as described in eqn (12.14).

Step 6: Drive a plane function using eqn (12.15).

Step 7: Insert the *x* and *y* coordinates of a road station point into the plane equation and obtain a final *z* coordinate.

As discussed in section 12.1, the precision of earthwork cost estimation can be improved either through better estimation of cross sections or by using shorter stations. Both approaches are tried in this work. The new algorithm's efficiency and reliability were already discussed at the example study in section 12.1.4.

13.2 Algorithm for combining functions of bridges and tunnels

Cost functions of bridges and tunnels were developed in sections 12.3 and 12.6, respectively. To incorporate those cost functions, locating many stations on a new alignment is desirable. Overly long stations are not suitable for precisely determining the lengths and costs of bridges and tunnels. 20 ft stations are used in this book. We must also determine critical elevation differences for bridge (tunnel) construction rather than fills (cuts). Finding the critical elevation difference is a function of the associated earthwork volumes as well as the lengths of bridges and tunnels. Thus it is an implicit process. That is, optimal elevation differences cannot be predetermined before the lengths are obtained.

However, vertical clearances should be maintained for construction of bridges and tunnels and safety of the drivers. This book considers the vertical clearance of highways as a minimum elevation difference. 15 ft is assumed for this purpose. Algorithms for combining bridges and tunnels with original genetic algorithms are developed as follows:

Algorithm 13.2: for combining bridges and tunnels cost functions

(1) Step 1: Set as many stations on a new alignment as computationally feasible. In this study we use 20 ft (6 meter) intervals for stations.

(2) Step 2: Find the stations at which bridges and tunnels are preferable. If elevation differences exceed vertical clearances (15 ft, (4.5 m)), save those stations as candidates for bridge or tunnel construction, depending on the sign of elevation difference.

2.1: Discard any alignment through creeks, rivers and lakes that violates the high water level constraint, ε_w. Thus, if any station's road elevation is not above ε_w ($z_r \le \varepsilon_w$) , the alignment is infeasible. Also, assign large right-of-way costs to any alignment through environmentally sensitive properties.

2.2: Find stations for bridges and tunnels.

2.2.1: Find stations for bridges: If (z_g > 0 and z_r - z_g > 4.5 m) or (z_g = 0 or z_r - z_g > 0), save (z_r - z_g) and the station number for bridge construction.

2.2.2: Find stations for bridges: If (z_g > 0 and z_g - z_r > 4.5 m), save (z_g - z_r) and the station number for tunnel construction.

2.3: Obtain the lengths of bridges and tunnels using the saved station number in the step 2.2. This step only applies when the lengths equal or exceed the user-specified length which, in this book, is set at 100 ft.

2.4: Estimate the costs of bridges and tunnels

2.4.1: Bridge costs

(1) Find an optimal span length using the method developed in the previous section.

(2) Estimate superstructure costs and substructure costs based on the optimal bridge span and lengths using the cost functions.

(3) To adjust for inflation, use a gross domestic product (GDP) deflator which can be obtained from NASA's cost models web site or from the Bureau of Economic Analysis, to convert the coefficients of eqns (12.57) and (12.58), which are based on 1969 dollars, into 2001 dollars.

2.4.2: Calculate tunnel costs using eqns (12.63) and (12.64).

2.5: Calculate earthwork costs for stations where bridges and tunnels are not considered. For precisely estimating earthwork costs, planar interpolation is applied.

(3) Step 3: Return the total costs.

3.1: If no bridges and tunnels are considered, return pure earthwork costs.

3.2: If a new alignment hits environmentally sensitive areas, return the assigned earthwork costs using a large unit cost in step 2.1.

3.3: If a new alignment consists of at least one bridge or tunnel, return the bridge and tunnel costs summed up with earthwork costs for stations where cuts and fills are applied.

13.3 Algorithms for incorporating intersections, grade separations and interchanges

This section consists of four parts: (1) a data format for saving the coordinates of the existing roads, (2) methods for selecting the crossing type among intersections, grade separations (overpasses or underpasses) and interchanges, (3) methods for determining a signal type for intersections and an optimal signal timing for signalized intersections and (4) methods for incorporating the developed cost functions into genetic algorithms. These algorithms and methods are needed whenever a new alignment crosses the existing roads.

13.3.1 Data format for saving the coordinates of the existing roads

The existing roads should be saved in ways that are easy to handle and access for computerization. Ideally, the existing road's information (coordinates) can be defined as a functional form. The developed method for finding the crossing point in section 12.2.1 then can be used. However, that is not the case in normal conditions. It is assumed in this book that the coordinates of an existing road are obtained from the road construction databases and a piecewise linear data format is used to save and extract the existing road information. Suppose that Figure 13.3(a) depicts an existing road on a study area. The existing road then can be converted into several piecewise linear segments, as shown in Figure 13.3(b).

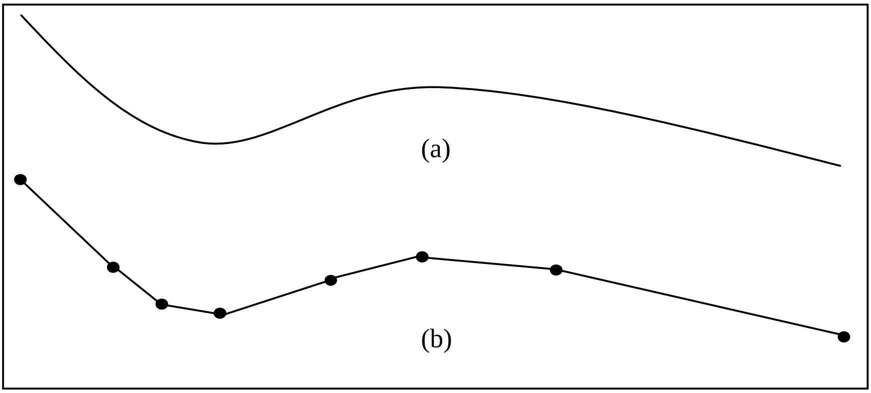


Figure 13.3: Converting the existing road into piecewise linear segments.

13.3.2 Methods for selecting the crossing type among intersections, grade separations and interchanges

Where a crossing occurs, a crossing type should be pre-selected by evaluating the total costs of alternatives. The possible alternatives include intersections, grade separations (overpasses or underpasses) and interchanges. This selection process is somewhat related to political decisions and public preferences that cannot easily be explained by some functional forms. In most cases, interchanges cost more than the other types since they normally occupy more land and properties. However, in some extreme cases, especially in rural areas, when vertical clearances are excessive, intersections may cost more than a simple diamond interchange. In this book, all three types of structures are evaluated for each crossing location and the type with the minimum cost is selected as a crossing structure. (Detailed procedures for estimating the costs are addressed later in algorithms 13.3, 13.4, and 13.5.)

A new alignment can meet several types of existing roads. The existing road could be a freeway, a highway or a collector. In this book, eight-lane freeways and two-lane rural highways are considered, but other types can be modeled in similar ways.

13.3.3 Methods for determining a signal type for interchanges and an optimal cycle for signalized intersections

Estimating the cost of an intersection requires determining the signal type for that intersection. This section develops a method to pre-select the signal type in a way that sufficiently supports intersection cost estimation. Three references are available to determine whether signal installation at intersections is warranted: the 2000 Manual on Uniform Traffic Devices (FHWA), the 1988 Manual on Uniform Traffic Devices (FHWA), and Manual of Traffic Signal Design (MTSD) by Institute of Transportation Engineers (ITE, 1991). The 1988 MUTCD lists 11 warrants: (1) minimum vehicle volume, (2) interruption of continuous traffic, (3) minimum pedestrian volume, (4) school crossings, (5) progressive movement, (6) accident experience, (7) systems, (8) combination of warrants, (9) four hour volumes, (10) peak hour delay and (11) peak hour volume. The updated 2000 MUTCD reduces those 11 warrants to 8: (1) eight-hour vehicular volume, (2) four-hour vehicular volume, (3) peak hour, (4) pedestrian volume, (5) school crossings, (6) coordinated signal system, (7) accident experience and (8) roadway network.

The MTSD (ITE, 1991) suggests verifying the requirements of the warrants in the following order based on the 1988 MUTCD warrants (McDonald Jr., 2001):

(1) Warrants 1, 2, 8, 9 and 11 if the available volume data is available;
(2) Warrant 6 after collecting accident data;
(3) Warrant 3 after collecting pedestrian data;
(4) Warrant 8 (combination warrant);
(5) Warrant 4 after collecting specialized school data; and then
(6) Warrants 5 and 7 (controlling arterial and system flow).

In this book, warrant 1 of the MUTCD 1988 is simply employed. Table 13.1 shows how it can be applied.

If some signalization is warranted for an intersection, the signal cycle should be optimized in order to estimate delay costs later. Eqn (13.1) developed by Webster (McShane *et al*, 1990) is used to optimize the cycle.

$$C_o = \frac{1.5L + 5}{1 - \sum (v/s)_c}, \qquad (13.1)$$

where C_o = optimal cycle for minimum design speed, sec
L = lost time per cycle, sec
v = traffic volumes, veh
s = saturation flow rate, veh
$\sum(v/s)_c$ = sum of v/s ratios for critical lanes or critical lane groups

Table 13.1: Minimum vehicular volumes for warrant 1.

Number of lanes for moving traffic on each approach		**Vehicles per hour on major street (total of both approaches)**	**Vehicles per hour on higher-volume minor-street approach (one direction only)**
Major street	**Minor street**		
1	1	500	150
2 or more	1	600	150
2 or more	2 or more	600	200
1	2 or more	500	200

Algorithm 13.3: Incorporating the developed cost functions for grade separations into genetic algorithms

(1) Step 1: Check if vertical clearances (15 ft in this study) are satisfied. (Apply the same rule for both underpasses and overpasses by changing sign of elevation differences.)

(2) Step 2: If vertical clearances are satisfied, check the existing road type.

2.1: (1) If the existing road is a freeway (eight lanes for both directions), use Figure 13.4 as a template for estimating the total costs of grade separations. Three pier locations, on the center line and on the both shoulders, are set.

(2) Estimate grade separation costs using bridge cost functions and algorithms developed in section 13.2. (Estimate substructure and superstructure costs.)

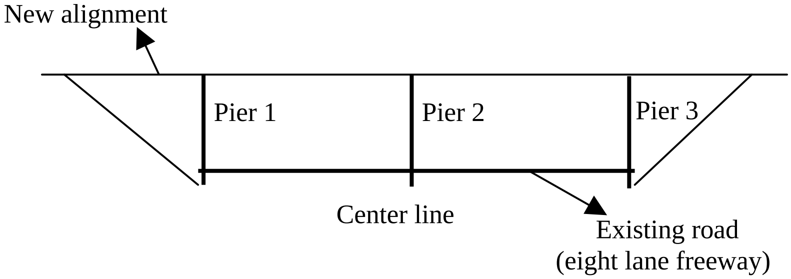


Figure 13.4: An example freeway overpass with three piers.

2.2: (1) If the existing road is a two-lane rural highway, use Figure 13.5 as a template for estimating the total costs of grade separations. Only one pier, on the center line of the existing road, is considered.

(2) Estimate grade separation costs using bridge cost functions and algorithms developed in section 13.2.

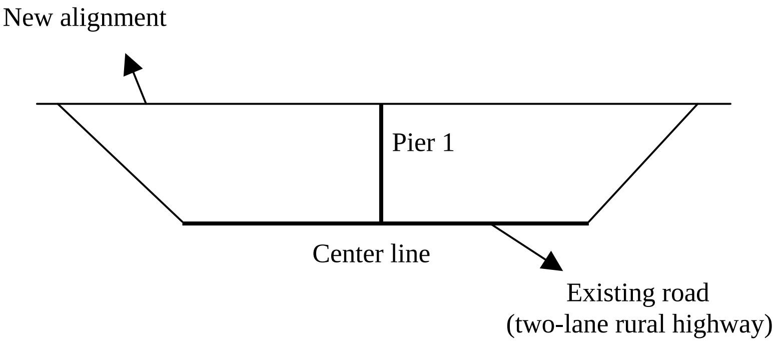


Figure 13.5: A grade separation template for a two-lane rural highway.

(3) Step 3: If vertical clearances are not satisfied, (1) Estimate additional earthwork costs for providing vertical clearances using eqn (12.61). These costs are treated as penalty costs. (2) Estimate grade separation costs using the same procedures described in step 2 above. (3) Add two cost items and return it.

Algorithm 13.4: Incorporating the developed cost functions for intersections into genetic algorithms

(1) Step 1: Estimate pavement costs based on the design standards using eqn (12.27).

(2) Step 2: Estimate right-of-way costs

2.1: Set boundaries using the method developed in section 12.2.3. (Find four distinctive points of four legs from the crossing points by the design standards.)

2.2: Estimate right-of-way costs using the methods described in section 12.2.4.

2.3: Subtract the double counted right-of-way costs using boundary information obtained in step 2.1. This means subtracting the alignment right-of-way costs that are duplicated.

(3) Step 3: Estimate earthwork costs

3.1: Set boundaries using the developed method in section 12.2.3.

```
            3.2:  Estimate earthwork costs using the methods developed in section 12.2.3.
            3.3:  Subtract the double counted earthwork costs.
(4)  Step 4:  Estimate intersection delay costs
            4.1:  Adopt the determined control type in section 13.3.3.
            4.2:  Estimate intersection delay costs
                  4.2.1:  For signalized intersections, (1) assume a simple two-phase signal installation and  (2) estimate delays using eqn (12.50).
                  4.2.2:  For unsignalized intersections, estimate delays using eqn (12. 54).
(5)  Step 5:  Estimate intersection fuel costs (for both control types)
            5.1:  Obtain percentages of stops using Figure 12.22 according to delays found in step 4.2.
            5.2:  Estimate fuel costs due to delays and stops using Table 12.5.
(6)  Step 6:  Estimate intersection accident costs
            6.1:  For signalized intersections, estimate the costs using eqn (12.47) and Figure 12.21.
            6.2:  For unsignalized intersections, estimate the costs using eqns (12.48) or (12.49), depending on the number of intersection legs.
```

Algorithm 13.5: Incorporating the developed cost functions for interchanges into genetic algorithms

```
(1)  Step 1:  Obtain a user specified interchange type from the input file. (This study does not search for an optimal interchange type.)
(2)  Step 2:  For diamond interchanges:
            2.1:  Find four points on the interchange legs as in Figure 12.29.
            2.2:  Estimate pavement costs based on the design standards.
            2.3:  Estimate right-of-way costs and then subtract the double counted right-of-way costs.
            2.4:  Estimate earthwork costs using Figure 12.29, and then subtract the double counted earthwork costs.
            2.5:  Estimate the associated grade separation costs using ways developed in section 13.3.1.
            2.6:  Estimate two new intersection costs using the steps 4, 5 and 6 in algorithm 13.4 (Unsignalized control type is assumed for two intersections.)
(3)  Step 3: For clover and trumpet interchanges, apply same procedures for diamond interchanges using concepts illustrated in Figures 12.30 and 12.31, respectively, without considering new intersections.
```

This page intentionally left blank

Chapter 14

Local optimization of intersections for highway alignment optimization

This chapter develops a method to locally optimize intersections while changing the existing road as necessary. Section 14.1 explains the motivation for this chapter. Methods for local optimization of intersections are developed in section 14.2 followed by formulation of the objective function in section 14.3. An algorithm for incorporating the developed local intersection optimization methods into alignment optimization processes is also presented at the end of section 14.3. Section 14.4 shows examples using an artificial study area and a real GIS map. The examples show how the developed methods work for both local intersection optimization and for entire highway alignment optimization processes.

14.1 Motivation for local intersection optimization

When an intersection is considered as a crossing type between a new alignment and an existing road, the crossing angle is a very important constraint to satisfy. Figure 14.1 shows one example where a new highway alignment is flawed, since the crossing angle is excessive.

It may be, however, that the new alignment shown in Figure 14.1 is superior to other alternatives and that discarding it simply because of the intersection angle might be undesirable overall. It seems preferable to pursue a method that could perturb the local geometry to produce a better intersection, yet retain the broader geometry of the good candidate alignment. Figure 14.2 shows an example of how a better solution might be obtained. In Figure 14.2, the existing road is assumed to be a minor road.

To date, efforts at alignment optimization fall largely into two categories. A number of studies focused on the problem of determining the best alignment between two fixed points, using some objective function and constraints. These studies assumed that the new alignment would not cross any existing highways, or at least that such crossings need not to be considered. The other class of problems dealt with refining the local geometry of intersections (Barnett, 1939; Easa, 1998; Taber, 1998; Harwood *et al*, 2000), but not in the context of a larger alignment optimization problem.

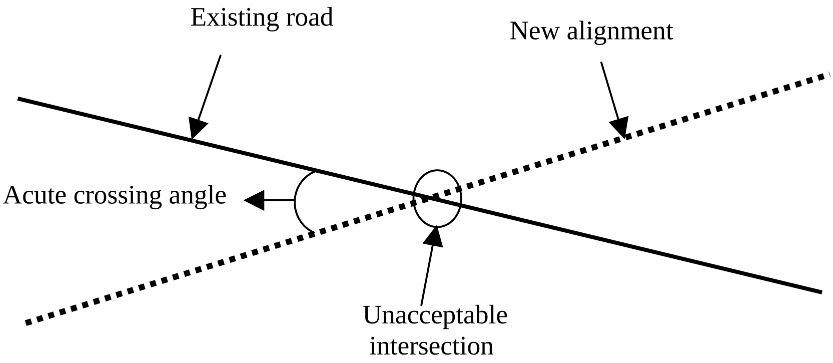


Figure 14.1: Example of intersection with unacceptably acute angle.

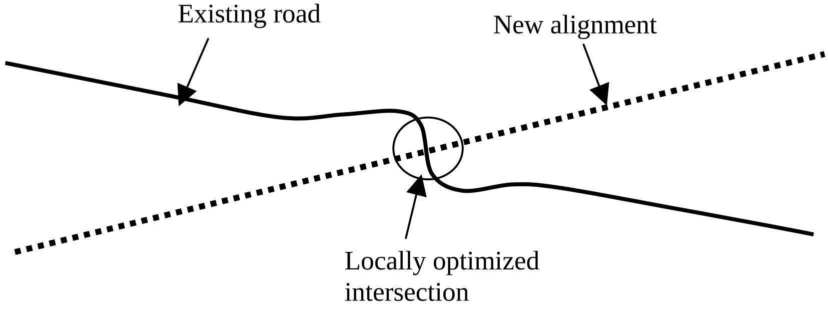


Figure 14.2: A locally optimized intersection for Figure 14.1.

A model that enables local intersection optimization can enhance flexibility in searching for highway alignments, yielding more effective intersections. Most importantly it can provide a basis for highway network optimization.

14.2 Methods for local intersection optimization

Intersections can vary greatly based on the number of legs, degree of channelization, control types and locations. Figure 14.3 shows realignment variations at intersections where overly acute crossings would otherwise occur. AASHTO suggests that the crossing angle should be in the range of approximately 60 to 120 degrees.

At the same time, however AASHTO also recommends that "intersecting roads should generally meet at or nearly at right angles." In this book, we focus on cases such as A or B in Figure 14.3, and assume that right-angle crossings can be constructed. Based on this assumption, Figure 14.4 shows some of the alternatives for a realigned intersection.

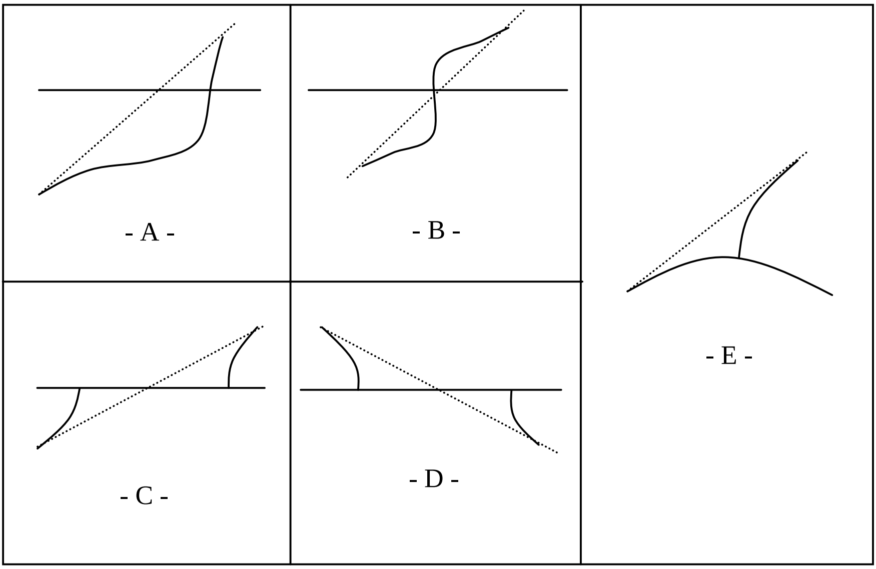


Figure 14.3: Realignment variations at intersections (AASHTO).

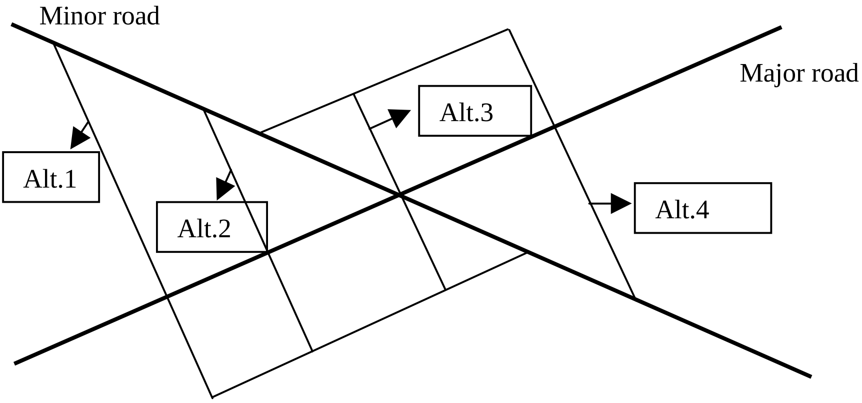


Figure 14.4: Some alternatives for local intersection optimization.

It is assumed that the local optimization process described herein resides within a larger alignment optimization framework. For local optimization to take place, it must be the case that an alignment alternative has been generated that crosses an existing road at an unacceptable angle, θ, as described earlier. The existing roadway presumably is described in a database (for example, GIS), and the most common form would be piecewise linear, with points $\{\mathbf{E}_i\}$ representing the linear segment endpoints. The proposed new alignment can be described similarly, although we adopt a form more common in highway design, consisting of a sequence of tangent sections and circular arcs. We assume that station points, $\{\mathbf{D}_i\}$ are defined along this alignment at regular intervals specified by the user.

The collection of station points in the vicinity of the proposed intersection constitutes the domain of our decision variable, $\{\mathbf{D}_i\}$, which is the location of the newly aligned intersection. The question of what constitutes the "vicinity" is up to the model user. The point, $\mathbf{I}$, is the hypothetical intersection of the existing and new roadways. It is determined exogenously by methods such as in (Lovell 1999). On either side of the proposed intersection, $\mathbf{I}$, we consider at least one of the existing roadway nodes, $\{\mathbf{E}_i\}$. These need not fall within our vicinity. However, if several of them happen to do so, then they all must be considered. This is described in an example later in this book.

The decision variable $\mathbf{D}$ represents the potential location of the intersection. If discrete optimization is being used, then the domain of $\mathbf{D}$ could be the set of station points $\{\mathbf{D}_i\}$ described earlier; otherwise, it must be constrained to fall along the alignment that they describe. Considering Figure 14.5, suppose that a new alignment has been developed, and represents the major road at the intersection. The existing minor road, therefore, will need to be perturbed.

The user can specify a minimum leg length for the new intersection. This is the distance upstream of the intersection over which the roadway is required to be straight, and is shown as a_1. For a given minimum distance a_1, the distance a_2 has a minimum value as well. The points $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ and $\mathbf{P}_4$ are the corners of the rough geometry of the perturbed alignment. For intersections with minimum leg lengths, these points can be obtained using $d_1 = d_2 = a_1 / \cos(90 - \theta)$ and the following relations:

$$\mathbf{P}_1 = \mathbf{D} + \left(\frac{\mathbf{D} - \mathbf{E}_2}{\|\mathbf{D} - \mathbf{E}_2\|} \right)(-d_1) \text{ and } \mathbf{P}_4 = \mathbf{D} + \left(\frac{\mathbf{D} - \mathbf{E}_2}{\|\mathbf{D} - \mathbf{E}_2\|} \right)(d_2), \tag{14.1}$$

$$\mathbf{P}_2 = \mathbf{P}_1 + \left(\frac{\mathbf{D} - \mathbf{S}}{\|\mathbf{D} - \mathbf{S}\|} \right)(a_2) \text{ and } \mathbf{P}_3 = \mathbf{P}_4 + \left(\frac{\mathbf{D} - \mathbf{S}}{\|\mathbf{D} - \mathbf{S}\|} \right)(a_2), \tag{14.2}$$

where $\mathbf{S}$ = one of the adjacent station points.

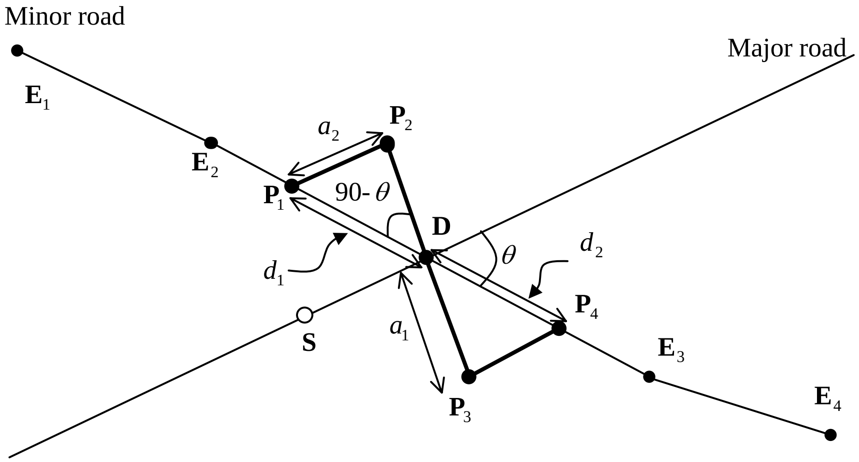


Figure 14.5: Alternative with minimum leg length.

Figure 14.6 shows a more general alternative where intersection legs are longer. This type of alternative generally costs more than the alternative in Figure 14.5. However, it may avoid environmentally and socially sensitive areas and thus be justified. More steps are needed to obtain the coordinates of the alternative in Figure 14.6. Since $\mathbf{P}_1$ is between $\mathbf{E}_1$ and $\mathbf{E}_2$, we need to know θ'. It can be obtained as follows:

$$\theta' = \cos^{-1}\left(\frac{(\mathbf{E}_2 - \mathbf{E}_1)\cdot(\mathbf{I}-\mathbf{D})}{\|\mathbf{E}_2 - \mathbf{E}_1\|\|\mathbf{I}-\mathbf{D}\|}\right). \tag{14.3}$$

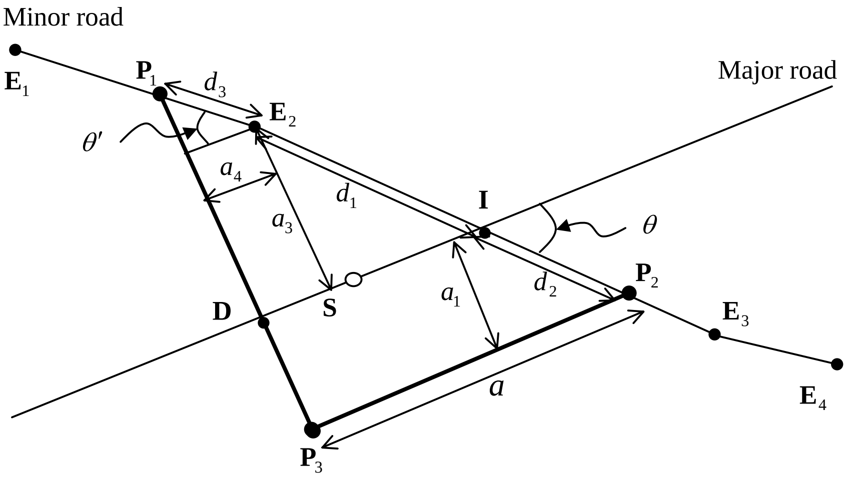


Figure 14.6: Interesting points for a general alternative.

Then, using $a_3 = d_1 \sin\theta$, $a_4 = \|\mathbf{I}-\mathbf{D}\| - \sqrt{d_1^2 - a_3^2}$, $d_3 = a_4 / \cos\theta'$ and $d_2 = a_1 / \cos(90-\theta)$, all the remaining coordinates can be obtained as follows:

$$\mathbf{P}_1 = \mathbf{E}_2 + \left(\frac{\mathbf{E}_2 - \mathbf{E}_1}{\|\mathbf{E}_2 - \mathbf{E}_1\|}\right)(-d_3), \tag{14.4}$$

$$\mathbf{P}_2 = \mathbf{I} + \left(\frac{\mathbf{I} - \mathbf{E}_2}{\|\mathbf{I} - \mathbf{E}_2\|}\right)(d_2), \tag{14.5}$$

$$\mathbf{P}_3 = \mathbf{D} + \left(\frac{\mathbf{D} - \mathbf{P}_1}{\|\mathbf{D} - \mathbf{P}_1\|}\right)(a_1). \tag{14.6}$$

Based on these coordinates, any point on the newly evaluated intersection legs can be obtained. This helps us formulate each cost item by easily identifying where the legs and the crossing point (intersection) are located within a study area.

14.3 Formulation of the objective function for local intersection optimization

The objective function consists of several cost items. Among many cost items, it is important to include dominating costs that are also sensitive to alignments. Since the problem analyzes highway alignments and intersections, cost items for both need to be checked. Among the items previously discussed, this book considers in the model formulation right-of-way, earthwork and pavement costs for intersection legs (that is, highway alignment parts) and additional right-of-way and earthwork costs for locally optimized intersections.

It should be remembered that all five cost items can be estimated using methods already developed in previous chapters and sections. Intersection operational costs such as delay and fuel costs are excluded since those cost estimates do not seem to greatly change by configurations of locally optimized intersections. Based on these findings, the objective function can be formulated. It is the summation of five sensitive cost items, as shown below.

$$\text{Minimize} \quad C_T = C_R^L + C_E^L + C_P^L + C_R^I + C_E^I , \tag{14.7}$$

where C_T = total costs ($)
C_R^L = right-of-way costs of intersection legs ($)
C_E^L = earthwork costs of intersection legs ($)
C_P^L = pavement costs of intersection legs ($)
C_R^I = additional right-of-way costs of intersections ($)
C_E^I = additional earthwork costs of intersections ($).

Since local intersection optimization occurs within highway alignment processes, we must devise an algorithm for effectively incorporating the developed methods within genetic algorithms. Algorithm 14.1 specifies how local intersection optimization is initiated and works within alignment optimization.

Algorithm 14.1: Incorporating local intersection optimization within alignment optimization

At section 13.3.2 in chapter 13, a method for selecting the minimum cost crossing type among intersections, grade separations and interchanges was developed assuming the existing road is unchangeable. There are two possible cases where local intersection optimization is applicable. In one, the selected crossing type is an intersection with an overly acute angle and in the other, a user specifies an intersection as a structure type. The following steps specify how local intersection optimization is incorporated into alignment optimization.

(1) Step 1: Check if there is a crossing point between a new alignment and existing roads.

- 1.1: If there is no crossing point, stop.
- 1.2: If there is a crossing point, go to step 2.

(2) Step 2: Check if a user specifies intersection construction.

- 2.1: If a user specified intersection construction, check if the crossing angle exceeds 60 degrees.
  - 2.1.1: If the crossing angle exceeds 60 degrees, adopt algorithm 13.4 for evaluating intersection costs.
  - 2.1.2: If the crossing angle is less than 60 degrees, apply developed local intersection optimization method.
- 2.2: If a user did not specify intersection construction, check if the selected structure type at section 13.3.2 is an intersection.
  - 2.2.1: If the crossing angle exceeds 60 degrees, (1) Use an intersection cost estimate which was already obtained using algorithm 13.3 to compare with the costs for other structure types (grade separations and interchanges). (2) Select the minimum cost structure type.
  - 2.2.2: If the crossing angle is less than 60 degrees, (1) Apply developed local intersection optimization method. (2) Compare with the costs for other structure types (grade separations and interchanges). (3) Select the minimum cost structure type.

14.4 Example study

Two example studies are presented to show performance of the proposed method. One is based on an artificial study area that is designed to check if the developed methods work properly. The other is based on a real GIS map to see if a real application is executable and the result is reasonable

14.4.1 Example study based on an artificial area

In Figure 14.7, a darker cell means a higher elevation. The cross-patterned areas of the map represent inaccessible or environmentally untouchable regions, through which no new alignment is allowed. To check how the local intersection optimization performs in this artificial study area, just 30 generations were run. The results in Figure 14.7 demonstrate the usefulness of the developed method, since the alignment in Figure 14.7 might have been discarded without local intersection optimization due to its unacceptable crossing angle of approximately 58 degrees. (The acceptable range is between 60 to 120 degrees.) After that, 500 generations were run and the final solution was obtained, as shown in Figure 14.8. The figure shows three main window areas: (1) horizontal alignment, (2) vertical alignment and (3) generation number and best solution value.

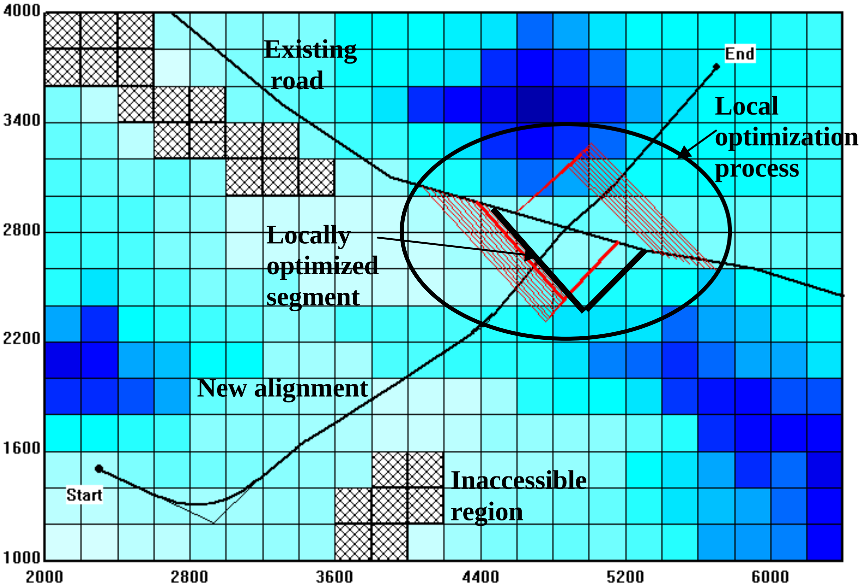
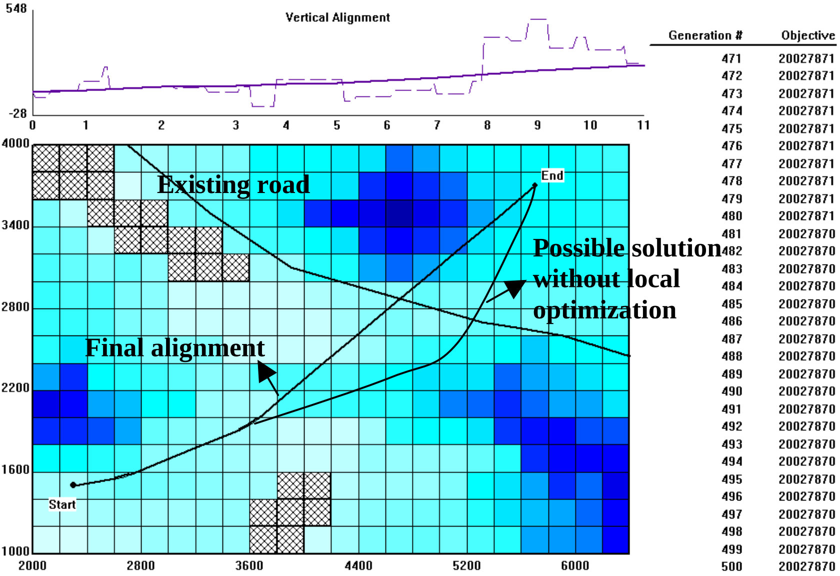


Figure 14.7: Artificial study area after 30 generations.

As we can see in Figure 14.8, the optimized solution might have been discarded since its crossing angle with the existing road is not still acceptable. However, the developed local intersection optimization method kept this solution through the end of 500 generations. The total costs of the new alignment are about $20 million. We can easily imagine that without the developed method the final solution would be different. The possible solution in Figure 14.8 looks better than the final solution. However, it should be noted that the objective function for alignments include user costs that normally account for 70-80% of total alignment costs. Therefore a longer alignment costs more for fuel and travel time even if it costs less for construction.

14.4.2 Example study based on a real GIS map

This section demonstrates the application of the local optimization procedure when a real GIS map is used. The connection of GIS with genetic algorithms has been established by using specialized dynamic link libraries. The local optimization algorithm is embedded into genetic algorithms (written in C and Visual C++ languages). An Arc-View GIS based algorithm using Avenue language is written. The study section is taken from Baltimore County, Maryland. It contains 90 properties and most of them are residential, a creek, and several arterial streets. Its area is about 0.47 mi^2 (1.22 km^2). Unit land costs range from \$0.02/ft^2 (\$0.22/m^2) to \$1.72/ft^2 (\$18.51/m^2) and costs of structure range from $10,900 to $535,560.



Generation #	Objective
471	20027871
472	20027871
473	20027871
474	20027871
475	20027871
476	20027871
477	20027871
478	20027871
479	20027871
480	20027871
481	20027870
482	20027870
483	20027870
484	20027870
485	20027870
486	20027870
487	20027870
488	20027870
489	20027870
490	20027870
491	20027870
492	20027870
493	20027870
494	20027870
495	20027870
496	20027870
497	20027870
498	20027870
499	20027870
500	20027870

Figure 14.8: Solution in an artificial study area with 500 generations.

It is desired to build a new two-lane road connecting Rayville Rd. and Middletown Rd. at specified points. The Euclidean distance between the end points is about 1000 feet (306 meters). The lane and shoulder widths of the proposed alignment are assumed to be about 12 feet (3.7 m) and 3.3 feet (1 m), respectively. The design speed is assumed to be 65 mph (105kmph). It is noted that genetic algorithms which are used for optimal search first generate a set of random points that are passed to the GIS to construct the alignment. When an existing arterial is intersected, additional points are generated to perform local optimization for the design of the resulting intersection. Spatial analysis is used to compute the net right-of-way cost (for the entire alignment including the intersection design) due to the overlap between existing properties and the proposed design. This cost is passed to genetic algorithms where other costs that are part of the objective function are computed. 100 generations are run for this particular example, which required 5 hours and 3 minutes on a desktop computer with 1 GHz CPU speed and 261 MB RAM.

A particular alternative that initiates local optimization of intersections at the 16th generation is shown in Figure 14.9. The background map shows the land-cost map, with darker shades representing more expensive land parcels. The local optimization site can be seen at an intermediate location when the proposed alignment intersects an existing arterial at approximately 57 degrees. At every generation, several alternatives initiating local intersection optimization are observed through 100 generations. That is the major reason for the much longer computation time.

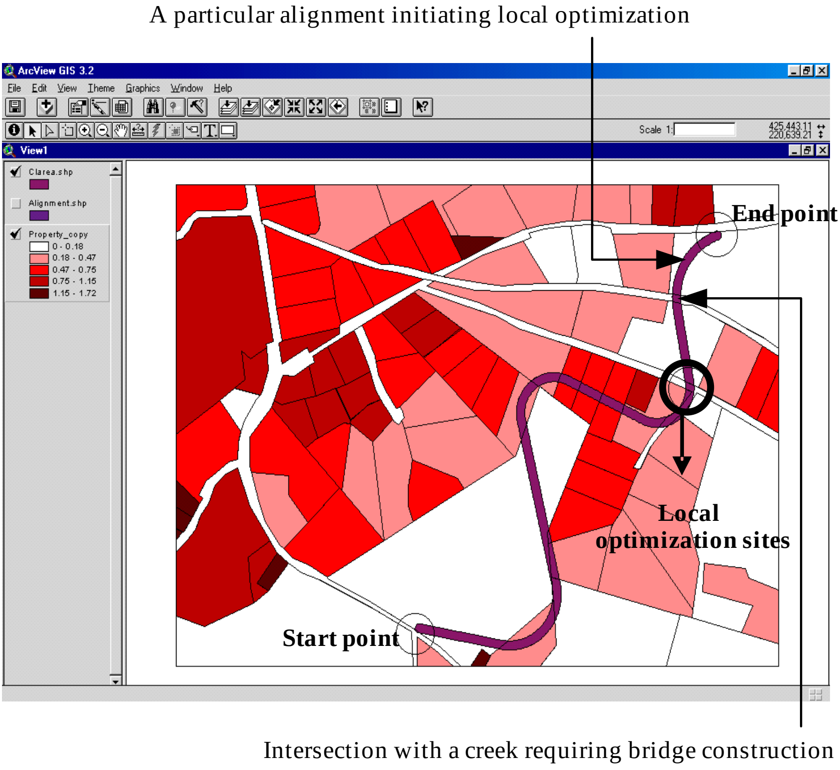


Figure 14.9: A particular alternative initiating local optimization using a real GIS map, Baltimore County, Maryland.

Figure 14.10 shows the optimized alignment obtained after 100 generations. It is very straight, crosses the existing road with 68.2 degrees and, thus, does not involve local intersection optimization.

In conclusion, by adding local intersection optimization to the existing highway alignment optimization, we can avoid wastefully discarding an alignment alternative that crosses an existing road with an overly acute angle but is relatively good otherwise. Moreover, the developed method can produce a more practical alignment and accurate cost estimates while (1) determining the best alignment between two fixed points and (2) refining the local geometry of intersections.

Other examples along with detailed analysis of local intersection optimization and its usefulness for alignment optimization are examined in chapter 15.

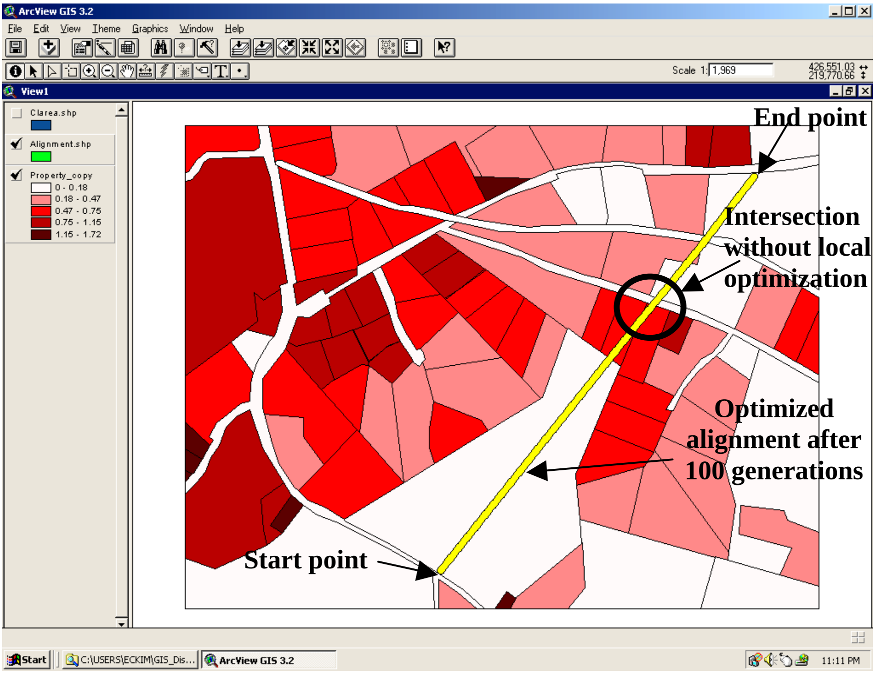


Figure 14.10: Optimized solution found for Baltimore County, Maryland.

Chapter 15

Case studies with intersections and road structures

In this chapter, case studies are conducted for various study areas and scenarios. Either artificial or real GIS maps describe the study areas. Artificial study areas are deliberately designed for checking and verifying the performance of the developed method. Real GIS maps are then selected from GIS databases for the State of Maryland to assess the applicability and reliability of the each developed method. For instance, to check the performance of the modeling method for bridges and tunnels, it is desirable to design or find study areas which consist of lakes (or rivers), hills and mountainous areas along with relatively flat properties. Overall, the employed study areas are complex enough for assessing the applicability of each developed method.

Results obtained after incorporating newly developed algorithms and cost functions for structures are compared to those found in previous studies. Section 15.1 applies the planar interpolation developed in chapter 12 in highway alignment optimization. Changes in the optimized alignments and total costs are discussed. Section 15.2 shows how solutions that may incorporate bridges and tunnels differ from solutions that just consider fills and cuts. Also, a sensitivity analysis for tunnels is conducted based on different unit excavation costs and various elevation differences at which tunnels compete with cuts. In section 15.3 several case studies are conducted based on complex artificial study areas and real GIS maps. All developed cost functions for intersections and other road structures along with the local intersection optimization algorithms developed in chapters 12, 13 and 14 are embedded and implemented for these case studies. In section 15.4, a two-stage optimization is introduced and applied to improve both computational efficiency and quality of results. Finally, sensitivity analysis of important parameters is conducted to check if different starting points in the search space yield similar results at the end of the search. Throughout the chapter, the feasibility and reliability of the developed cost functions and algorithms for structures are assessed.

15.1 Application of planar interpolation for estimating earthwork costs

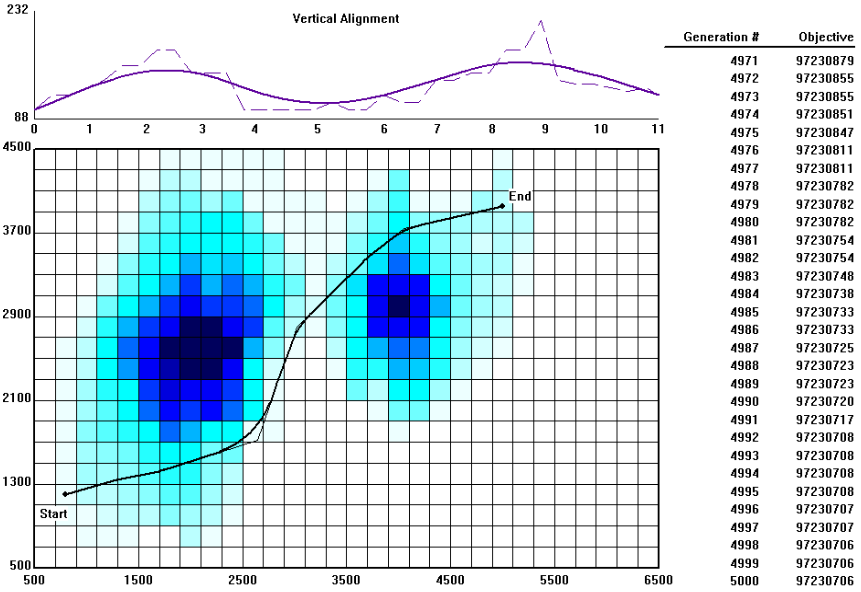
In section 12.1, planar interpolation and proportionally weighted interpolation were developed for better estimating cross section areas through mathematical expressions using vector and parametric representation. However, those were not implemented in the alignment optimization process, although their accuracy was shown to be sufficient for preliminary highway planning purposes after manual testing on a simple test area. Thus, their applicability has not been verified within the context of alignment optimization.

This section employs the planar interpolation on an artificial study area and a real map using GIS. Figure 15.1 shows an artificial study area in which darker cells represent higher elevations. It is notable that the study area is quite mountainous and should benefit from analysis with planar interpolation. For comparison, five replications are run for the algorithm without planar interpolation and for the new algorithm incorporating planar interpolation, using the common random number method (Law *et al*, 1991). 5000 generations are run, requiring approximately 1 minute and 30 seconds for each case. The locations of solutions obtained from two approaches are found to be very close (see Figures 15.1 and 15.2) and their lengths are approximately 1.3 miles long. However, the total costs show differences. It can be seen from Table 15.1 that the differences resulted mainly from the earthwork costs, for which estimates incorporating planar interpolation are 1.8 to 3.3 times larger.

Table 15.1: Cost comparisons for two approaches.

Replications	**Seed**	**Computation time (seconds)**		**Total costs (million)**		**Earthwork costs (million)**	
		Without planar Interpolation	**With planar Interpolation**	**Without planar interpolation**	**With planar Interpolation**	**Without planar Interpolation**	**With planar Interpolation**
1	20695	83	85	97.592	102.204	4.590	8.496
2	21067	83	88	98.141	105.300	3.177	6.163
3	21166	88	89	98.232	102.342	1.684	5.497
4	21509	84	85	98.399	102.505	3.674	8.714
5	21781	85	88	97.231	103.745	2.349	7.547
Mean				97.919	103.219	3.095	7.283
Min.				97.231	102.204	1.684	5.497
Max.				98.399	105.300	4.590	8.714

It is also observed that the differences between the maximum and minimum earthwork costs from the two approaches show some variations. Thus, the highest earthwork cost found without planar interpolation is $4.6 million, which is 2.7 times bigger than the smallest estimate, $1.68 million. The largest earthwork cost ($8.7 million) with the new approach is only 1.5 times the smallest ($5.5 million). Since the locations of solutions from each replication are not exactly the same, some variations among earthwork cost estimates are understandable. However, variations of 2.7 seem excessive. It is found that the algorithm without planar interpolation underestimates earthwork costs in most cases. Billed costs from real applications might eventually indicate which method is more accurate. However, the manual calculation conducted in section 12.1.4 demonstrated that earthwork estimates using planar interpolation are more reliable than just assuming every point of each cell (or property) in which station points lie has exactly the same ground elevation. Thus, incorporating planar interpolation not only insures more precise earthwork cost estimation but benefits the whole highway alignment optimization process by producing a better solution.

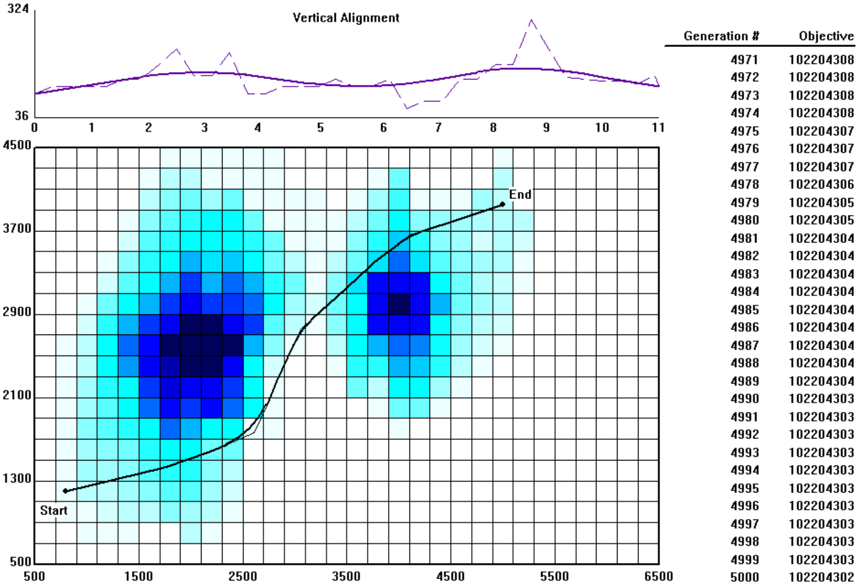


Generation #	Objective
4971	97230879
4972	97230855
4973	97230855
4974	97230851
4975	97230847
4976	97230811
4977	97230811
4978	97230782
4979	97230782
4980	97230782
4981	97230754
4982	97230754
4983	97230748
4984	97230738
4985	97230733
4986	97230733
4987	97230725
4988	97230723
4989	97230723
4990	97230720
4991	97230717
4992	97230708
4993	97230708
4994	97230708
4995	97230708
4996	97230707
4997	97230707
4998	97230706
4999	97230706
5000	97230706

Figure 15.1: Best solution obtained without planar interpolation.

To employ the planar interpolation in a realistic case through the use of a GIS, a study area from Garrett County, Maryland is selected. It contains 128 properties, most of which are residential. There are a few agricultural and commercial properties, a creek, and two two-lane rural highways: Oakland Sang Run Road and Mayhew Inn Road. Its area is approximately 0.08 mi^2 (0.03 km^2). Unit land costs range mostly from $0.02/ft^2 ($0.22/m^2) to $1.47/ft^2 ($16.3/m^2) and costs of structures range from $2,930 to $8,621,800 per property. The elevation information for this case study is primarily taken from the 1983 US

National Geodetic Survey database and a few additional ground elevations are added using the United States Geological Survey (USGS) databases, such as topography maps and aerial photographs available via the web page to be http://terraserver.homeadvisor.msn.com. Figure 15.3 shows the terrain of the study area to be relatively mountainous with a kink at its center.



Generation #	Objective
4971	102204308
4972	102204308
4973	102204308
4974	102204308
4975	102204307
4976	102204307
4977	102204307
4978	102204306
4979	102204305
4980	102204305
4981	102204304
4982	102204304
4983	102204304
4984	102204304
4985	102204304
4986	102204304
4987	102204304
4988	102204304
4989	102204304
4990	102204303
4991	102204303
4992	102204303
4993	102204303
4994	102204303
4995	102204303
4996	102204303
4997	102204303
4998	102204303
4999	102204303
5000	102204302

Figure 15.2: New best solution incorporating planar interpolation.

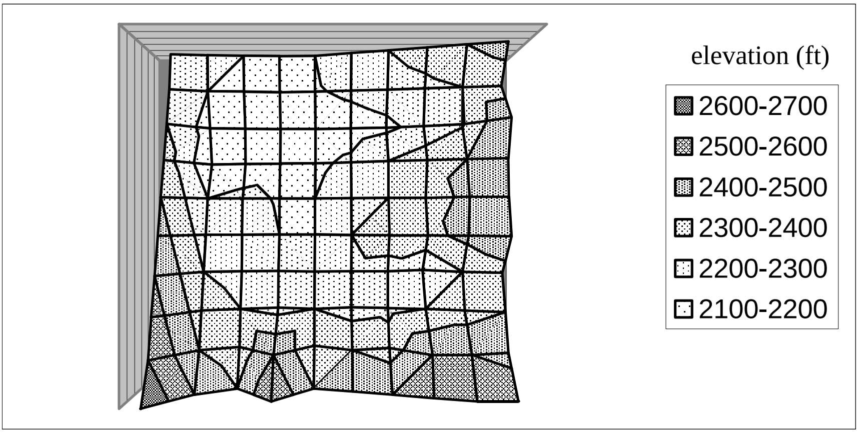


Figure 15.3: Topography of the study area.

Oakland Sang Run Road and Mayhew Inn Road are to be connected at two specified end points. Figure 15.4 shows the best alignment successfully obtained with planar interpolation after 100th generations. Computation time for this particular search was 7,924 seconds (approximately 2 hours and 12 minutes) on a desktop computer with 1 GHz CPU speed and 261 MB RAM. The background map shows the properties and their associated land costs, with darker shades representing more expensive parcels. The total highway costs are approximately $40 million. By carefully observing topography of the study area of Figure 15.3, we can see that the best alignment is obtained in a way that minimizes earthwork (fills and cuts) costs and avoids the hilly central area.

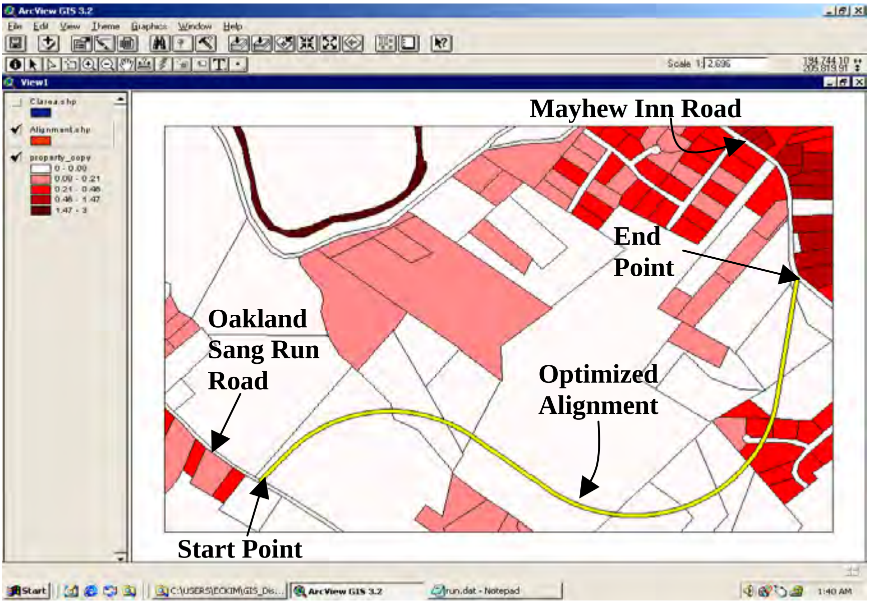
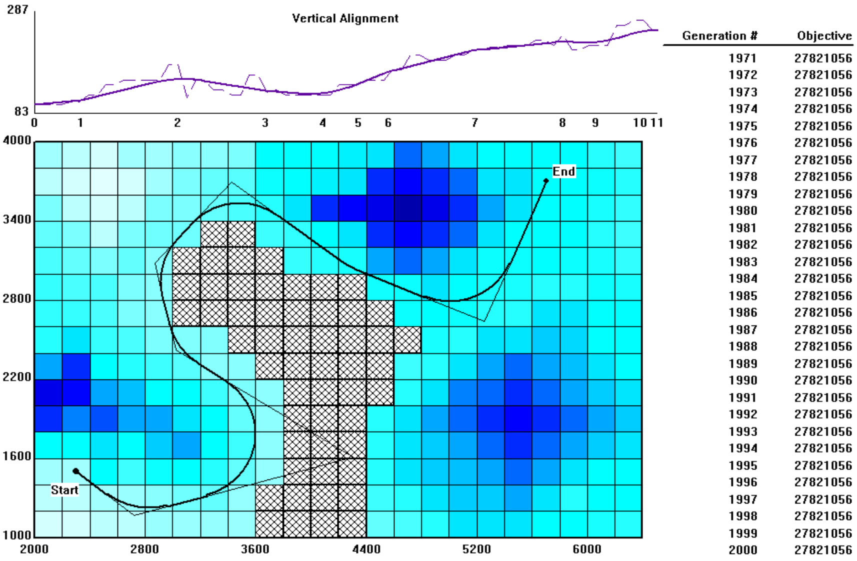


Figure 15.4: Best solution with the planar interpolation using a real GIS map.

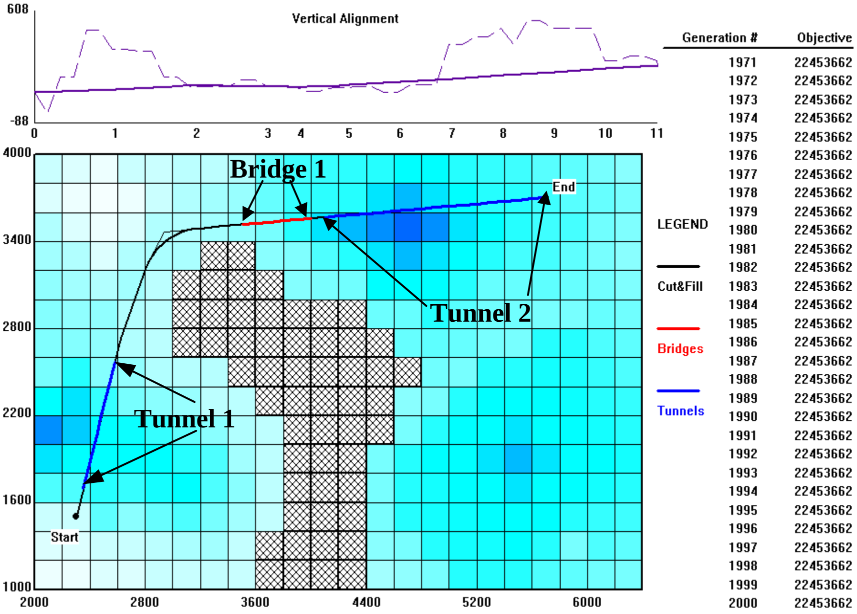
15.2 Results incorporating bridges and tunnels into alignment optimization

To illustrate modeling processes of bridges and tunnels in alignment optimization, two artificial study areas and a real GIS map are designed. Figure 15.5 shows an artificial study area where darker cells represent higher elevations. The algorithm not incorporating bridges and tunnels obtained the best solution at generation 1999. (2000 generations were run in 25 seconds on a desktop PC with 1 GHz CPU speed and 261 MB RAM.) That solution costs approximately $27.8 million. In it, bridges and tunnels are precluded by assigning a high right-of-way value (approximately 50 times higher than for other cells) to the lake.



Generation #	Objective
1971	27821056
1972	27821056
1973	27821056
1974	27821056
1975	27821056
1976	27821056
1977	27821056
1978	27821056
1979	27821056
1980	27821056
1981	27821056
1982	27821056
1983	27821056
1984	27821056
1985	27821056
1986	27821056
1987	27821056
1988	27821056
1989	27821056
1990	27821056
1991	27821056
1992	27821056
1993	27821056
1994	27821056
1995	27821056
1996	27821056
1997	27821056
1998	27821056
1999	27821056
2000	27821056

Figure 15.5: Best solution found not incorporating bridges and tunnels for an artificial study area.



Generation #	Objective
1971	22453662
1972	22453662
1973	22453662
1974	22453662
1975	22453662
1976	22453662
1977	22453662
1978	22453662
1979	22453662
1980	22453662
1981	22453662
1982	22453662
1983	22453662
1984	22453662
1985	22453662
1986	22453662
1987	22453662
1988	22453662
1989	22453662
1990	22453662
1991	22453662
1992	22453662
1993	22453662
1994	22453662
1995	22453662
1996	22453662
1997	22453662
1998	22453662
1999	22453662
2000	22453662

Figure 15.6: New solution with bridges and tunnels for the artificial study area.

The artificial study area in Figure 15.5 is then used without changing the high unit right-of-way cost of the lake to check how the developed cost functions and algorithms for modeling bridges and tunnels search for a solution.

When incorporating the modeling processes for bridges and tunnels, a completely different solution (Figure 15.6) is found at generation 1802, which costs $22.45 million. Computations took 2 minutes and 43 seconds, which is 6.5 times longer than the 25 seconds needed for the solution in Figure 15.5.

The new alignment has two tunnels and one bridge. Since the two tunnels exceed 600 ft (Tunnels 1 and 2 are 800 ft and 1650 ft long, respectively), the assumed additional quadratic cost function of tunnel length described in section 12.6 is applied. The linear excavation cost function used here should be replaced in future models by a quadratic excavation cost function for tunnels longer than 600 ft.

As shown in the upper part of Figure 15.6, new vertical alignment exhibits less emphasis on balancing earthwork volumes than in Figure 12.5. This results in a straighter vertical alignment than in Figure 15.5, with steadier grades and less degree of vertical curvature, which improves safety. Since alignments are significantly changed when incorporating bridges and tunnels, differences of total costs stem from many cost components and not just from the newly added bridge and tunnel costs, which are approximately $7.7 million (34% of the total costs), but all other costs such as right-of-way, pavement and user costs. Table 15.2 as well as Figures 15.7 and 15.8 show the comparative cost breakdown for the two solutions.

Table 15.2: Cost breakdown of two solutions incorporating bridges and tunnels.

	Costs ($) and fractions (%) in Jong's solution	**Costs ($) and fractions (%) in new solution modeling bridges and tunnels**	**Changes in costs ($)**
Total costs	27,821,060 (100.00)	22,443,664 (100.00)	-5,377,396
Pavement costs	2,529,846 (9.09)	1,747,798 (7.79)	-782,048
Right-of-way costs	4,226,407 (15.19)	3,380,102 (15.06)	-846,305
Vehicle operation costs	1,341,328 (4.82)	854,104 (3.81)	-487,224
User time value	13,291,080 (47.77)	5,442,970 (24.25)	-7,848,110
Accident costs	453,062 (1.63)	165,910 (0.74)	-287,152
Earthwork costs	5,978,119 (21.49)	3,134,942 (13.97)	-2,843,177
Tunnel costs		6,981,317 (31.11)	+6,981,317
Bridge costs		715,201 (3.19)	+715,201

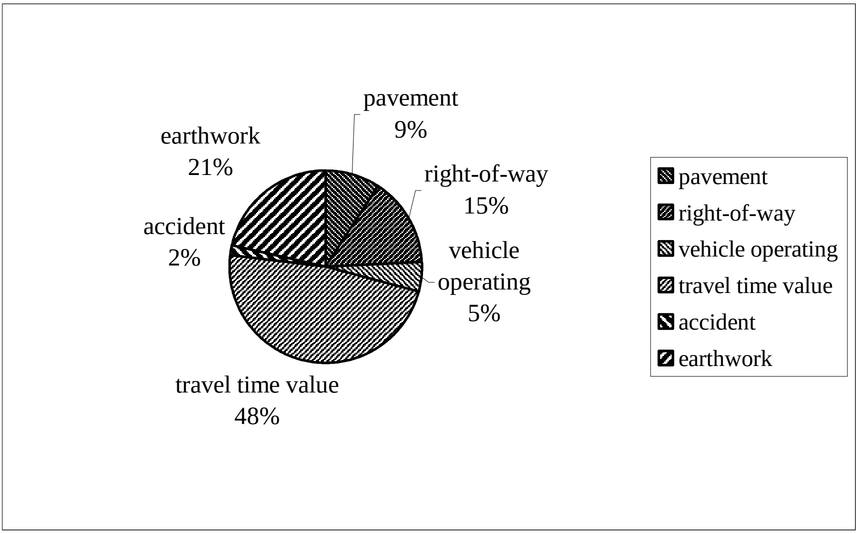


Figure 15.7: Cost breakdown for the alignment in figure 15.5.

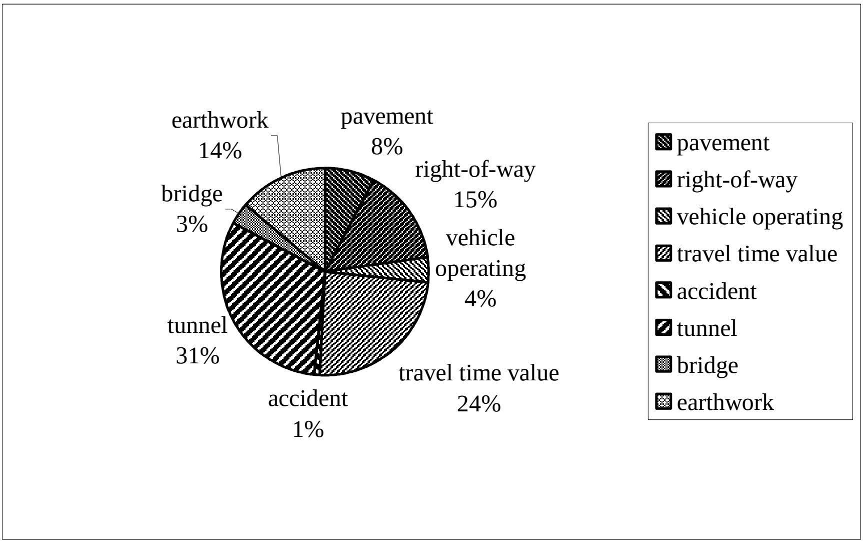
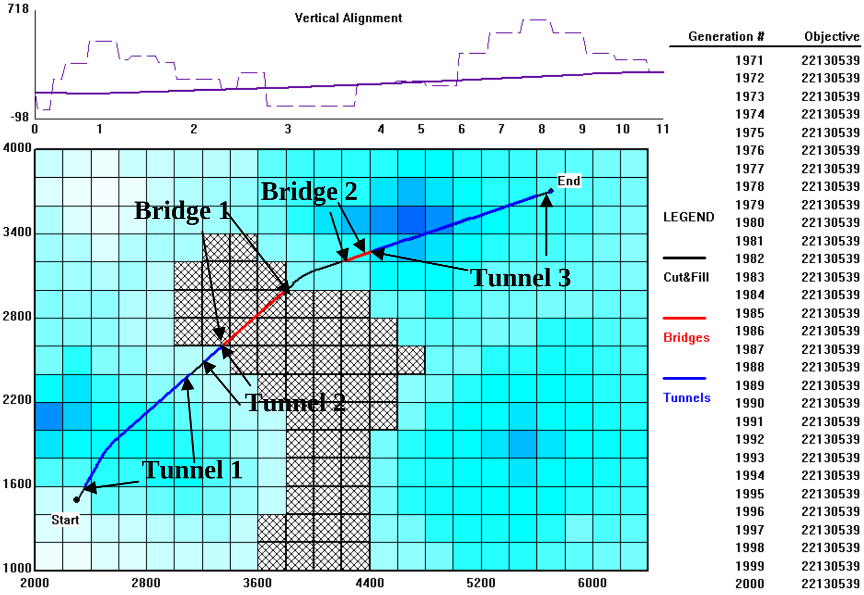


Figure 15.8: Cost breakdown for the new alignment in Figure 15.6.

Based on Table 15.2, and Figures 15.5 through 15.8, it is observed that solutions with significantly different costs and locations are obtained when bridges and tunnels are considered in alignment optimization models.

Since the unit right-of-way cost over the lake of Figures 15.5 and 15.6 was assigned a deliberately high value, the solutions obtained are not supposed to cross the lake. For sensitivity analysis, the following example decreases this unit cost to allow a solution to cross the lake. This unit cost multiplier is now decreased from 50 to 5. From the new right-of-way cost of the lake, a changed solution is obtained (Figure 15.9), which is straighter horizontally and as well as vertically. The new alignment includes three tunnels and two bridges. This solution results from the mountainous topography of the study area and the advantages of a straight alignment which can greatly decrease user costs (by 55% of the solution in Figure 15.5 and by 28% of the solution in Figure 15.8).



Generation #	Objective
1971	22130539
1972	22130539
1973	22130539
1974	22130539
1975	22130539
1976	22130539
1977	22130539
1978	22130539
1979	22130539
1980	22130539
1981	22130539
1982	22130539
1983	22130539
1984	22130539
1985	22130539
1986	22130539
1987	22130539
1988	22130539
1989	22130539
1990	22130539
1991	22130539
1992	22130539
1993	22130539
1994	22130539
1995	22130539
1996	22130539
1997	22130539
1998	22130539
1999	22130539
2000	22130539

Figure 15.9: New best solution for the artificial study area.

The new optimized solution is obtained after 2000 generations and costs $22.1 million. (In Figure 15.9, the optimized solution seems to be found at generation 1971. However, comparisons are based on floating numbers rather than on integers.) Its computation time is 12 minutes and 44 seconds, which is 4.7 times longer than for the example in Figure 15.6. This longer computation time is attributable to the shorter distance between station points (i.e., more station points on the alignment). To sensitively detect the effects of bridge and tunnel lengths, it is desirable to consider more stations. In this example, a 20 ft (6 meter) distance between stations is employed instead of the 100 ft used in the previous two cases. These many stations practically initiate more earthwork computations during the program run in addition to the computation time for bridge and tunnel cost estimation. Table 15.3 shows fractions of each cost item for this solution.

Table 15.3: Cost breakdown for the best new solution.

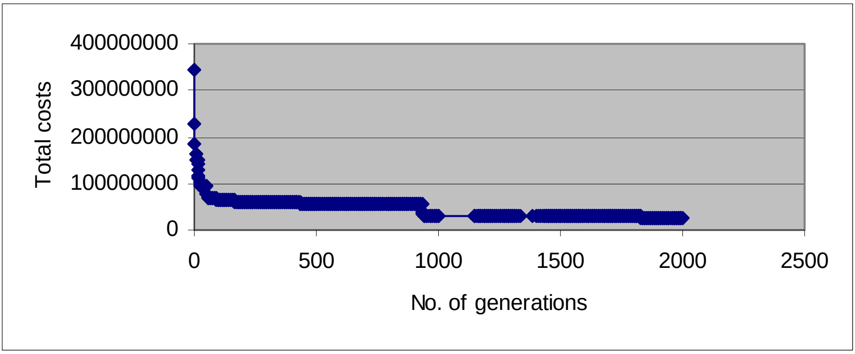
	Costs ($) and fractions (%)
Total costs	22,130,539 (100.00)
Pavement costs	1,517,186 (6.86)
Right-of-way costs	3,960,061 (17.89)
Vehicle operation costs	748,133 (3.38)
User time value	4,564,516 (20.63)
Accident costs	97,823 (0.44)
Earthwork costs	2,303,970 (10.41)
Tunnel costs	6,981,316 (31.55)
Bridge costs	1,957,534 (8.85)

Through the new solutions optimized by considering bridges and tunnels, it is notable that the user travel time and the accident costs are greatly decreased. This is attributable to the straighter alignments with fewer steep gradients. For instance, the travel time value of $13.4 million and accidents costs of $453,062 in the example not considering bridges and tunnels decreased by 65% to $4.6 million and by 78% to $97,823, respectively, in the last example.

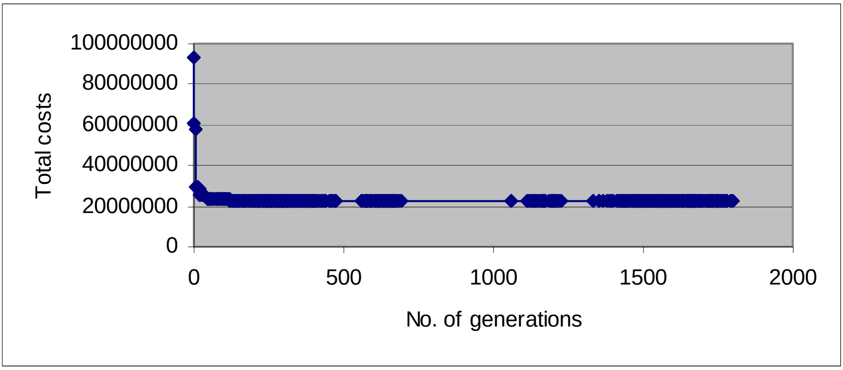
Figure 15.10 shows the changes of the objective values (total costs) through successive generations for three different cases. It is observed that the methods considering bridges and tunnels approach optimal values more quickly than the previous method. The intuitive explanation is that the previous method that cannot consider bridges and tunnels must try more alternatives that are quite circuitous.

Based on the results for the artificial case studies, it can be stated that modeling bridges and tunnels in highway alignment optimization insures less expensive solutions in Table 15.2.

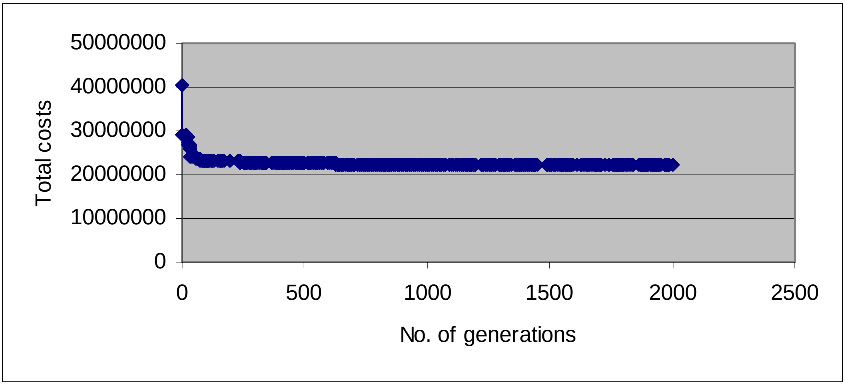
To employ the developed methods with a real GIS map, a study area is taken from Garrett County, Maryland. It contains 90 properties (46 residential, 19 commercial and 24 agricultural) and most parts of the area are agricultural. It includes the Youghiogheny river crossing it from North to South and two two-lane rural highways: White Rock Run Road and Bishoff Road. Its area is approximately 0.26 mi^2 (0.66 km^2). Unit land costs range mostly from $0.02/ft^2 ($0.22/m^2) to $0.47/ft^2 ($5.22/m^2) and costs of structure range from $730 to $122,040. The elevation information for this case study is primarily taken from the same sources presented in section 15.1. Figure 15.11 shows the available topography information. A hill is evident in the middle of the study area.



(a) Case without considering bridges and tunnels.



(b) Case of Figure 15.6.



(c) Case of Figure 15.10.

Figure 15.10: Changes in objective function values over successive generations.

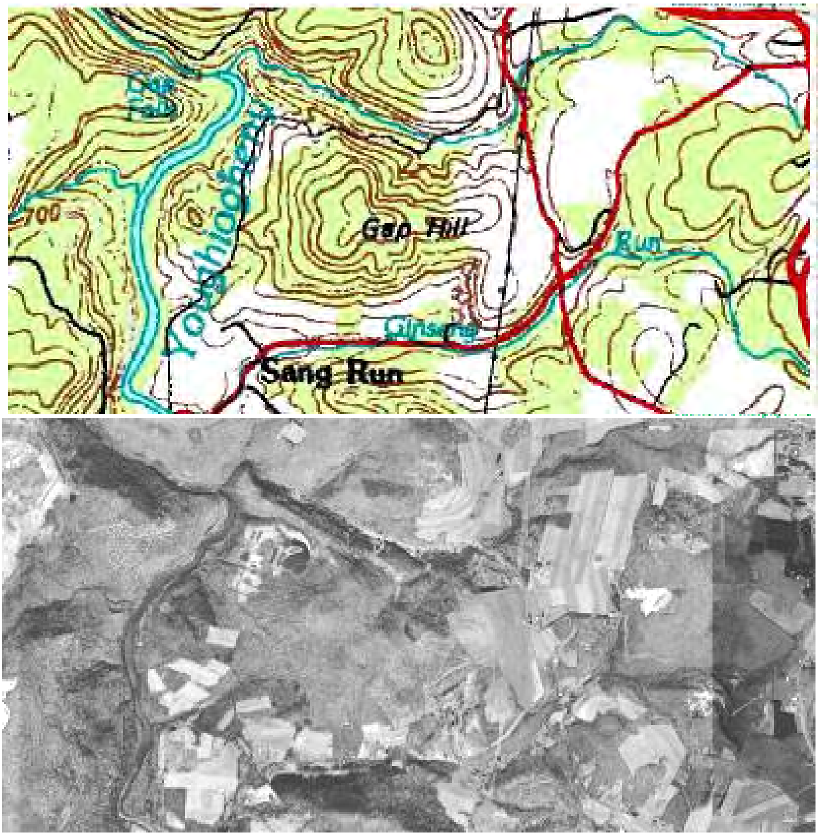


Figure 15.11: Topography and aerial photo of the study area.

Suppose that we construct a new two-lane road connecting two existing roads at specified two end points. We can clearly envision that a solution should have at least one bridge. Figure 15.12 shows the best alignment successfully obtained by applying the developed methods to the study area after 50 generations. (From Figure 15.10, we can see that a near optimal solution was obtained within 50 generations.)

Computation took 7,419 seconds (approximately, 2 hours and 3 minutes). Since data transfers occur between the optimization module operating in the “C” environment and the cost (especially right-of-way) estimation module operating in the GIS environment, extensive linkage and data transaction time was required.

The optimized alignment has two bridges and two tunnels. In Figure 15.13, tunnel 1 and bridge 1 of Figure 15.12 are shown in detail. The alignment is shown from station zero (its start point) through station 41. Tunnel 1 is 200 ft long (between station 9 and station 19, with 20 ft between stations) followed by the bridge 1 which is 220 ft long (station 20 through 31). Between the tunnel 1 and the bridge 1, a 20 ft gap is found. The differences between road and ground elevations of the tunnel 1 and the bridge 1 are found to be approximately 100 ft each. The costs for bridge 1 and tunnel 1 are $368,686 and $558,505, respectively (see Table 15.4). $2.1 million and $966,666 are estimated if fills and cuts are employed for bridge 1 and tunnel 1. (Side slopes of 2.5:1 for fills and 2:1 for cuts are employed.) This result justifies construction of tunnel 1 and bridge 1. The span of bridge 1 is optimized at 75 ft and two equal height (100 ft) piers are obtained. (More detailed analyses for optimizing spans are shown in section 15.3.)

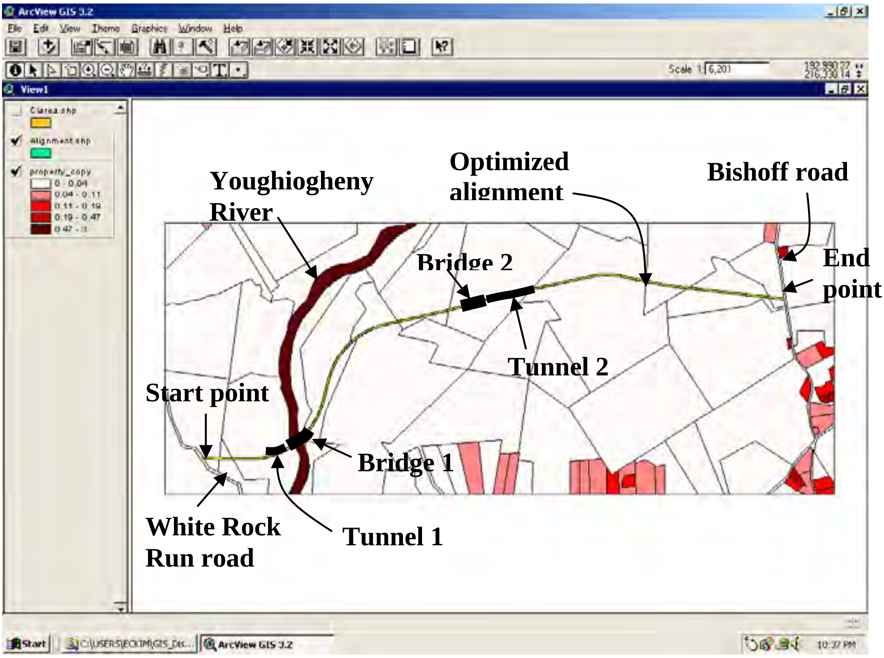


Figure 15.12: Optimized solution for Garrett County example.

Figure 15.14 shows a detailed profile for bridge 2 and tunnel 2 from Figure 15.12. The stations from 96 through 141 are shown. Cuts are seen from station 96 to station 106 and fills are found between station 139 and station 149. Bridge 2 is 220 ft long and tunnel 2 is 420 ft long. The differences between road and ground elevations of bridge 2 and tunnel 2 are found to be approximately 60 ft and 87 ft, respectively. The span of bridge 1 is also optimized at 75 ft. The costs for bridge 2 and tunnel 2 are $402,456 and $1.17 million, respectively (in Table 15.4). $818,399 and $1.6 million are estimated if fills and cuts are substituted for bridge 2 and tunnel 2.

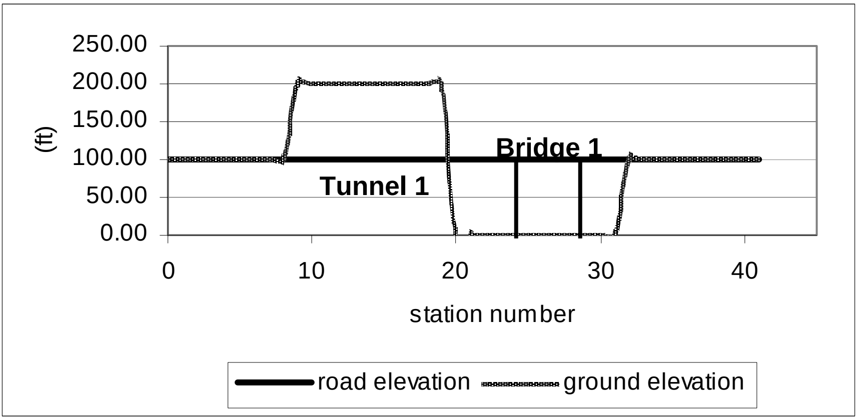


Figure 15.13: Zoomed-in profile of tunnel 1 and bridge 1 for Garrett County example.

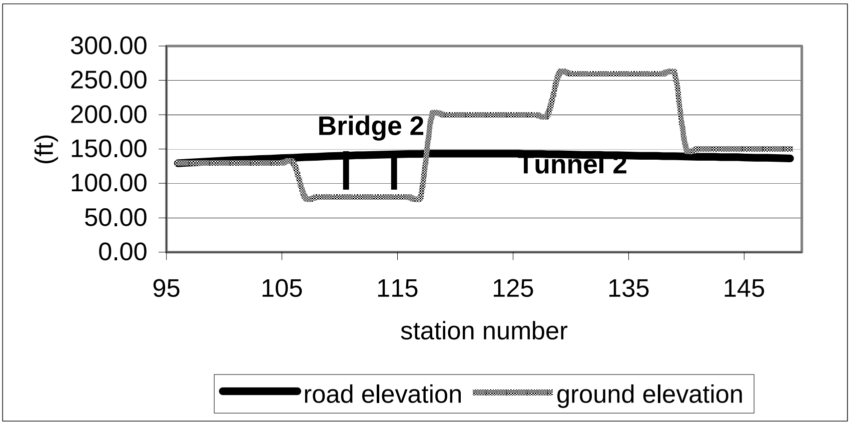


Figure 15.14: Zoomed-in profile of bridge 2 and tunnel 2 for Garrett County example.

The total highway alignment costs are approximately $12.05 million. Table 15.4 and Figure 15.15 show fractions of the total costs. Since the alignment passes through agricultural properties, right-of-way costs are relatively small for this particular example. Costs for two tunnels and two bridges are $1.73 million (14.37%) and $0.77 million (6.40%), respectively. User costs ($7 million, 58%) account for most of the total costs. Based on the above analysis, the bridges and tunnels obtained on alignments are reasonable and efficient compared to fills and cuts.

Table 15.4: Cost breakdown of the optimized Garrett County solution.

	Costs ($) and fractions (%)	
Total costs	12,049,816 (100.00)	
Pavement costs	1,512,952 (12.56)	
Right-of-way costs	107,254 (0.89)	
Vehicle operating costs	768,089 (6.37)	
User time value	6,245,318 (51.83)	
Accident costs	297,212 (2.47)	
Earthwork costs	616,483 (5.12)	
Tunnel costs	Tunnel 1	558,505 (4.63)
	Tunnel 2	1,172,861 (9.73)
Bridge costs	Bridge 1	368,686 (3.06)
	Bridge 2	402,456 (3.34)

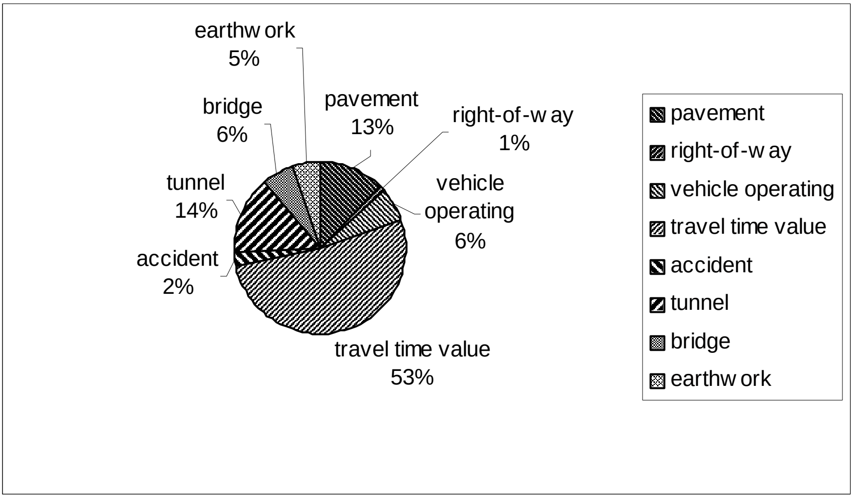


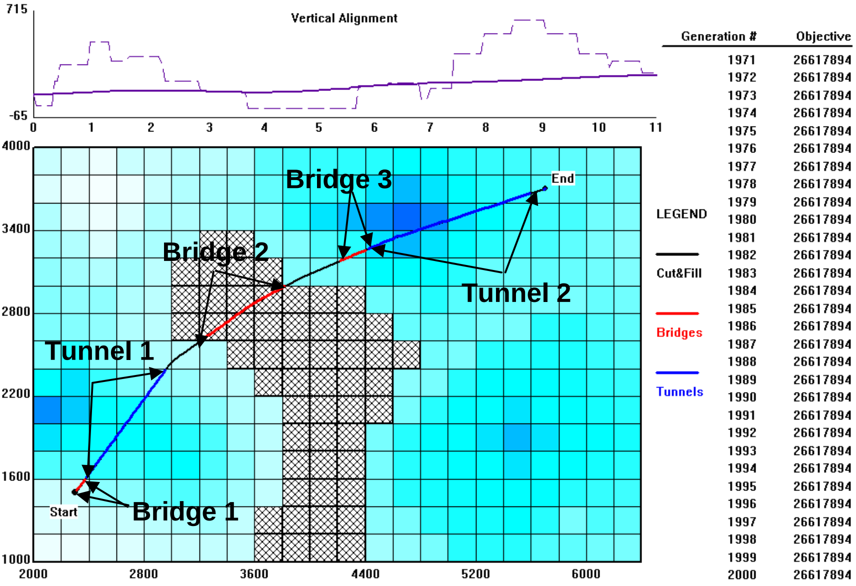
Figure 15.15: Fractions for the optimized solution for Garrett County example.

The optimized solutions and cost estimates in the previous section indicate that the developed cost functions and algorithms for modeling bridges and tunnels are performing reasonably to produce acceptable solutions. In this section, a sensitivity analysis of tunnels is conducted for examining the influence of different unit excavation costs and elevation differences from which tunnels are considered rather than cuts to insure the effectiveness of algorithms for tunnels. It is conceivable that many geologic conditions can be found and different elevation differences can be used when tunnel construction is considered. Since unit tunnel excavation costs significantly differ with the soil types, it is desirable to check how different values affect the solutions and cost estimates. The unit cut cost per cubic yard employed for the previous examples was $15. This value is used for the excavation costs of tunnels and normal cut sections. Several excavation unit costs are employed for tunnels while keeping $15 for normal cut sections. To check sensitivity to different excavation unit costs and elevation differences, five scenarios are designed as shown in Table 15.5.

Table 15.5: Scenarios for sensitivity analysis of tunnels.

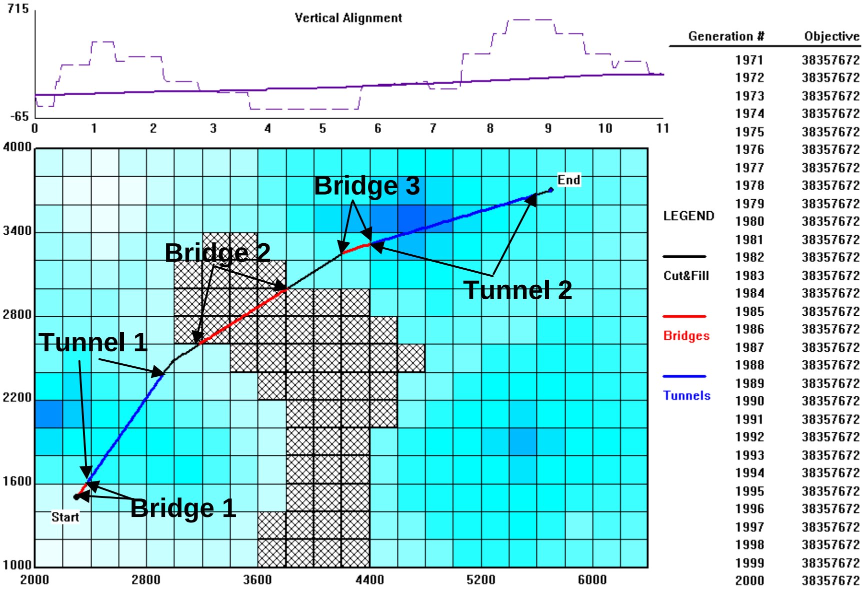
Scenarios	**Elevation differences between the road and the ground beyond which tunnels are considered (ft)**	**Tunnel excavation unit cost ($/yd^3)**
Scenario 1	15	30
Scenario 2	18	60
Scenario 3	21	120
Scenario 4	24	150
Scenario 5	27	250

The analysis is conducted based on the artificial area used previously in Figure 15.9. Figures 15.16 through 15.20 show the results for each scenario after 2000 generations. It is clear that the optimized solutions become more circuitous as elevation differences and tunnel excavation unit costs increase. Thus very large excavation unit costs can be assumed for tunnels if highway agencies do not want to consider tunnels on alignments.



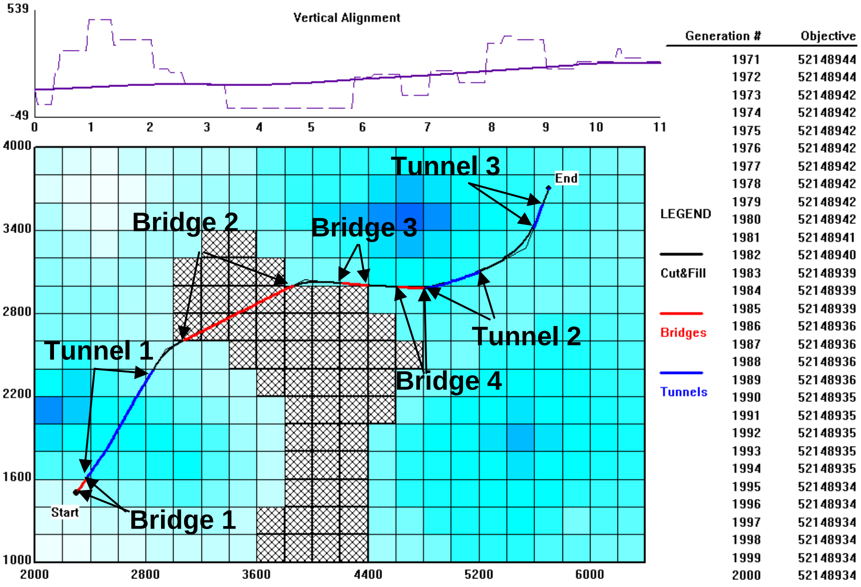
Generation #	Objective
1971	26617894
1972	26617894
1973	26617894
1974	26617894
1975	26617894
1976	26617894
1977	26617894
1978	26617894
1979	26617894
1980	26617894
1981	26617894
1982	26617894
1983	26617894
1984	26617894
1985	26617894
1986	26617894
1987	26617894
1988	26617894
1989	26617894
1990	26617894
1991	26617894
1992	26617894
1993	26617894
1994	26617894
1995	26617894
1996	26617894
1997	26617894
1998	26617894
1999	26617894
2000	26617894

Figure 15.16: Optimized solution (scenario 1) for sensitivity analysis of tunnels.



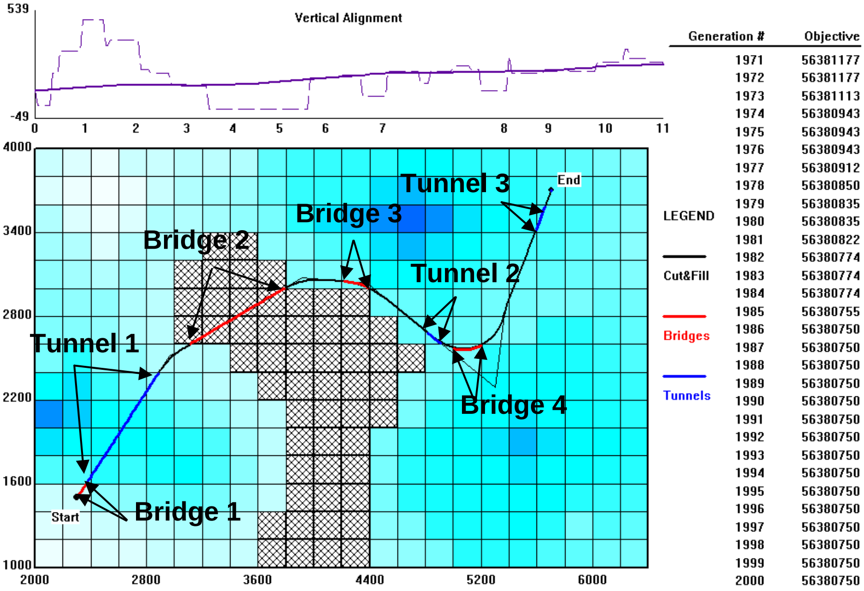
Generation #	Objective
1971	38357672
1972	38357672
1973	38357672
1974	38357672
1975	38357672
1976	38357672
1977	38357672
1978	38357672
1979	38357672
1980	38357672
1981	38357672
1982	38357672
1983	38357672
1984	38357672
1985	38357672
1986	38357672
1987	38357672
1988	38357672
1989	38357672
1990	38357672
1991	38357672
1992	38357672
1993	38357672
1994	38357672
1995	38357672
1996	38357672
1997	38357672
1998	38357672
1999	38357672
2000	38357672

Figure 15.17: Optimized solution (scenario 2) for sensitivity analysis of tunnels.



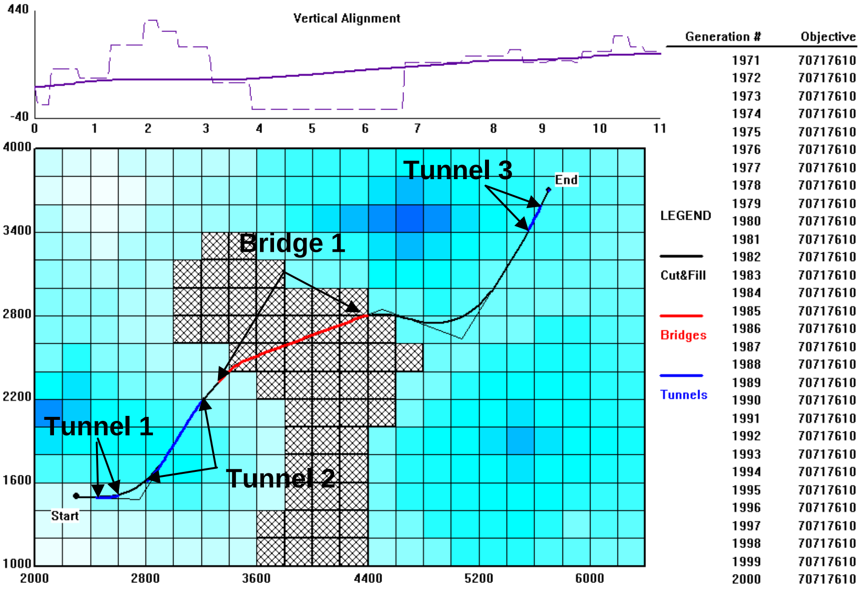
Generation #	Objective
1971	52148944
1972	52148944
1973	52148942
1974	52148942
1975	52148942
1976	52148942
1977	52148942
1978	52148942
1979	52148942
1980	52148942
1981	52148941
1982	52148940
1983	52148939
1984	52148939
1985	52148939
1986	52148936
1987	52148936
1988	52148936
1989	52148936
1990	52148935
1991	52148935
1992	52148935
1993	52148935
1994	52148935
1995	52148934
1996	52148934
1997	52148934
1998	52148934
1999	52148934
2000	52148934

Figure 15.18: Optimized solution (scenario 3) for sensitivity analysis of tunnels.



Generation #	Objective
1971	56381177
1972	56381177
1973	56381113
1974	56380943
1975	56380943
1976	56380943
1977	56380912
1978	56380850
1979	56380835
1980	56380835
1981	56380822
1982	56380774
1983	56380774
1984	56380774
1985	56380755
1986	56380750
1987	56380750
1988	56380750
1989	56380750
1990	56380750
1991	56380750
1992	56380750
1993	56380750
1994	56380750
1995	56380750
1996	56380750
1997	56380750
1998	56380750
1999	56380750
2000	56380750

Figure 15.19: Optimized solution (scenario 4) for sensitivity analysis of tunnels.



Generation #	Objective
1971	70717610
1972	70717610
1973	70717610
1974	70717610
1975	70717610
1976	70717610
1977	70717610
1978	70717610
1979	70717610
1980	70717610
1981	70717610
1982	70717610
1983	70717610
1984	70717610
1985	70717610
1986	70717610
1987	70717610
1988	70717610
1989	70717610
1990	70717610
1991	70717610
1992	70717610
1993	70717610
1994	70717610
1995	70717610
1996	70717610
1997	70717610
1998	70717610
1999	70717610
2000	70717610

Figure 15.20: Optimized solution (scenario 5) for sensitivity analysis of tunnels.

15.3 Case studies with intersections and other structures

Until now, the local optimization of intersections, the modeling of bridges and tunnels and the application of planar interpolation were performed and tested independently. In this section the developed cost functions and algorithms for modeling intersections and other structures are all incorporated together within highway alignment optimization. One fairly complex artificial study and one real GIS application in Washington County, Maryland, are employed to test applicability of the developed methods. Figure 15.21 shows the quite complex topography of the artificial study area, which includes a two-lane rural highway from the center of North to South East, three hills and a creek crossing from North East edge to South. The plan is to build a two-lane rural highway connecting two specified end points while allowing the existing road to be re-optimized.

Three types of test scenarios in Table 15.6 are designed to examine how the algorithms work under various situations. Since the model is designed to automatically select the minimum cost crossing type of the new alignment with the existing road, as developed in section 13.3.2, none of the crossing types is specified for the first test run (scenario 1). In the second and third scenarios, it is assumed that users specify an intersection and an interchange, respectively, as the crossing type with the existing road. A desktop with 1 GHz CPU speed and 261 MB RAM is again used to run the program.

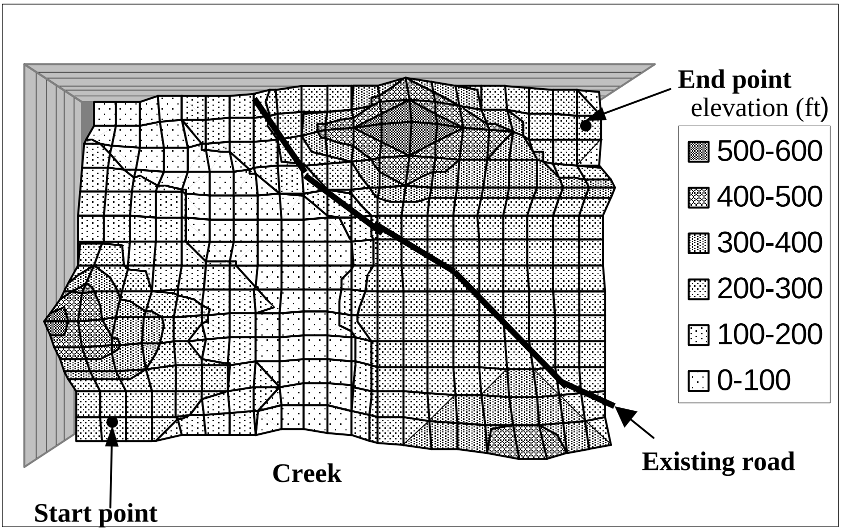
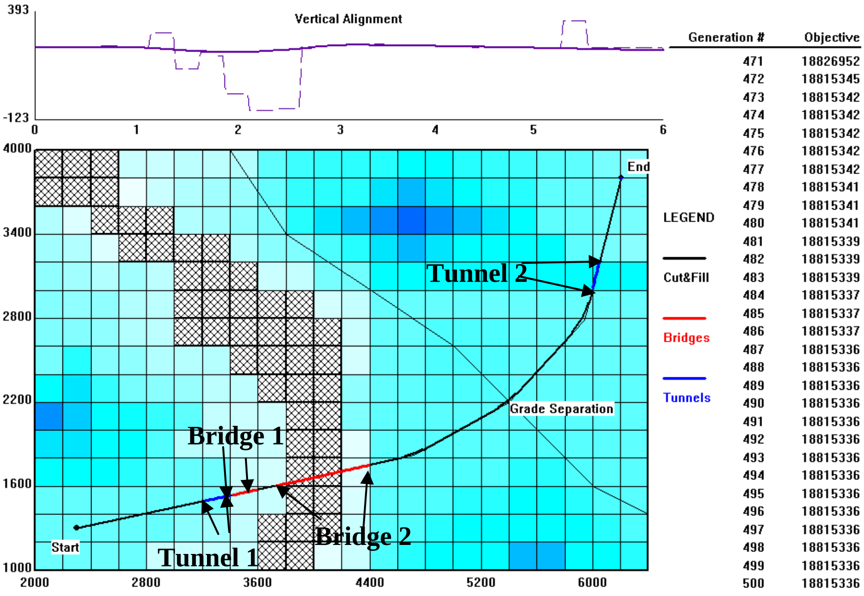


Figure 15.21: Topography of the artificial study area.

Table 15.6: Scenarios for modeling intersections and other structures.

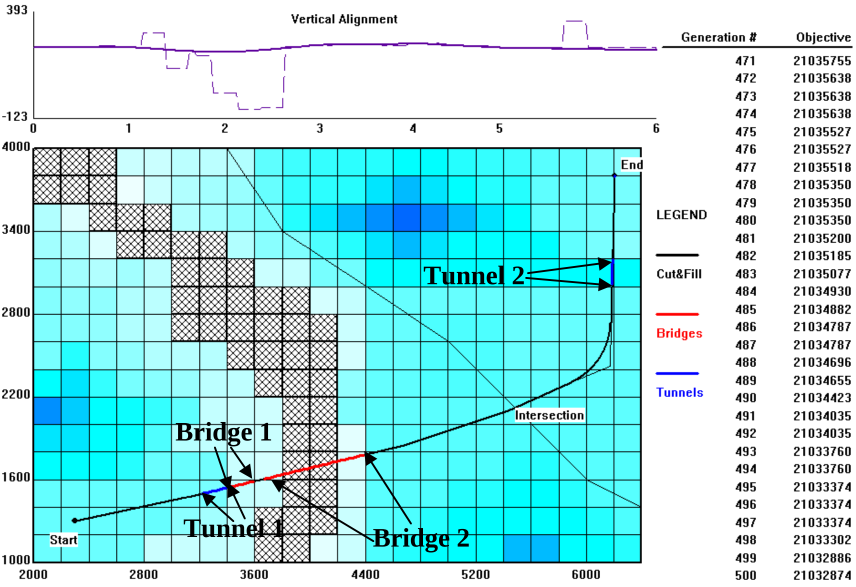
Scenarios	**User specified crossing type with the existing road**	**No. of generations**	**Local optimization of intersections**
Scenario 1	None	500	Yes, if initiated
Scenario 2	Intersection	500	Yes, if initiated
Scenario 3	Interchange (diamond)	500	No

Figures 15.22, 15.23 and 15.24 show the optimized solutions under scenarios 1, 2 and 3. For scenario 1, grade separation is selected for the minimum cost crossing type with the existing road and the best solution has two bridges and two tunnels. For scenario 2, local intersection optimization is not initiated for the best solution obtained by the algorithm since the crossing angle is approximately 70 degrees. The best solution for the scenario 2 also contains two bridges and two tunnels at similar location to scenario 1. For scenario 3, a similar solution, but involving only one bridge, is found with a diamond interchange.



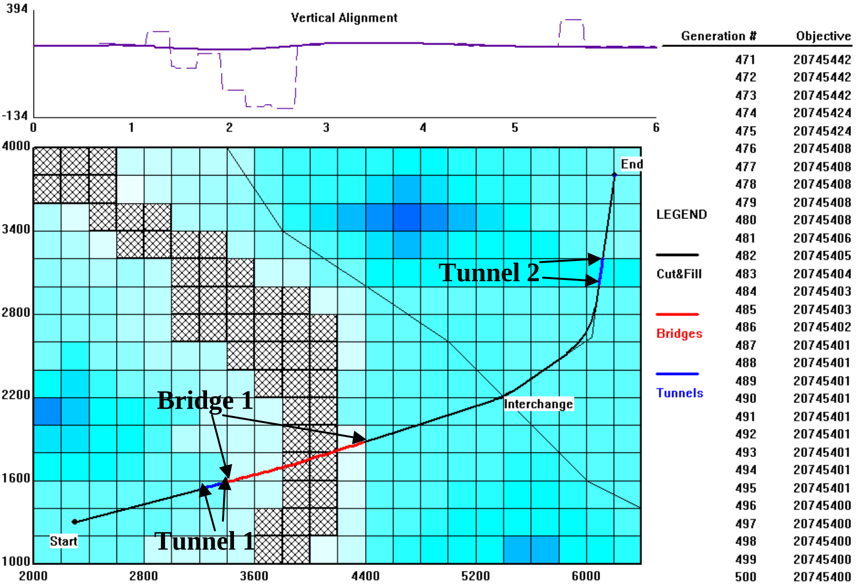
Generation #	Objective
471	18826952
472	18815345
473	18815342
474	18815342
475	18815342
476	18815342
477	18815342
478	18815341
479	18815341
480	18815341
481	18815339
482	18815339
483	18815339
484	18815337
485	18815337
486	18815337
487	18815336
488	18815336
489	18815336
490	18815336
491	18815336
492	18815336
493	18815336
494	18815336
495	18815336
496	18815336
497	18815336
498	18815336
499	18815336
500	18815336

Figure 15.22: Solution for scenario 1.



Generation #	Objective
471	21035755
472	21035638
473	21035638
474	21035638
475	21035527
476	21035527
477	21035518
478	21035350
479	21035350
480	21035350
481	21035200
482	21035185
483	21035077
484	21034930
485	21034882
486	21034787
487	21034787
488	21034696
489	21034655
490	21034423
491	21034035
492	21034035
493	21033760
494	21033760
495	21033374
496	21033374
497	21033374
498	21033302
499	21032886
500	21032874

Figure 15.23: Solution for scenario 2.



Generation #	Objective
471	20745442
472	20745442
473	20745442
474	20745424
475	20745424
476	20745408
477	20745408
478	20745408
479	20745408
480	20745408
481	20745406
482	20745405
483	20745404
484	20745403
485	20745403
486	20745402
487	20745401
488	20745401
489	20745401
490	20745401
491	20745401
492	20745401
493	20745401
494	20745401
495	20745401
496	20745400
497	20745400
498	20745400
499	20745400
500	20745400

Figure 15.24: Solution for scenario 3.

Table 15.7 provides general information for each scenario. Computation time for scenario 2 (4 minutes 50 seconds) was longer than for scenarios 1 (3 minutes 24 seconds) and 3 (3 minutes 25 seconds) since it uses an additional module for local intersection optimization.

Table 15.8, Figures 15.25, 15.26 and 15.27 show detailed cost breakdowns for three scenarios.

Table 15.7: General information for three scenarios.

Scenarios	**Best generation**	**Total costs ($)**	**Computation time**	**Crossing**		**No. of tunnels**	**No. of bridges**
				Type	**Costs ($)**		
Scenario 1	500	18.82 million	3 minutes 24 seconds	Grade separation	91,260	2	2
Scenario 2	500	21.03 million	4 minutes 50 seconds	Intersection	1.49 million	2	2
Scenario 3	499	20.75 million	3 minutes 25 seconds	Diamond interchange	1.19 million	2	1

Table 15.8: Cost breakdowns for three scenarios.

	Scenario 1	**Scenario 2**	**Scenario 3**
Total costs	18,815,336 (100.00)	21,032,874 (100.00)	20,745,408 (100.00)
Structures	91,260 (0.49) grade separation	1,488,056 (7.07) intersection	1,185,976 (5.72) diamond interchange
Pavement	1,850,426 (9.83)	1,944,525 (9.25)	1,872,715 (9.03)
Right-of-way	4,436,439 (23.58)	4,583,726 (21.79)	4,877,867 (23.51)
Vehicle operation	909,814 (4.84)	957,023 (4.55)	921,015 (4.44)
User time value	5,320,836 (28.28)	5,685,093 (27.03)	5,409,726 (26.08)
Accident	241,482 (1.28)	195,153 (0.93)	147,109 (0.71)
Earthwork	1,554,727 (8.26)	1,722,632 (8.19)	1,674,379 (8.07)
Bridges	2,064,630 (10.97)	2,110,944 (10.04)	2,310,899 (11.14)
Tunnels	2,345,722 (12.47)	2,345,722 (11.15)	2,345,722 (12.31)

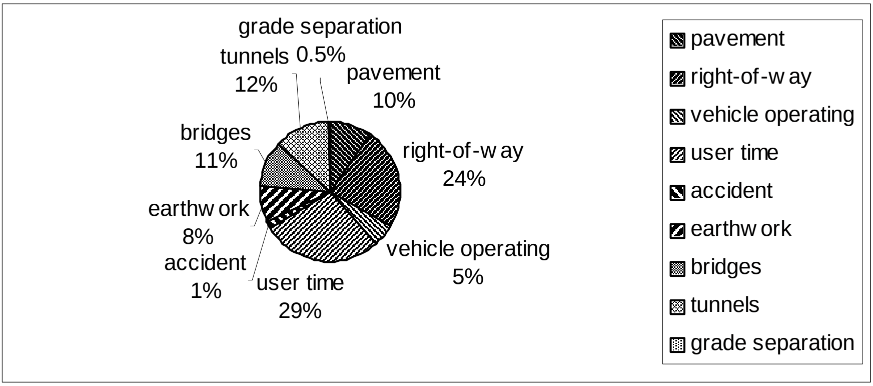


Figure 15.25: Cost breakdown for scenario 1.

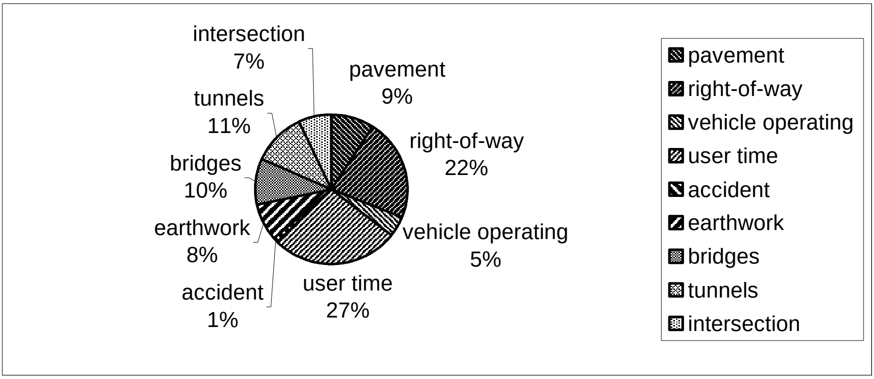


Figure 15.26: Cost breakdown for scenario 2.

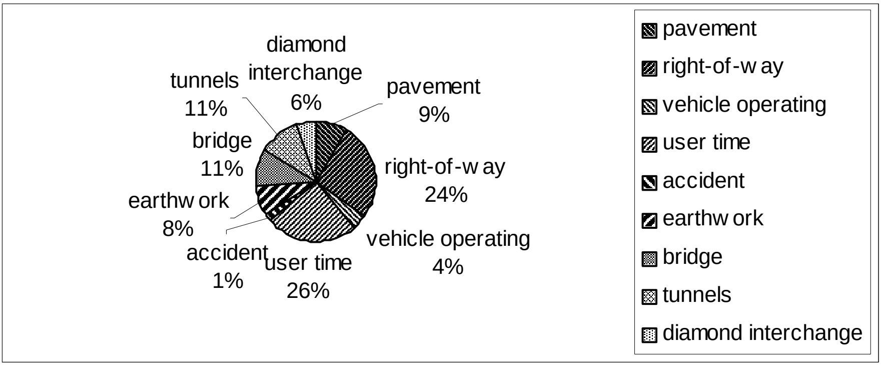


Figure 15.27: Cost breakdown for scenario 3.

Tunnel costs for the three scenarios are similar since the lengths and locations of the two tunnels are almost the same. Interestingly, the intersection costs of scenario 2 exceed those of a diamond interchange for scenario 3. The reasons are clear when we see the detailed cost fractions for the intersection and the interchange as shown in Table 15.9. The intersection of scenario 2 has more accidents, delay and fuel costs than the interchange.

Table 15.9: Fractions for the intersection of scenario 2 and interchange of scenario 3.

	Intersection of scenario 2	**Interchange of scenario 3**
Total costs	1,488,056 (100.00)	1,185,976 (100.00)
Pavement	11,809 (0.79)	9,928 (0.84)
Earthwork	34,625 (2.33)	342,396 (28.87)
Right-of-way	279,953 (18.81)	559,724 (47.20)
Additional delay	509,549 (34.24)	0
Additional fuel	27,415 (1.84)	0
Additional accident	624,705 (41.98)	0
Structure (grade separation)	0	91,260 (7.69)
Additional intersection	0	182,668 (15.40)

Throughout the scenarios, we have three different types of bridges (two types in scenarios 1 and 2 and one type in scenario 3). It is desirable to check how the optimized spans for those three bridges are found. As an example, the bridge found in scenario 3 is analyzed. This bridge is 820 ft long. Five types of spans (25 ft through 125 ft with 25 ft increment, as described in chapter 5) are numerically tested to find a near optimal span for this bridge, as developed in section 12.3. The best span is found to be 75 ft. (A more refined search for better solutions in the vicinity of 75 ft could be conducted in a second stage.) Table 15.10 and Figure 15.28 show how the 75 ft span is selected.

Table 15.10: Optimal span and cost breakdown for the bridge in scenario 3.

Span (ft)	**Total bridge costs ($)**	**Substructure costs ($)**	**Superstructure costs ($)**	**Optimal span (ft)**	**Minimum bridge costs ($)**
25	3,838,944	2,770,651	1,068,293	75	2,310,899
50	2,660,324	1,408,475	1,251,849		
75	2,310,899	901,550	1,409,349		
100	2,529,634	723,172	1,806,462		
125	2,598,741	582,474	2,016,267		

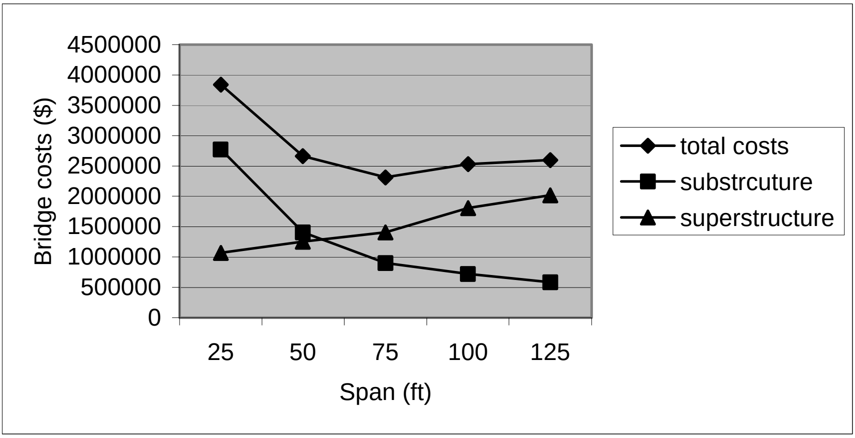


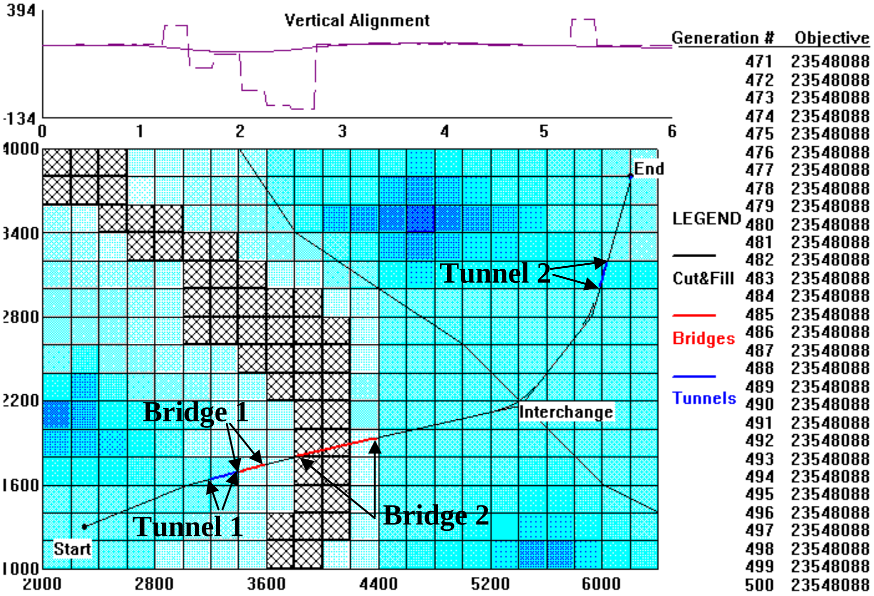
Figure 15.28: Optimal span and cost breakdown for the bridge in scenario 3.

In Table 15.10 and Figure 15.28, the optimized span is found to be 75 ft for the 820 ft long bridge, when testing only five span values to speed up search processes for preliminary highway alignment optimization. However, for structural continuity, we prefer to use equal spans, rather than having one last span of only 70 rather than 75 ft. To preserve the continuity of spans, integer numbers of piers with equal span lengths are analyzed. Also, the high water level flooding constraint (ε_w = 10 ft) in case of is applied. The study area from scenario 3 is used and a diamond interchange is specified for the crossing type with the existing road.

Figure 15.29 shows the optimized solution involving two bridges and two tunnels. Bridge 1 is 180 ft long and bridge 2 is 580 ft long. Bridge 2 is then analyzed for the optimized span. Table 15.11 and Figure 15.30 illustrate how the optimized span is obtained with integer numbers of piers.

Table 15.11: Optimized span and cost breakdown for bridge 2, considering a high water constraint and integer numbers of piers.

No. of spans	**Span (ft)**	**Total bridge costs ($)**	**Substructure costs ($)**	**Superstructure costs ($)**	**Optimized span (ft)**	**Minimum bridge costs ($)**
25	23.2	3,281,835	2,658,577	623,258	**96.7**	**1,359,190**
20	29.0	2,709,342	2,074,619	634,723		
15	38.7	2,088,050	1,478,341	609,709		
10	58	1,772,312	1,011,664	760,648		
8	72.5	1,732,725	862,955	869,770		
7	82.9	1,505,560	638,570	866,990		
6	**96.7**	**1,359,190**	**490,708**	**868,482**		
5	116.0	1,592,510	449,513	1,142,997		
4	145.0	1,753,063	344,941	1,408,122		
2	290.0	2,761,115	157,868	2,603,247		



Generation #	Objective
471	23548088
472	23548088
473	23548088
474	23548088
475	23548088
476	23548088
477	23548088
478	23548088
479	23548088
480	23548088
481	23548088
482	23548088
483	23548088
484	23548088
485	23548088
486	23548088
487	23548088
488	23548088
489	23548088
490	23548088
491	23548088
492	23548088
493	23548088
494	23548088
495	23548088
496	23548088
497	23548088
498	23548088
499	23548088
500	23548088

Figure 15.29: Optimized solution, considering a high water constraint and integer numbers of piers.

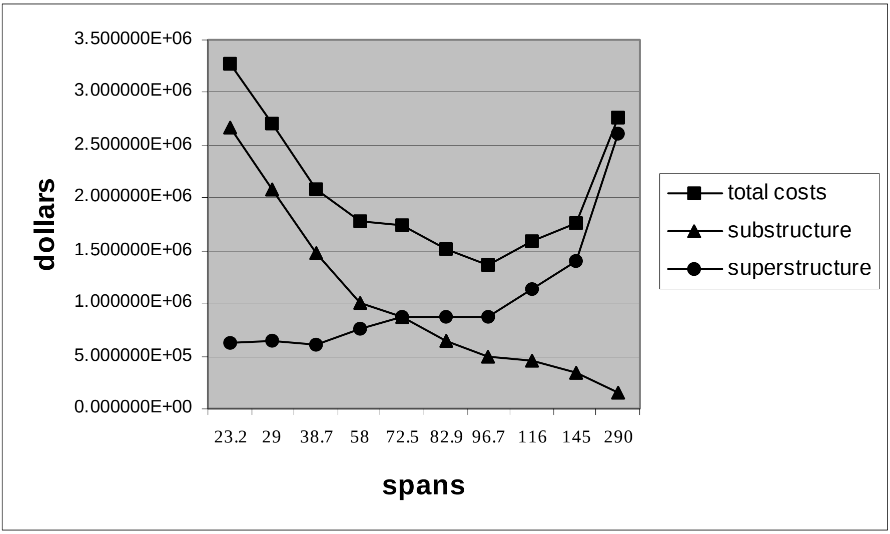


Figure 15.30: Optimized span and cost breakdown for bridge 2, considering a high water constraint and integer numbers of piers.

Ten span lengths (23.2 ft through 290 ft) are analyzed, using integer number of spans from 25 to 2. The optimized span is obtained as 96.7 ft, with 5 piers. It is normally desirable to optimize bridge spans according to the integer numbers of possible spans with real number span lengths. However, the first approach used in Table 15.10 and Figure 15.28 for the bridge of Figure 15.24 can be also employed with a smaller increment. It is found that the optimized span using the first approach with a one foot increment is 97 ft. (In this case, the last span is 95 ft.)

Until now, the optimized solutions did not involve local intersection optimization. To check how local intersection optimization affects the solutions, the original start and end points are deliberately moved, as shown in Figure 15.31. After this change, a new solution is obtained in which local optimization is applied, as shown in Figure 15.32. The new solution is obtained at generation 500. Its total cost is $23.54 million, and its computation time is 33 minutes and 54 seconds. Two tunnels and one bridge are in the new solution and the highway starts with a tunnel.

The crossing angle is approximately 48 degrees. Table 15.12 shows the cost breakdown of the optimized solution before local optimization of the intersection is initiated.

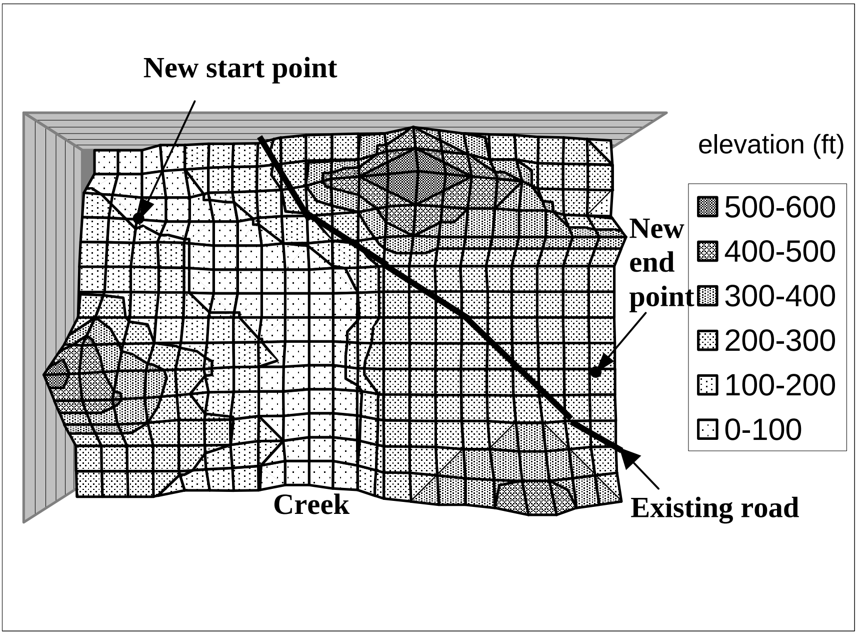
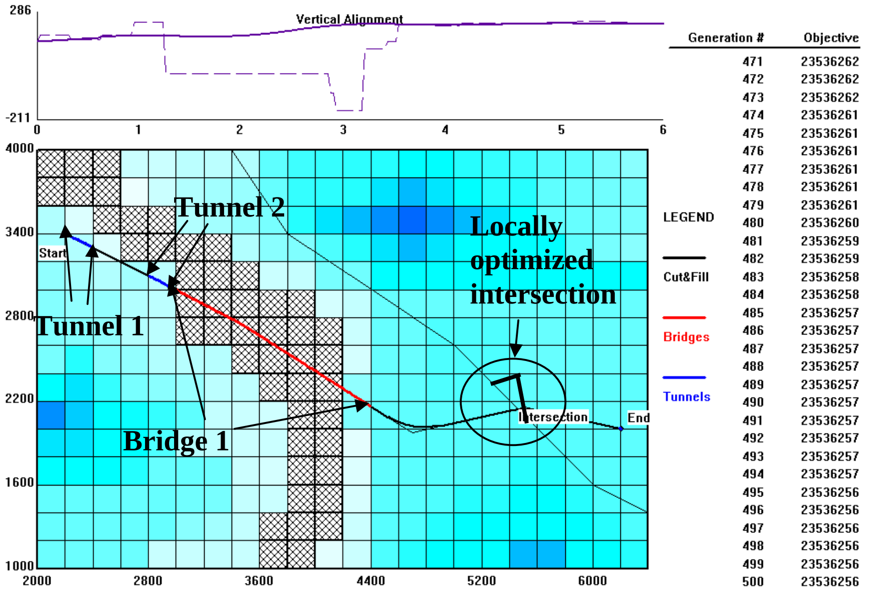


Figure 15.31: Deliberately moved start and end points for new example.



Generation #	Objective
471	23536262
472	23536262
473	23536262
474	23536261
475	23536261
476	23536261
477	23536261
478	23536261
479	23536261
480	23536260
481	23536259
482	23536259
483	23536258
484	23536258
485	23536257
486	23536257
487	23536257
488	23536257
489	23536257
490	23536257
491	23536257
492	23536257
493	23536257
494	23536257
495	23536256
496	23536256
497	23536256
498	23536256
499	23536256
500	23536256

Figure 15.32: The optimized solution involving local intersection optimization with moved start and end points.

Table 15.12: Cost breakdown of the solution before local optimization of the Intersection.

Cost items	Costs ($) and fractions (%)
Total costs	24,554,573 (100.00)
Intersection	3,608,136 (14.69)
Pavement	1,598,651 (6.51)
Right-of-way	5,984,036 (24.37)
Vehicle operation	790,614 (3.22)
User time value	4,997,932 (20.35)
Accidents	242,485 (0.99)
Tunnels	2,606,358 (10.61)
Bridges	3,857,584 (15.71)
Earthwork	859,343 (3.50)
Penalty costs	9,434 (0.04)

Since the best solution is found after local intersection optimization is performed, we are especially interested in intersection costs of Table 15.12. Table 15.13 compares the original intersection costs to the total local intersection optimization costs. It is notable that user costs of the locally optimized intersection decrease by approximately 49% from those for the original intersection due to the perturbed new intersection configurations, namely the 48 degrees crossing angle changed to 90 degrees. (Since it is very hard to find literature about how much crossing angles affect accidents and delays at intersections, a simple linear relation between crossing angles and accident frequencies is assumed here.)

Table 15.13: Cost comparison between original and locally optimized intersections.

Original intersection costs		**Local optimization costs**	
Cost items	Costs ($) and fractions (%)	Cost items	Costs ($) and fractions (%)
Total costs	3,608,136 (100.00)	Total costs	2,589,819 (100.00)
Pavement	11,809 (0.33)	Link earthwork	261,765 (10.11)
Earthwork	314,358 (8.71)	Link right-of-way	312,793 (12.08)
Right-of-way	680,150 (18.85)	Link pavement	41,579 (1.61)
Delay	509,549 (14.12)	New intersection earthwork	185,185 (7.15)
Fuel	27,416 (0.76)	New intersection right-of-way	506,622 (19.56)
Accidents	2,064,854 (57.23)	New intersection delay	254,775 (9.84)
		New intersection fuel	24,673 (0.95)
		New intersection accident	1,002,427 (38.71)

With this local optimization of intersections, we can see that approximately $\$10^6$ are saved. More importantly, the best solution found is not discarded during successive generations just because of the unacceptable crossing angle between the existing road and the new alignment. The final cost breakdown is shown in Table 15.14 and Figure 15.33.

Table 15.14: Cost breakdown of the solution after local optimization.

Cost items	**Costs ($) and fractions (%)**
Total costs	23,536,256 (100.00)
Local intersection optimization	2,589,819 (10.55)
Pavement	1,598,651 (6.79)
Right-of-way	5,984,036 (25.42)
Vehicle operation	790,614 (3.36)
User time value	4,997,932 (21.24)
Accidents	242,485 (1.03)
Tunnels	2,606,358 (11.07)
Bridges	3,857,584 (16.39)
Earthwork	859,343 (3.65)
Penalty costs	9,434 (0.04)

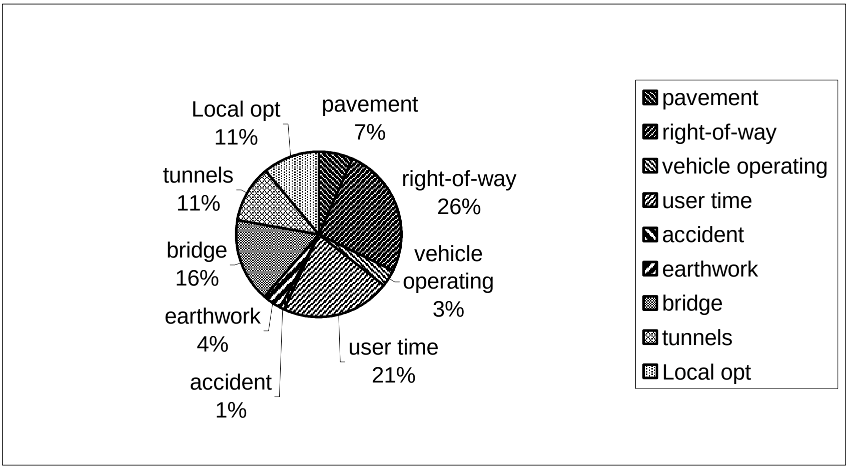


Figure 15.33: Cost breakdown after local optimization.

In the previous examples in which local intersection optimization was not employed, it was normal that objection functions decrease very rapidly in the early generations and very slightly near the end.

However, when alternatives involving local intersection optimization are found to be good solutions at some particular generations and thus selected for producing the population for the next generations, it is possible to observe a sudden drop of objective functions in fairly late generations. This can be observed in Figure 15.34. Thus, many generations should be run whenever local optimization of intersections may be desirable.

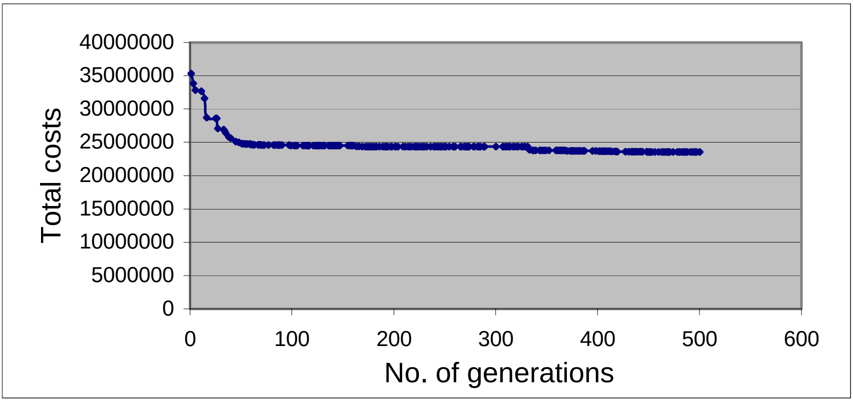


Figure 15.34: Changes of objective functions for local intersection optimization.

An additional case study with a real GIS map from Washington County, Maryland is conducted to check how the developed cost functions and algorithms for intersections and other structures can be integrated with a GIS.

The study area is relatively large and quite complex compared to the previous case examples. It contains 416 properties (306 residential, 75 agricultural and 12 commercial) including highways and creeks. Conocoheague creek is winding in the upper and left parts of the study area and there are four major highways: MD 494 crossing the study area from West to East, MD 58 at the right top, Broadfording Road at the left bottom and Shinham Road parallel with the creek at the top. Its area is approximately 0.4 mi^2 (1.02 km^2). Unit land costs range from \$0.01/ft^2 (\$0.11/m^2) to \$3.0/ft^2 (\$33.3/m^2) and costs of structures range from \$2,880 to \$767,300 per property. The elevation information for this case study is also primarily taken from the same sources presented in section 15.1. In Figure 15.35 a photograph from the web shows a relatively flat study area.

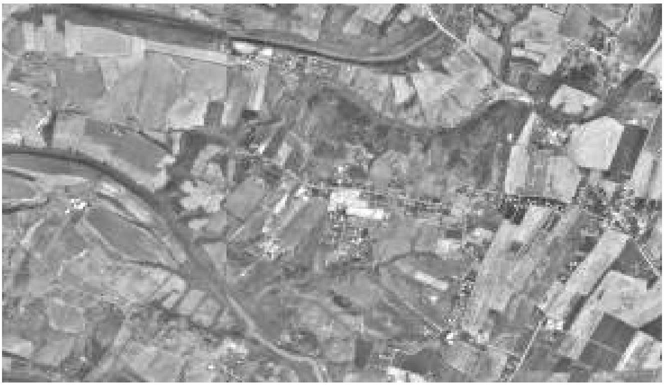


Figure 15.35: Aerial photo of the study area, Washington County, Maryland. Source: http://terraserver.homeadvisor.msn.com

It is assumed that a connection between Broadfording Road and MD 58 is considered to provide a better access with residents of the study area. This requires a new intersection between the new alignment and MD 494. After all information is combined and 100 generations are run, an optimized solution (Figure 15.36) involving local intersection optimization is obtained at generation 99.

The total costs are $8.11 million and computation time is approximately 19 hours and 7 minutes. The much longer computation times than those observed in artificial case studies resulted from many factors such as (1) communication between two different computing environments, GIS and C software, (2) local intersection optimization during the program run, (3) many alternatives causing local intersection optimization through the successive generations, (4) a relatively large number of properties in the study area and (5) more generations, i.e., 100. It is especially notable that local intersection optimization is the main reason for longer computations while in the real GIS example case studied in section 15.2, where no local optimization was involved, the computation time was only about 2 hours.

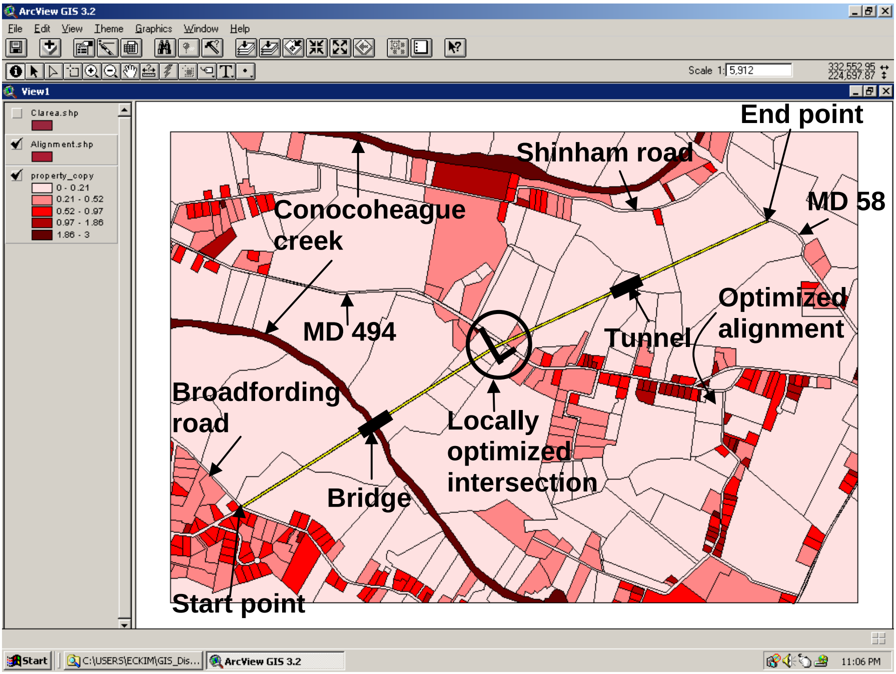


Figure 15.36: Optimized solution for the study area, Washington County, Maryland.

The optimized solution is relatively straight. It has one bridge crossing Conocoheague creek and one tunnel. The crossing angle with the existing MD 494 is approximately 58 degrees. Table 15.15 shows the cost breakdown for the optimized solution before local optimization of the intersection is initiated.

As in the case in Figure 15.32, we also need to compare the original intersection costs to local intersection optimization costs, since the best solution is found after local optimization.

Table 15.16 shows the comparative results. With local intersection optimization, approximately $0.9 million are saved. For this example, accident costs are dominating for two cases while agency costs such as right-of-way, earthwork and pavement account for only small parts of the total costs. In particular, earthwork costs are found to be nearly zero due to very flat topography at the crossing point between the existing road and the optimized alignment.

Table 15.15: Cost breakdown of Washington County solution before local optimization of the intersection.

Cost items	**Costs ($) and fractions (%)**
Total costs	9,011,469 (100.00)
Intersection	1,991,359 (22.10)
Pavement	1,280,939 (14.21)
Right-of-way	516,505 (5.73)
Vehicle operation	627,861 (6.97)
User time value	3,403,653 (37.77)
Accidents	47,897 (0.53)
Tunnels	558,505 (6.20)
Bridges	266,927 (2.96)
Earthwork	317,823 (3.53)

Table 15.16: Cost comparison between original and locally optimized intersections for Washington County.

Original intersection costs		**Local optimization costs**	
Cost items	**Costs ($) and fractions(%)**	**Cost items**	**Costs ($) and fractions(%)**
Total costs	1,991,359 (100.00)	Total costs	1,089,186 (100.00)
Pavement	11,809 (0.59)	Link earthwork	35 (0.00)
Earthwork	0 (0.00)	Link right-of-way	69,650 (6.39)
Right-of-way	62,789 (3.15)	Link pavement	13,060 (1.20)
Delay	509,549 (25.59)	New intersection earthwork	0 (0.00)
Fuel	27,415 (1.38)	New intersection right-of-way	37,095 (3.41)
Accidents	1,379,797 (69.29)	New intersection delay	254,775 (23.39)
		New intersection fuel	24,672 (2.27)
		New intersection accident	689,899 (63.34)

It should be remembered that the optimized solution cannot be obtained using the previous methods which did not incorporate local intersection optimization.

The final cost breakdown for this case is shown in Figure 15.37 and Table 15.17. User costs account for more than 50 % of the total costs. Bridge and tunnels costs are 3.29 % and 6.89 %, respectively. The costs for local intersection optimization are found to be $1.09 million and account for 13.43 % of the total costs.

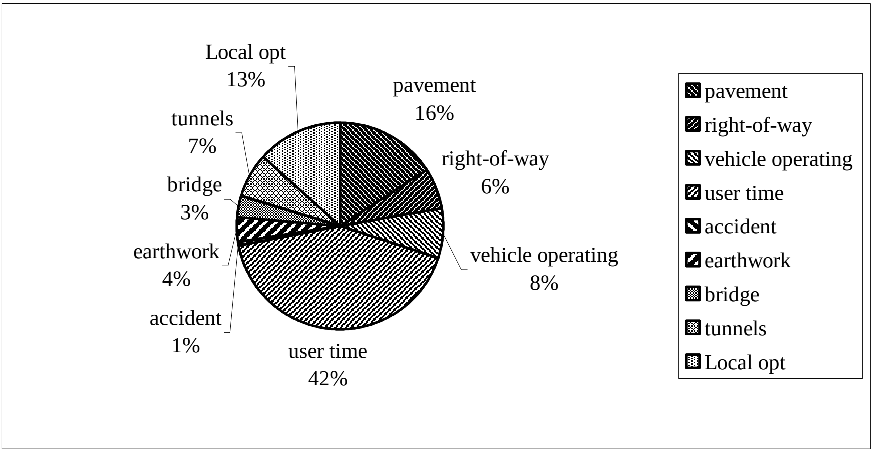


Figure 15.37: Final cost breakdown for the solution of Washington County after local optimization.

Table 15.17: Final cost breakdown for the solution of Washington County after local optimization.

Cost items	**Costs ($) and fractions (%)**
Total costs	8,109,296 (100.00)
Local intersection optimization	1,089,186 (13.43)
Pavement	1,280,939 (15.80)
Right-of-way	516,505 (6.37)
Vehicle operation	627,861 (7.74)
User time value	3,403,653 (41.97)
Accidents	47,897 (0.59)
Tunnels	558,505 (6.89)
Bridges	266,927 (3.29)
Earthwork	317,823 (3.92)

For the one bridge in the optimized solution, the length is 220 ft and the optimal span is found at 25 ft, as shown in Table 15.18 and Figure 15.38. In section 12.3, it was shown that the bridge cost function might have local optima. Figure 15.38 shows such a case. The 75 ft span might have been wrongly selected if the 25 ft span had not been evaluated.

When estimating the total bridge costs, it is normally observed as in Figure 15.30 that the cost curves for substructure and superstructure cross each other. However, a crossing point is not observed in Figure 15.38 in this example. This probably occurs because elevation differences between the ground and the new alignment are relatively small (approximately 20ft), and so are the substructure costs of the bridge, due to the short heights of the associated piers. (In the previous cases, elevation differences were 60 ft to100 ft.)

Table 15.18: Optimal span and cost breakdown of the bridge in Washington County case.

Span (ft)	**Total bridge costs ($)**	**Substructure costs ($)**	**Superstructure costs ($)**	**Optimal span (ft)**	**Minimum bridge costs ($)**
25	266,927	32,424	234,503	25	266,927
50	314,621	16,562	298,059		
75	310,459	8,456	302,003		
100	501,302	8,631	492,671		

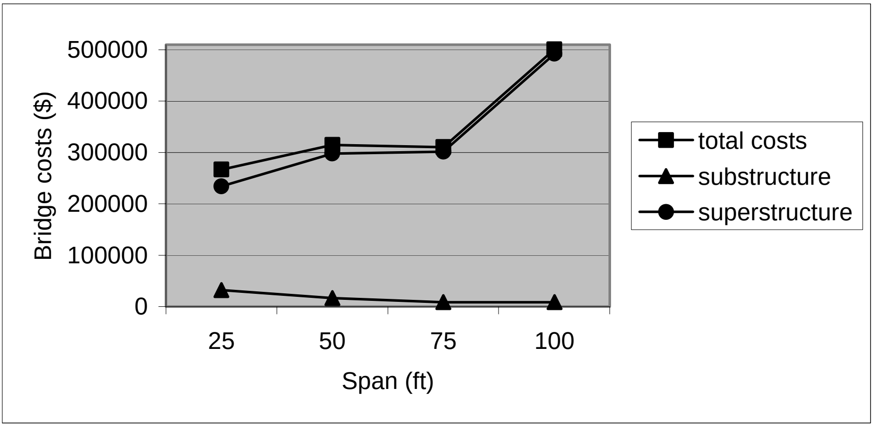


Figure 15.38: Optimized span and cost breakdown of the Washington County bridge.

When a solution involves local intersection optimization, it is conceivable that the objective function value may drop suddenly because the solution is not obtained from the fine-tuning process of the best solution found at the previous generation. Since the solution obtained in Figure 15.36 involves local intersection optimization, it is presumable that significant changes in the objective function may occur in late generations as well as early ones. This is why 100 generations were chosen for the program run as recommended after observing Figure 15.34, although extensive computations were required. Figure 15.39 shows changes in the objective function over 100 generations for the Washington County case study.

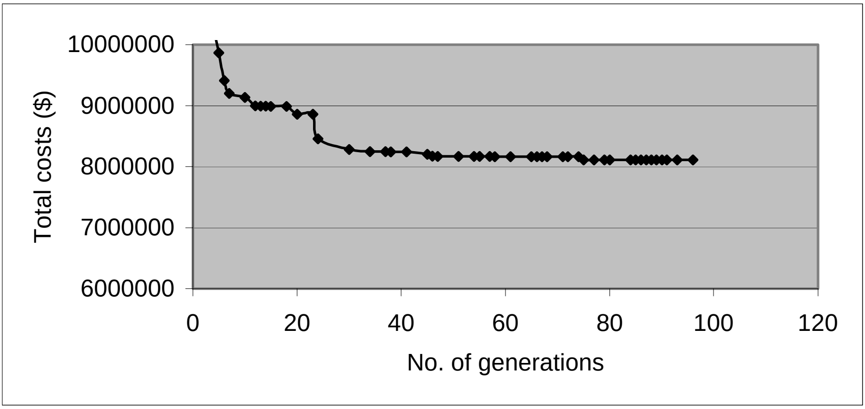


Figure 15.39: Changes of objective functions for Washington County case study.

15.4 Two-stage alignment optimization

In this section, a two-stage alignment optimization is performed to improve both computation time and the quality of solutions based on an artificial study area. In many optimization processes, subdividing large problems into suitable pieces can decrease the computation time and produce a better solution. This argument also applies to this study, since modeling intersections and other structures in alignment optimization repeatedly involves fine search steps for structures.

Another issue for computational efficiency and search performance is the population size. Goldberg (1989) has shown that the efficiency of a GA in reaching a global optimum instead of local ones largely depends on the population size while requiring more computation time.

In this section, the population size for each generation is set proportionally to the number of decision variables (points of intersections, PI’s) as shown in Table 15.19. For example, if three points of intersections are used for generating highway alignments, then the population size is set at 30 (= 3 × 10) while a population of 150 is used for 15 points of intersections.

One artificial study area (see Figure 15.23) previously used in section 15.3 is chosen for a two-stage alignment optimization and three scenarios shown in Table 15.19 are designed to check the search performance and computational efficiency of the developed methods for intersections and other structures.

Table 15.19: Scenarios for a two-stage optimization.

Type of optimization	**Scenarios**	**Subdivided highway segments**	**Number of points of intersections (PI's) between start and end points for an one-stage optimization**	**Population size**	**Number of generations**
One-stage optimization	Scenario 1	0	3	30	2000
	Scenario 2	0	15	150	2000
Two-stage optimization	Scenario 3	4	Number of points of intersections (PI's) within each segment	Population size for each segment	Number of generations for each segment
			3	30	2000

Scenarios 1 and 2 are devised for a one-stage optimization while scenario 3 is for a two-stage optimization. The results of scenarios 1 and 2 can be used for assessing the effects of the population size in computational time and the quality of solutions while the result of scenario 3 can be compared to the results of both scenario 1 and 2 for checking how much improvement is found with a two-stage optimization.

A crossing type with an existing road is assumed here to be an intersection. $100 and 20 ft are assumed for the unit excavation cost and the elevation difference beyond which tunnels are considered rather than cuts. Three replications are run for scenarios 1 and 2. Table 15.20 shows comparison between two scenarios and Figures 15.40 and 15.41 show the best solutions among three replications under scenarios 1 and 2.

In scenario 1, two bridges, one tunnel and an intersection are found while scenario 2 shows one bridge, an intersection and no tunnels in the best solutions. The best solutions for both scenarios do not involve locally optimized intersections. (Crossing angles are found to be 65.0 degrees for both.) However, it is observed that local optimization of intersections was initiated for many alternatives during successive generations.

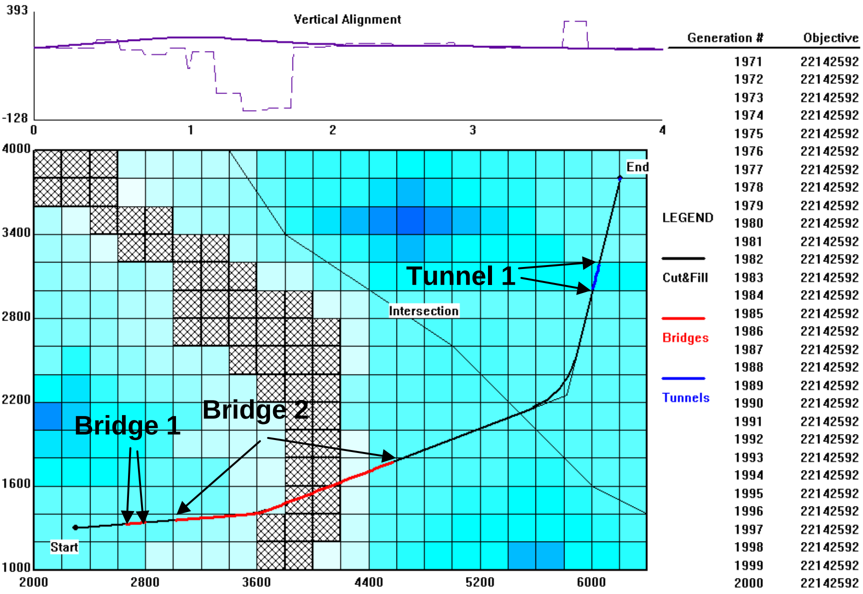
Table 15.20: Comparison between scenarios 1 and 2.

Type of optimi -zation	**Scenario**	**Replications and seeds**	**Total costs (million)**	**Computation time (hour/minute/second)**	**Number of alternatives involving local intersection optimization and (total numbers of alternatives)**
One-stage optimi-zation	Scenario 1	1, 21088	22.29	0 / 32 / 34	4,988 (60,000)
		2, 20035	22.14	0 / 31 / 14	4,796 (60,000)
		3, 21276	22.43	0 / 31 / 18	4,814 (60,000)
	Scenario 2	1, 21088	20.78	2 / 39 / 8	6,197 (300,000)
		2, 20035	17.29	2 / 27 / 38	5,425 (300,000)
		3, 21276	18.07	2 / 35 / 20	5,462 (300,000)

Total costs of the best solutions for each scenario significantly decreased from $22.14 million to $17.29 million ($4.85 million, 21.9% improvement) while computation time for scenario 2 is 4.72 times longer for scenario 1. These results indicate the tradeoff between evaluating more alternatives (exploring more of the search space thus obtaining better solutions) and computation time.

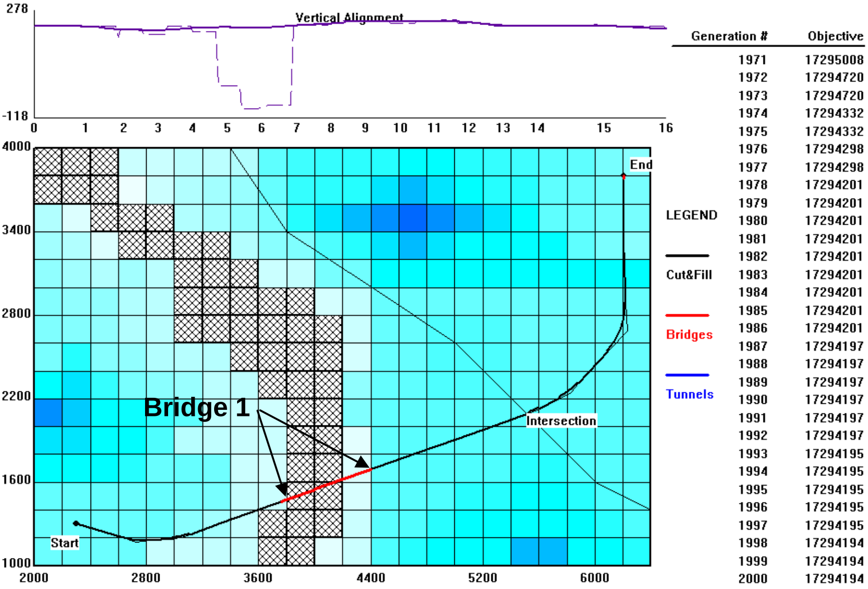
To check computation time and the quality of solutions of a two-stage optimization, the three points of intersections of scenario 1 are obtained after the one-stage optimization and used to subdivide the whole alignment into segments. Figure 15.42 shows the resulting subdivision. Since direct use of the points of intersection for the start or end points of each segment may not insure the smoothness when connecting the solutions from each segment, a 150 ft gap is inserted between segments. It is notable that only segment 3 requires evaluation of intersections and local intersection optimization.

Table 15.21 shows the results for the two-stage optimization under scenario 3. Total computation time for scenario 3 is 6 minutes and 34 seconds, which is only approximately 20% of the time for scenario 1. It is notable that the computation times for segments 1, 2 and 4, which do not require local optimization of intersections, are less than 20 seconds. Meanwhile, the overall costs of each segment are found to be $16.74 million (76% of the one-stage optimization solution).



Generation #	Objective
1971	22142592
1972	22142592
1973	22142592
1974	22142592
1975	22142592
1976	22142592
1977	22142592
1978	22142592
1979	22142592
1980	22142592
1981	22142592
1982	22142592
1983	22142592
1984	22142592
1985	22142592
1986	22142592
1987	22142592
1988	22142592
1989	22142592
1990	22142592
1991	22142592
1992	22142592
1993	22142592
1994	22142592
1995	22142592
1996	22142592
1997	22142592
1998	22142592
1999	22142592
2000	22142592

Figure 15.40: The best solution among three replications for scenario 1.



Generation #	Objective
1971	17295008
1972	17294720
1973	17294720
1974	17294332
1975	17294332
1976	17294298
1977	17294298
1978	17294201
1979	17294201
1980	17294201
1981	17294201
1982	17294201
1983	17294201
1984	17294201
1985	17294201
1986	17294201
1987	17294197
1988	17294197
1989	17294197
1990	17294197
1991	17294197
1992	17294197
1993	17294195
1994	17294195
1995	17294195
1996	17294195
1997	17294195
1998	17294194
1999	17294194
2000	17294194

Figure 15.41: The best solution among three replications for scenario 2.

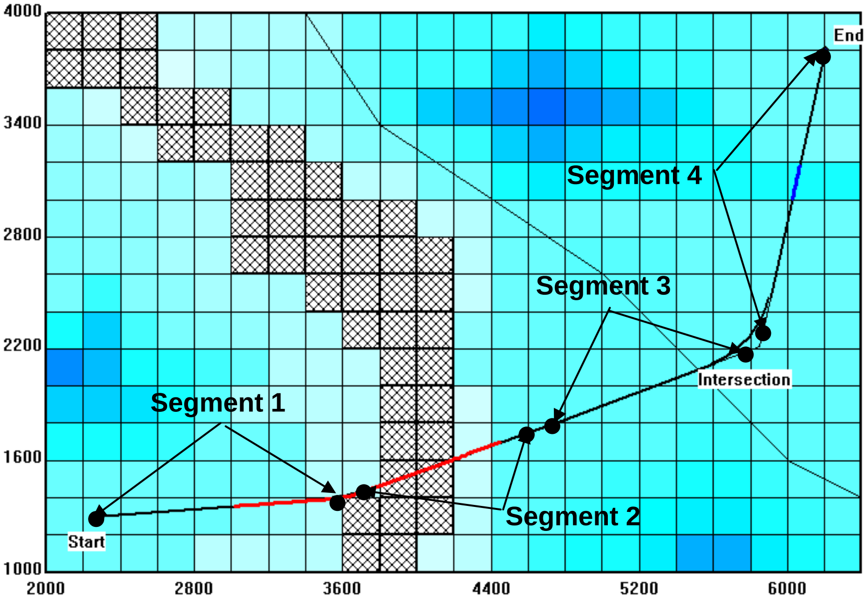
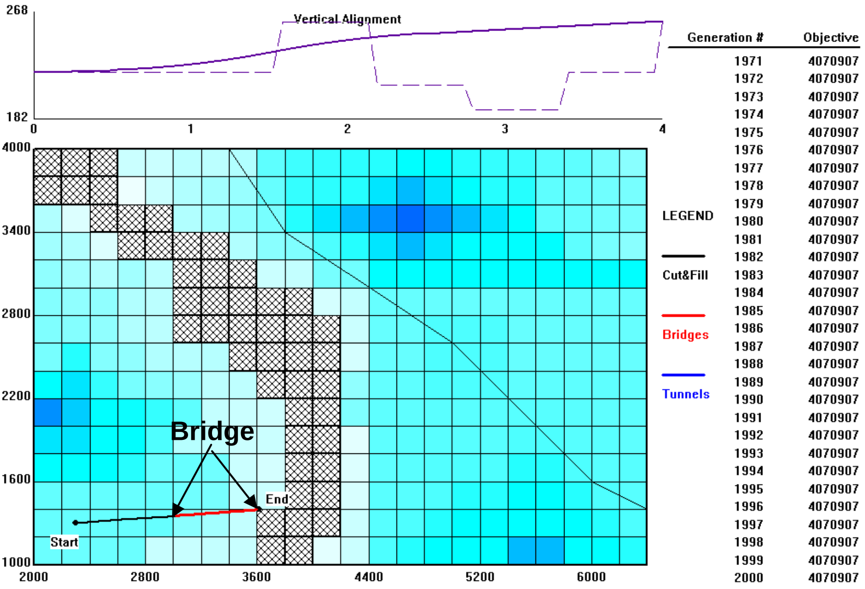


Figure 15.42: Subdividing each segment of the scenario solution 1 for a two-stage optimization.

Table 15.21: Results for each segment of scenario 3.

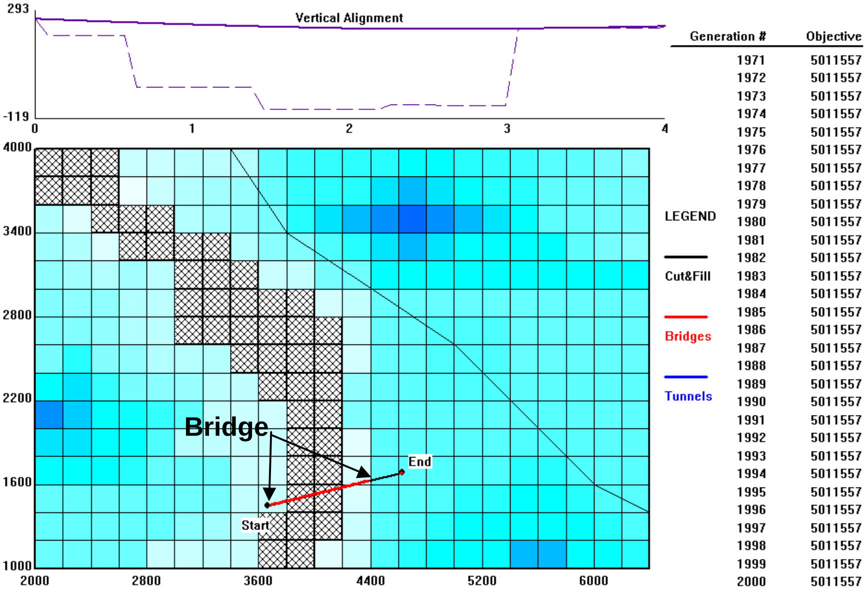
Type of optimization	Scenarios	Segments	Generation at which the optimized solution found	Total costs ($million)	Computation time
Two-stage optimization	Scenario 3	Segment 1	1985	4.07	15 seconds
		Segment 2	1905	5.01	14 seconds
		Segment 3	1998	3.69	5 minutes 48 seconds
		Segment 4	1998	3.97	17 seconds
Total	N. A.	N. A.	N. A.	16.74	6 minutes 34 seconds

Figures 15.43, 15.44, 15.45 and 15.46 show the optimized solutions for each segment with the two-stage optimization. Significant changes compared to the one-stage solution conducted in Figure 15.40 can be found in the solution of segment 4. For instance, the tunnel previously provided in Figure 15.40 has disappeared. Instead, only cuts and fills are observed in an alignment shifted to the right, which costs less than the alignment with the tunnel found in Figure 15.40.



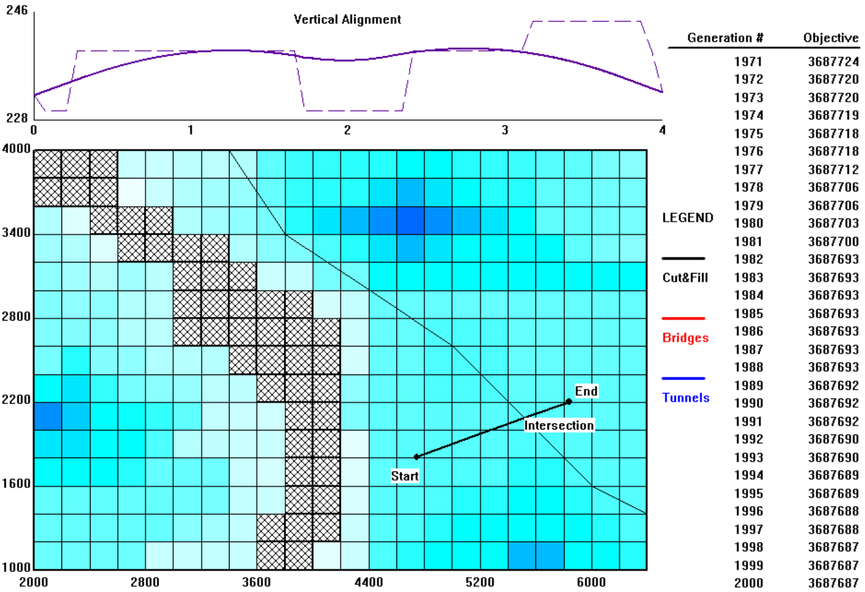
Generation #	Objective
1971	4070907
1972	4070907
1973	4070907
1974	4070907
1975	4070907
1976	4070907
1977	4070907
1978	4070907
1979	4070907
1980	4070907
1981	4070907
1982	4070907
1983	4070907
1984	4070907
1985	4070907
1986	4070907
1987	4070907
1988	4070907
1989	4070907
1990	4070907
1991	4070907
1992	4070907
1993	4070907
1994	4070907
1995	4070907
1996	4070907
1997	4070907
1998	4070907
1999	4070907
2000	4070907

Figure 15.43: Optimized solution for segment 1 under scenario 3.



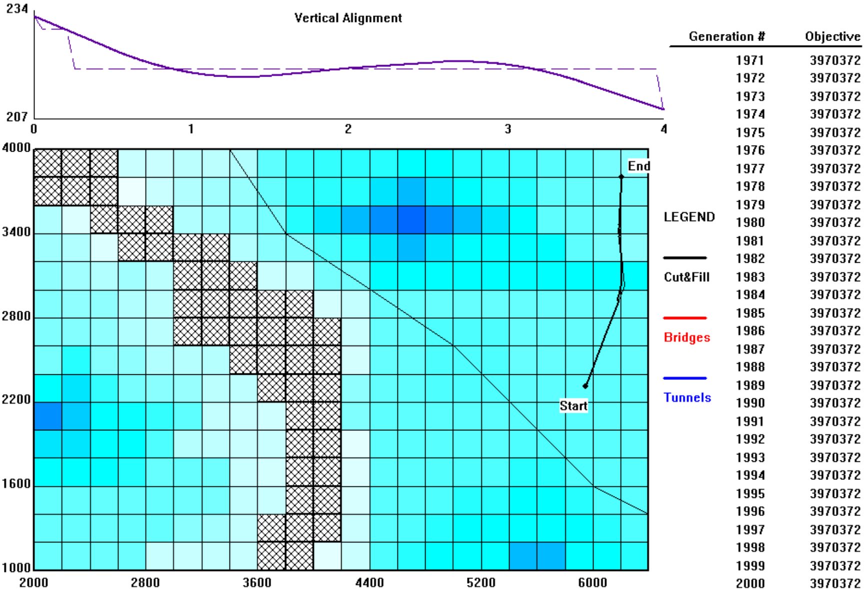
Generation #	Objective
1971	5011557
1972	5011557
1973	5011557
1974	5011557
1975	5011557
1976	5011557
1977	5011557
1978	5011557
1979	5011557
1980	5011557
1981	5011557
1982	5011557
1983	5011557
1984	5011557
1985	5011557
1986	5011557
1987	5011557
1988	5011557
1989	5011557
1990	5011557
1991	5011557
1992	5011557
1993	5011557
1994	5011557
1995	5011557
1996	5011557
1997	5011557
1998	5011557
1999	5011557
2000	5011557

Figure 15.44: Optimized solution for segment 2 under scenario 3.



Generation #	Objective
1971	3687724
1972	3687720
1973	3687720
1974	3687719
1975	3687718
1976	3687718
1977	3687712
1978	3687706
1979	3687706
1980	3687703
1981	3687700
1982	3687693
1983	3687693
1984	3687693
1985	3687693
1986	3687693
1987	3687693
1988	3687693
1989	3687692
1990	3687692
1991	3687692
1992	3687690
1993	3687690
1994	3687689
1995	3687689
1996	3687688
1997	3687688
1998	3687687
1999	3687687
2000	3687687

Figure 15.45: Optimized solution for segment 3 under scenario 3.



Generation #	Objective
1971	3970372
1972	3970372
1973	3970372
1974	3970372
1975	3970372
1976	3970372
1977	3970372
1978	3970372
1979	3970372
1980	3970372
1981	3970372
1982	3970372
1983	3970372
1984	3970372
1985	3970372
1986	3970372
1987	3970372
1988	3970372
1989	3970372
1990	3970372
1991	3970372
1992	3970372
1993	3970372
1994	3970372
1995	3970372
1996	3970372
1997	3970372
1998	3970372
1999	3970372
2000	3970372

Figure 15.46: Optimized solution for segment 4 under scenario 3.

From the results and analyses, it is found that a two-stage optimization produces better solutions in less time if a study area is relatively large or has many sensitive properties or requires complex structures such as intersections, bridges and tunnels.

To subdivide a study area, it is recommended that a one-stage optimization be run with a relatively small number of decision variables (PI's). Then the relevant PIs location for subdivision should be selected based on (1) the possibility for construction of structures and (2) the necessity of more precise evaluation. The lengths of segments may differ.

15.5 Sensitivity analysis of critical parameters

The sensitivity of critical parameters is analyzed in this section to check if different starting points of the search process yield closely similar solutions at the end. If the proposed methods for modeling intersections and other structures are suitably developed and the search algorithms are applied properly, similar solutions should be expected, regardless of starting points.

Many parameters are embedded in the genetic algorithms used for this study. Among them, critical parameters affecting search performance include the number of genetic operators, the number of decision variables (i.e., points of intersections, PI's), the parameter for selective pressure and the parameter for non-uniform mutation.

The problem specific genetic operators used in this study are four mutation operators and four crossover operators: (1) uniform mutation, (2) straight mutation, (3) non-uniform mutation, (4) whole non-uniform mutation, (5) simple crossover, (6) two-point crossover, (7) arithmetic crossover and (8) heuristic crossover.

In the computerized program, users can choose the employment frequency for each operator. For example, the simple crossover operator can be used once or twice or four times in each selection procedure to generate the population of highway alignment alternatives. Thus, it is desirable to check how the different numbers of each operator employed in selection procedures affect solutions and computation time.

The number of decision variables also affects search performance and computational efficiency, as discussed in the previous section. In addition, the effects of the number of decision variables combined with other parameters are further examined in this section.

Another important parameter is the coefficient of selective pressure. Within genetic algorithms, there is a step called selection/replacement (or referred sampling mechanism) for generating the population of successive generations. In the selection/replacement procedure, a probability density function is needed to calculate each alternative's chance of surviving to the following generation. In this book, the following nonlinear function of a user-defined parameter, q, proposed by Michalewicz (1996) is adapted.

$$p_k = cq(1-q)^{k-1}, \tag{15.1}$$

where p_k = the selection probability for the k^{th} chromosome in the ranking of the population,
c = coefficient
q = user defined parameter, $q \in (0,1)$.

It can be seen from eqn (15.1) that the user-defined parameter q is the selective pressure. A large value of q implies stronger selective pressure by imposing a higher selection probability on the good solutions. Conversely, with lower values of q, genetic algorithms may converge slowly but explore the search space more thoroughly.

The last parameter to be checked is the coefficient for non-uniform mutation operators. The non-uniform mutation operator was introduced by Michalewicz (1996) and called dynamic mutation (Gen and Cheng, 1997).

At early generations the mutation range for decision variables is relatively large, while the mutation is limited to a small range for fine-tuning the solution at later generations (refer to Figure 15.47). This operator uses a function of t (current generation number) and y (mutation range):

$$f(t,y) = y \times r_c[0,1] \times \left(1 - \frac{t}{n_T}\right)^{\xi}, \tag{15.2}$$

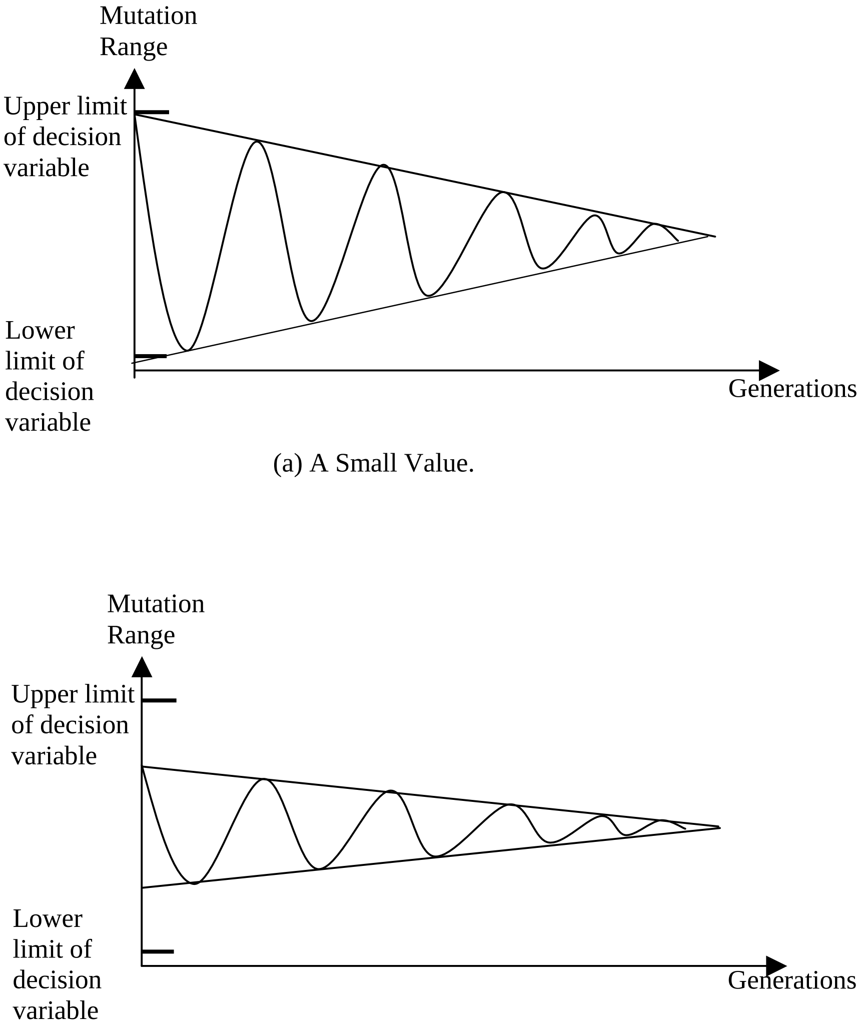
where n_T = the maximal generation number
ξ = a user defined parameter which determines the degree of non-uniformity
$r_c[0,1]$ = random number generated from a continuous uniform distribution

In eqn (15.2), a small user defined parameter, ξ, implies a wide range of mutation while a large value causes narrow range of mutation even if it is common that probability of $f(t,y)$ approaching 0 increases as t increases.

Figure 15.47 illustrates the concept of how this parameter mutates a decision variable through successive generations.

Table 15.22 shows 16 scenarios to be evaluated using these four critical parameters.

In fact, there are enormous numbers of possible scenarios. Furthermore, each scenario needs a sufficiently large number of replications to be checked thoroughly. In this section, only two distinct sets or values for each parameter are carefully selected and evaluated: (1) 6 and 7 for the parameter of non-uniform mutation, (2) 0.1 and 0.2 for the parameter of selective pressure, (3) 10 and 15 for the number of PI's and (4) two different combination sets of genetic operators. The first set of genetic operators is used in scenarios 1 to 9 and the second is used in scenarios 9 to 16. The first set uses more mutation operators while the second uses more crossover operators, as shown in Table 15.22. The example study area used in the previous section 15.4 is again selected for this section. (Scenario 9 is the one which was used in the previous section.)


Mutation Range
Upper limit of decision variable
Lower limit of decision variable
Generations
Mutation Range
Upper limit of decision variable
Lower limit of decision variable
Generations

(a) A Small Value.

(b) A Large Value.

Figure 15.47: Effects of the non-uniform mutation parameter.

Table 15.22: Scenarios for sensitivity analysis of critical parameters.

Scenarios	**Mutation operators**				**Crossover operators**				**No. of PI's**	**Selective pressure,** q	**Parameter for non-uniform mutation,** ξ
	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**	**(8)**			
S1	4	4	4	4	2	2	2	2	15	0.1	6
S2	4	4	4	4	2	2	2	2	15	0.1	7
S3	4	4	4	4	2	2	2	2	15	0.2	6
S4	4	4	4	4	2	2	2	2	15	0.2	7
S5	4	4	4	4	2	2	2	2	10	0.1	6
S6	4	4	4	4	2	2	2	2	10	0.1	7
S7	4	4	4	4	2	2	2	2	10	0.2	6
S8	4	4	4	4	2	2	2	2	10	0.2	7
S9	2	2	2	2	4	4	4	4	15	0.1	6
S10	2	2	2	2	4	4	4	4	15	0.1	7
S11	2	2	2	2	4	4	4	4	15	0.2	6
S12	2	2	2	2	4	4	4	4	15	0.2	7
S13	2	2	2	2	4	4	4	4	10	0.1	6
S14	2	2	2	2	4	4	4	4	10	0.1	7
S15	2	2	2	2	4	4	4	4	10	0.2	6
S16	2	2	2	2	4	4	4	4	10	0.2	7

where (1) = uniform mutation
(2) = straight mutation
(3) = non-uniform mutation
(4) = whole non-uniform mutation
(5) = simple crossover
(6) = two-point crossover
(7) = arithmetic crossover
(8) heuristic crossover.

For each scenario, 2000 generations are run using a common random number method to reduce variances among scenarios. Table 15.23 and Figure 15.48 shows the results. The best solution, whose total costs are $18.016 million, is found under scenario 3 while the worst solution, which costs $23.795 million, is found under scenario 1. The mean and sample standard deviation for the total costs are found to be $21.849 million and $1.331 million, respectively. It is notable that the sample standard deviation is relatively small, indicating the obtained solutions are fairly close. This result implies that the search process is converging toward a local optimum (at least).

Figures 15.49, 15.50, 15.51 and 15.52 show the best, second best, third best and worst among the 16 solutions.

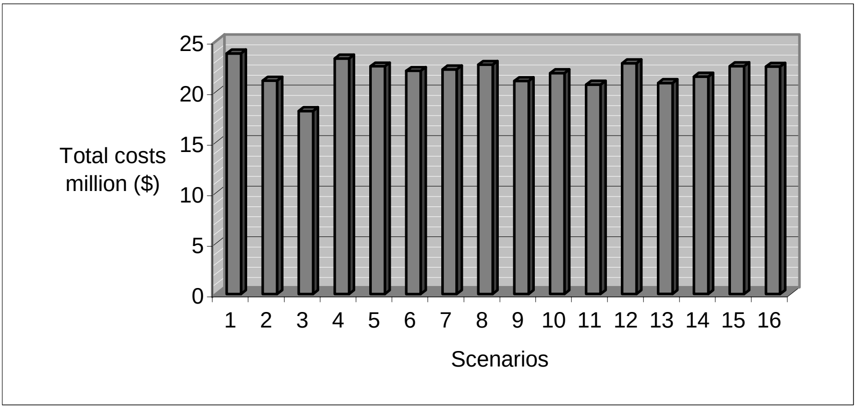
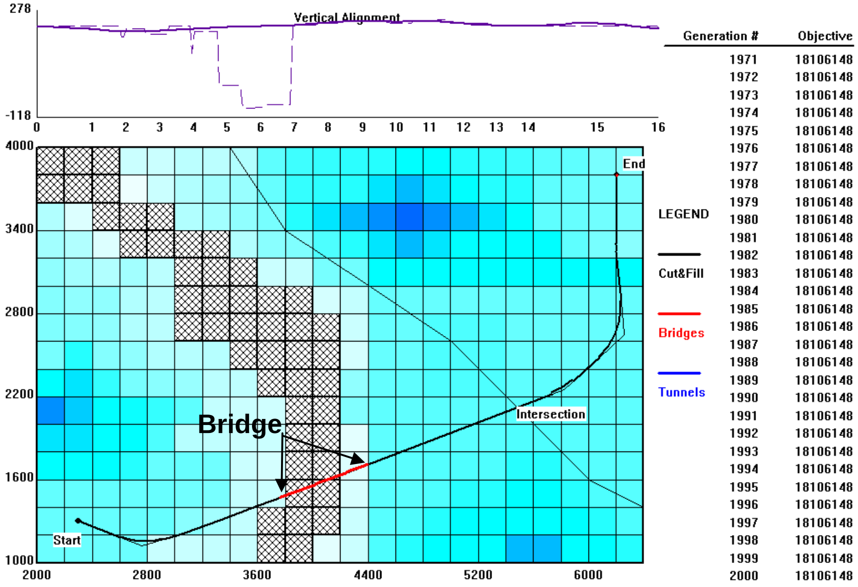


Figure 15.48: Total costs of 16 scenarios.

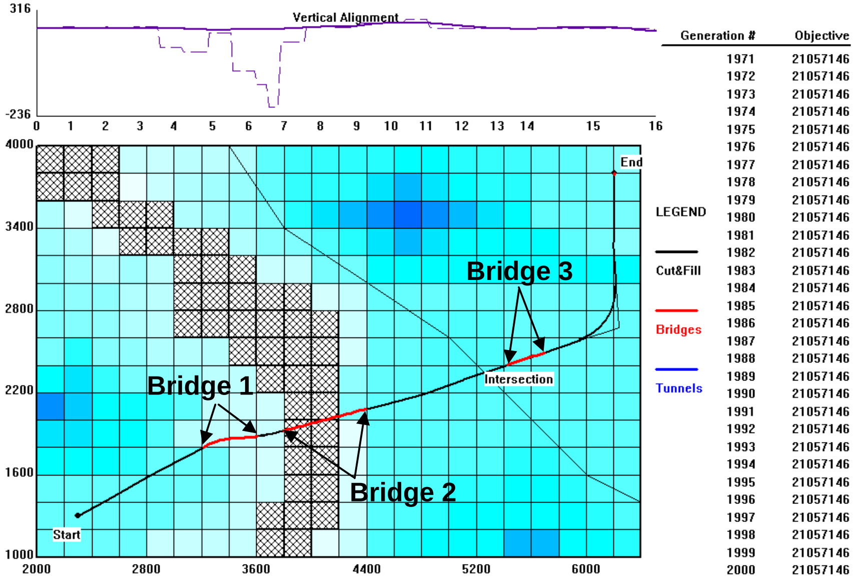
Table 15.23: Results for sensitivity analysis of critical parameters.

Scenarios	Total costs (million dollars)	Computation time (seconds)	No. of bridges	No. of tunnels	Generation no. at which the optimized solution found	Does the optimized solution involve local intersection optimization?
S1	23.795	6,473	1	1	1,992	Yes
S2	21.116	5,878	2	1	2,000	No
S3	18.106	3,886	1	0	1,948	No
S4	23.292	6,659	2	1	2,000	No
S5	22.522	3,856	1	1	1,990	No
S6	22.071	3,743	1	1	1,966	No
S7	22.221	5,616	1	1	1,998	No
S8	22.670	3,720	2	1	1,992	No
S9	21.057	1,846	3	0	2,000	No
S10	21.845	2,637	2	1	1,999	Yes
S11	20.795	1,501	1	1	1,997	No
S12	22.819	2,204	2	1	1,998	No
S13	20.862	2,479	3	1	1,999	No
S14	21.492	1,503	1	1	2,000	No
S15	22.534	2,940	2	1	1,999	Yes
S16	22.480	2,241	1	1	1,922	No
Mean	21.849					
Standard deviation	1.331					



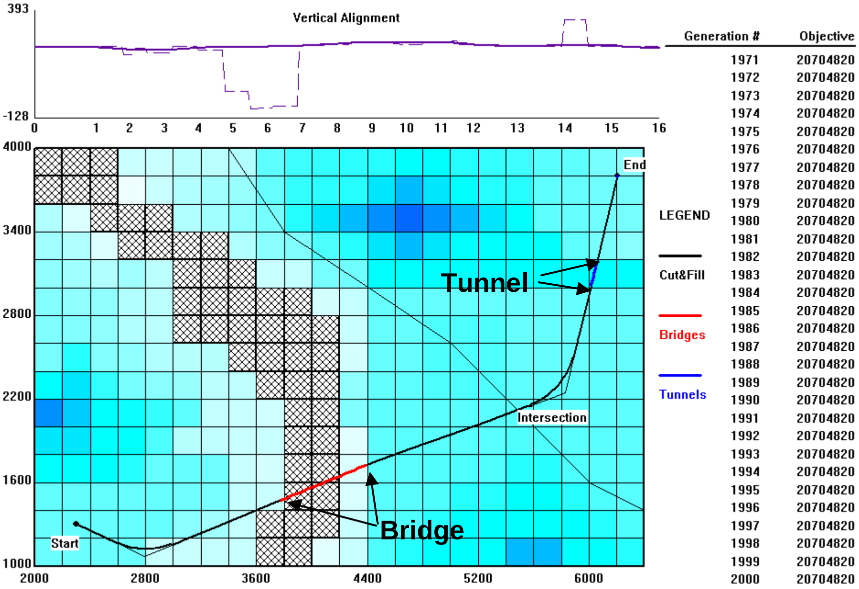
Generation #	Objective
1971	18106148
1972	18106148
1973	18106148
1974	18106148
1975	18106148
1976	18106148
1977	18106148
1978	18106148
1979	18106148
1980	18106148
1981	18106148
1982	18106148
1983	18106148
1984	18106148
1985	18106148
1986	18106148
1987	18106148
1988	18106148
1989	18106148
1990	18106148
1991	18106148
1992	18106148
1993	18106148
1994	18106148
1995	18106148
1996	18106148
1997	18106148
1998	18106148
1999	18106148
2000	18106148

Figure 15.49: The best solution (scenario 3).



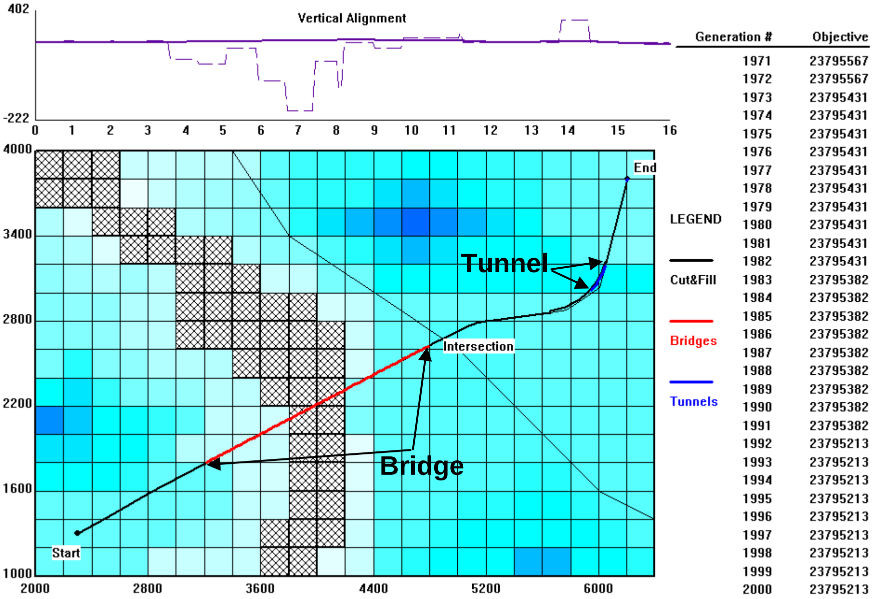
Generation #	Objective
1971	21057146
1972	21057146
1973	21057146
1974	21057146
1975	21057146
1976	21057146
1977	21057146
1978	21057146
1979	21057146
1980	21057146
1981	21057146
1982	21057146
1983	21057146
1984	21057146
1985	21057146
1986	21057146
1987	21057146
1988	21057146
1989	21057146
1990	21057146
1991	21057146
1992	21057146
1993	21057146
1994	21057146
1995	21057146
1996	21057146
1997	21057146
1998	21057146
1999	21057146
2000	21057146

Figure 15.50: The second best solution (scenario 9).



Generation #	Objective
1971	20704820
1972	20704820
1973	20704820
1974	20704820
1975	20704820
1976	20704820
1977	20704820
1978	20704820
1979	20704820
1980	20704820
1981	20704820
1982	20704820
1983	20704820
1984	20704820
1985	20704820
1986	20704820
1987	20704820
1988	20704820
1989	20704820
1990	20704820
1991	20704820
1992	20704820
1993	20704820
1994	20704820
1995	20704820
1996	20704820
1997	20704820
1998	20704820
1999	20704820
2000	20704820

Figure 15.51: The third best solution (scenario 11).



Generation #	Objective
1971	23795567
1972	23795567
1973	23795431
1974	23795431
1975	23795431
1976	23795431
1977	23795431
1978	23795431
1979	23795431
1980	23795431
1981	23795431
1982	23795431
1983	23795382
1984	23795382
1985	23795382
1986	23795382
1987	23795382
1988	23795382
1989	23795382
1990	23795382
1991	23795382
1992	23795213
1993	23795213
1994	23795213
1995	23795213
1996	23795213
1997	23795213
1998	23795213
1999	23795213
2000	23795213

Figure 15.52: The worst solution (scenario 1).

As expected, since mutation operators normally require more computation time, the mean computation time for the first combination set of operators (scenarios 1 through 8) is approximately 1 hour 23 minutes, which is 2.2 times larger than that (36 minutes) of the second combination set (scenarios 9 through 16). However, not much improvement in the objective functions is found for the first set. No clear sensitivity to changes in PI's from 10 to 15 is seen in this example. However, it is found that the results with 15 PI's vary more than those with 10 PI's. This is reasonable since with more PI's, more alternatives would be evaluated within the search process. Not much sensitivity is found with different values for either the selective pressure parameter or the parameter for non-uniform mutation operators.

It is clearly found that if there are many alignment alternatives involving local intersection optimization through successive generations, more computations are required. For instance, within the second combination set of operators (scenarios 9 through 16), the computation times of approximately 45 minutes for scenarios 10 and 15, which involve local intersection optimization, are approximately 1.5 times those of the others. With these analyses, we can judge with fair confidence where the global optimum might be located from Figures 15.49 to 15.51 and how much it costs from Table 15.23 and Figure 15.48.

In conclusion, it can be stated that the proposed methods for modeling intersections and other structures and the search algorithms perform well based on how closely the solutions from different starting points converge, although there is no rigorous way to prove that the best solution found under scenario 3 is the global optimum.

Chapter 16

Future work

The preceding chapters presented the advances towards intelligent road design by our research team. This chapter presents several issues that should be addressed in the future. These are discussed in the next sections.

16.1 Other artificial intelligence techniques for optimal search

In this book we developed genetic algorithms for optimizing highway alignments. Those algorithms seemed to perform well. However, for larger problem size computing efficiency seemed to be a concern due to large number of spatial manipulations required in the Geographic Information System (GIS) associated with right-of-way and environmental cost computations. Kim *et al* (2005) developed a stepwise genetic algorithms to ease the computational burden. Jha (2002) also developed formulations using swarm intelligence to solve the highway alignment optimization problem. More work is required to test swarm intelligence and compare the results with those obtained with genetic algorithms. Other artificial intelligence algorithms, such as simulated annealing or artificial neural networks may also be explored.

16.2 Optimization of networks

In this book we have only considered optimizing a single highway alignment between a given pair of end points. In other situations a network of highways has to be optimized. The traditional highway network design problems consider a subset of alignment-sensitive costs. In the proposed approach the highway alignments connecting the cities (see Figure 16.1) are optimized based on minimization of a comprehensive set of costs that are sensitive and significant to alignment selection.

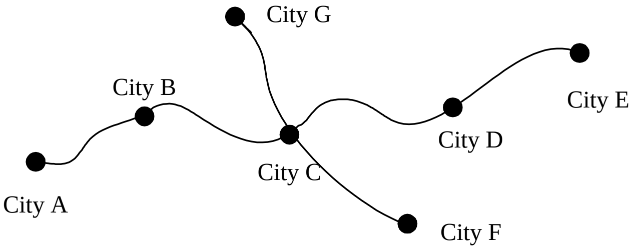


Figure 16.1: The highway network optimization problem.

Let $C = \{1,2,...,n\}$ be the set of cities to be connected (see, an example of connecting cities $A,.....,G$ in Figure 16.1). Then any two cities (i, j) can be randomly picked to be connected and an optimized highway alignment can be obtained between those cities using the GA-GIS models developed earlier. The optimized objective function thus obtained is the optimal cost C^*_{ij} of connecting cities (i, j). The corresponding optimal alignment is L^*_{ij}. Next, another city k can be randomly chosen from set C to which city j can be connected and the corresponding optimal alignment and cost can be called L^*_{jk} and C^*_{jk}, respectively. Thus, the objective function for the network optimization problem can be expressed as:

$$\text{Minimize} \underset{1,2,.....,N}{C_T} = \sum_{i,j} C^*_{ij} \, . \tag{16.1}$$

Please note that C^*_{ij} ,s are obtained by applying our previously developed genetic algorithm-GIS optimization model. In that model the objective function C_{ij} is expressed as a sum of user, operator, and penalty costs (Equation (16.2)). User cost consists of accident cost, travel-time delay cost, and vehicle operating cost. Operator cost consists of pavement and construction cost, earthwork cost, and right-of-way costs. Penalties are imposed for violation of minimum length of vertical curves and maximum curvature constraint. Penalties are also imposed for the environmental damage (such as damage to floodplain and wetland) in proportion to the damage.

$$\underset{x_{P_1}, y_{P_1}, z_{P_1}, ..., x_{P_n}, y_{P_n}, z_{P_n}}{\text{Minize}} \quad C_{ij} = C_O + C_U + C_P \, , \tag{16.2}$$

$$\text{subject to:} \quad x_L \le x_{P_i} \le x_U \, , \qquad \forall \, i = 1,....., n \, , \tag{16.3}$$

$$y_L \le y_{P_i} \le y_U \, , \qquad \forall \, i = 1,....., n \, , \tag{16.4}$$

$$z_L \le z_{P_i} \le z_U \, , \qquad \forall \, i = 1,....., n \, . \tag{16.5}$$

In eqn. (16.2), C_O , C_U , and C_P are operator, user, and penalty costs, (x_L, y_L, z_L) and (x_U, y_U, z_U) are lower and upper limits on the decision variables

$(x_{P_i}, y_{P_i}, z_{P_i})$'s. Eqns (16.3) to (16.5) impose upper and lower limits on the decision variables. The decision variables are the coordinates of points of intersections based on the concept of orthogonal cutting planes, which is extensively discussed in previously published works. Additional work is required in the future to extend search algorithms to network optimization.

16.3 GIS issues

The GIS offers a number of challenges. First, a digital GIS map is required to apply the developed algorithms. As noted in Chapter 10 in Maryland a desktop electronic property map MDPropertyView is developed that contains scanned image of property boundaries and their associated cost, area, zoning, and other relevant information. The scanned property map has to be first digitized either manually (for small scale applications) or using commercial digitizers (for large scale applications) before developed algorithms can be applied. For areas for which GIS database and map is not available aerial photos can be taken, relevant GIS data can be collected, and a digital GIS map can be created.

The second issue with GIS is accuracy of maps. Generally, right-of-way and environmental costs are computed in GIS, which depend on accuracy of maps. More work is required to develop algorithms that will improve map accuracy.

The third issue is associated with the complexity of different GIS layers that need to be superimposed for alignment optimization application. Since the developed optimization algorithm can only work with a single “final map” the relative weights of different GIS layers, such as properties, floodplains, wetlands, existing roads, and other geographic entities (parklands, schools, cemeteries, etc.) should be carefully analyzed.

16.4 Formulation of other costs

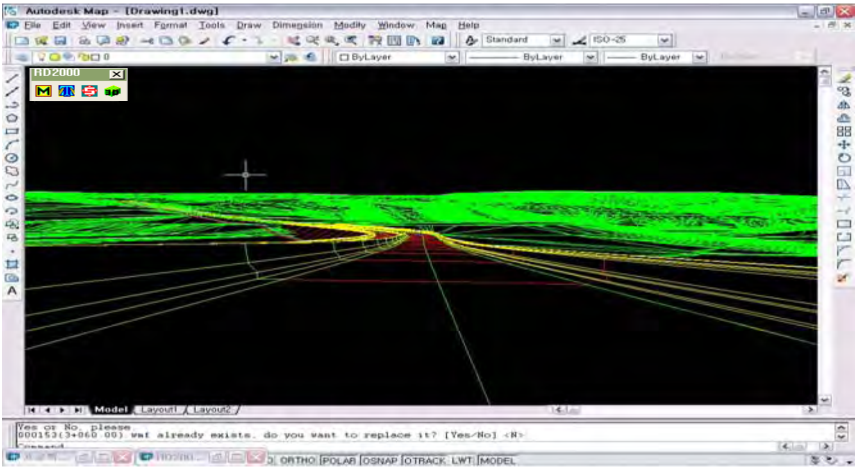
While we considered a number of alignment sensitive and significant costs in our analysis additional costs (or dollar equivalent) of noise and pollution, demand and socio-economic changes due to alignment construction, impacts of traffic congestion, and political considerations in alignment selection should be considered.

16.5 Hierarchal representation of cost components in optimization

In our analysis all costs considered are weighed equally. In reality some costs may have greater weights than others. Therefore, a hierarchy of costs considered in alignment optimization should be considered. For example, costs associated with political considerations, people’s preferences, and noise and environmental considerations will generally take precedence over traditional user and operators costs, such as earthwork and right-of-way costs.

16.6 Digital terrain modeling

Our models generally consider a digital GIS map to apply the optimization algorithm. It will be desirable to connect the optimization model with a traditional design package, such as Microstation or RD 2000 (an autoCAD-based highway design package), to perform detailed design in addition to optimization. Work is currently underway to integrate the developed optimization model with RD 2000. For detailed design a digital terrain model (DTM) needs to be generated. A snapshot of a DTM taken from RD 2000 is shown in Figure 16.2.


Autodesk Map - [Drawing1.dwg]
File Edit View Insert Format Tools Draw Dimension Modify Window Map Help
Standard
ISO-25
ByLayer
ByLayer
ByLayer
RD2000
Model
Layout1
Layout2
ORTHO POLAR OSNAP OTRACK LWT MODEL

```
Yes or No, please.
000153(3+060 00) wmf already exists. do you want to replace it? [Yes/No] <N>
Command:
```

Figure 16.2: Snapshot of a DTM taken from RD 2000.

16.7 Conceptual improvements

Several conceptual improvements may be needed in the developed alignment optimization model, which are left for future work. Some of these include:

(1) Choosing the spacing between PI's based on the complexity of the GIS map.
(2) Screening candidate alignments with genetic algorithms and only allowing GIS analysis for selected alignments, thus reducing the computation time.
(3) Multi-objective optimization instead of optimization with a single objective.
(4) Considering minimum water clearance to be maintained for bridge construction.
(5) Modeling public and political opinions as well as socio-economic considerations in highway alignment decision-making.

Appendix A

An overview of genetic algorithms

This appendix provides a brief introduction to Genetic Algorithms (GAs) so that they can be understood in this book. The principles of GAs have been well developed and are described in many texts (e.g., Gen and Cheng, 1997; Goldberg, 1989; Michalewicz, 1996). Some papers also provide an overview of Genetic Algorithms (for example, Beasley, Bull, and Martin, 1993a; 1993b). In addition, there are many World Wide Web (www) sites offering various types of information about GAs. This appendix is based on such sources, without original contributions. However, in the book, we will develop solution algorithms for optimizing highway alignments with several modifications to classical GAs. Briefly, a Genetic Algorithm is a technique for solving an optimization problem, but not all problems can be solved in the default format of GAs. For different systems, one may have to develop different solution procedures based on the philosophies and principles of GAs, and the nature of the problem. According to Michalewicz (1996), the term “Evolution Programs (EP)” is generally more suitable for problem solving by GAs.

A.1 What are genetic algorithms?

Genetic Algorithms are adaptive methods that may be used for search and optimization problems. They are motivated by the principles of natural selection and “survival of the fittest”. In 1975, Holland borrowed this idea to develop the basic framework of Genetic Algorithms. Holland’s techniques have been further developed and now GAs stand out as a powerful technique for solving complex optimization problems. Since GAs are derived from natural evolution, for better understanding, we summarize the correspondence between the terminology used in GAs and its biological counterpart in Table A.1.

Table A.1: Terminology in Genetic Algorithms and its explanations.

Genetic Algorithms	Explanations
Chromosome (genotype, individual, string)	Encoded solution
Phenotype	Decoded solution
Gene	A portion of chromosome
Locus	Position of gene in a chromosome
Alleles	Values of gene
Gene pool	The set of possible alleles
Population	A set of solutions

In nature, individuals in a population have to compete with each other for vital resources such as food or shelter. Also in the same species, individuals often have to compete to attract mates for reproduction. Due to such selection, poorly performing individuals have a lesser chance to survive and the most adapted or "fit" individuals produce a relatively larger number of offspring. After a few generations, species evolve spontaneously to become more and more adapted to their environment.

In Genetic Algorithms, the problem is treated as the environment, and a set of possible solutions to the problem is treated as the population. Each individual in the population is represented by an encoded solution called chromosome. The individuals then compete with each other to reproduce offspring according to their "fitness" (suitability) to the environment (the problem). After several generations, the most adapted individuals should survive, whereas poor solutions should die off. If a GA has been well designed, the population will converge toward an optimal solution to the problem.

We should note that in nature, individuals not only compete with others, but also struggle with their environment. In some situations, where a disaster happens to the environment, a species may become extinct. In GAs, however, the population typically survives and never becomes disappeared completely. We should also note that in natural evolution an offspring may inherit good features from its parents via gene recombination, or obtain good genes through mutation. In GAs, such recombination and mutation play key roles in the search process.

A.2 How do genetic algorithms work?

The basic structure of Genetic Algorithms is outlined as follows and illustrated in Figure A.1.

Algorithm A.1 Basic framework of genetic algorithms

```
BEGIN /* Genetic Algorithm */
    Generate initial population
    Compute fitness of each individual
```

```
WHILE NOT finished DO LOOP
    BEGIN
        1. Select individuals from old generations for mating
        2. Create offspring by applying recombination and/or mutation
           to the selected individuals
        3. Compute fitness of the new individuals
        4. Discard old individuals to make room for new chromosomes
           and insert offspring in the new generation
        5. IF population has converged
           THEN finish = TRUE
    END
END
```

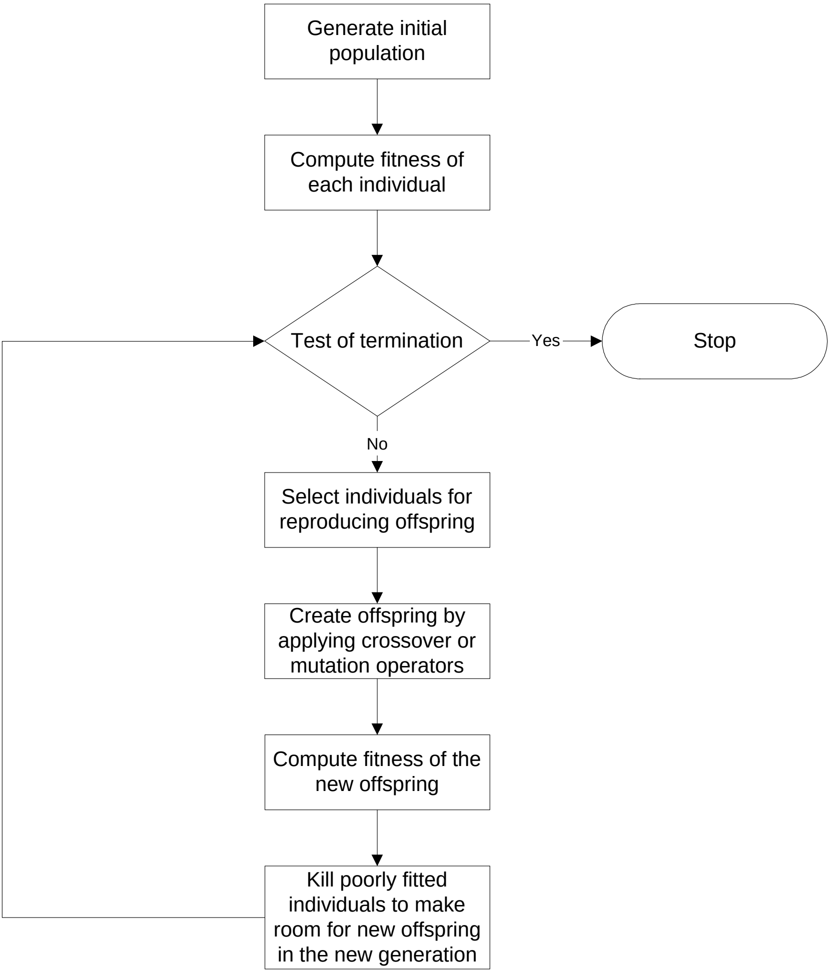

Generate initial population
Compute fitness of each individual
Test of termination
Yes
Stop
No
Select individuals for reproducing offspring
Create offspring by applying crossover or mutation operators
Compute fitness of the new offspring
Kill poorly fitted individuals to make room for new offspring in the new generation

Figure A.1: The structure of Genetic Algorithms.

The application of GAs to a specific problem includes several steps. A suitable encoding for the solution must be devised first. We also require a fitness function through which the individuals are selected to reproduce offspring by undergoing genetic operators. Each of the steps is described below:

A.2.1 Genetic encoding

To apply GAs to a specific problem, we must first devise an appropriate genetic representation for the solution. Originally, a potential solution to the problem is encoded into a string of binary digits of a given length, which is referred as a chromosome or genotype. For a problem whose solution contains floating-point numbers, we can convert each floating-point number into binary bits whose length is determined by the desired precision. Each bit in the chromosome is then referred to as a gene. Some researchers argued that a direct floating-point number or integer representation of the genes might enable us to define meaningful and problem-specific genetic operators, and achieve faster convergence. Therefore, modern GA applications are no longer restricted to a binary string for genetic encoding.

A.2.2 Fitness function

Given a particular chromosome, the fitness function returns a single value, which represents the merit of the corresponding solution to the problem. In most of the cases, the fitness function can be easily defined as the objective function that we want to optimize. However, in some problems, the determination of the fitness function may be difficult (for example in a multi-criterion optimization problem).

One GA feature is the difficulty in handling constraints. Ideally, chromosomes should be defined in such a way that only feasible solutions can be represented. However, this is not an easy task, especially for non-linear constraints. A common method to deal with constraints is to use a penalty function. This method allows constraints to be violated and the incurred penalty depends on the magnitude of the violation. The penalty function is then added to the fitness function so that a solution with greater constraint violations will yield a poorer fitness value and will rarely be selected for reproduction. This penalty method implies that a solution which carries useful information but slightly violates constraints may stay in the population for a while and eventually evolve to a better solution.

A.2.3 Selection

The individuals in the population are selected to reproduce offspring according to their fitness values. The higher the fitness function, the more chance an individual has to be selected. Typically there are two different types of selection schemes: proportionate selection and ordinal-based selection. The concept behind these two approaches is the selective pressure, which is defined as the degree to which the better individuals are favored in the selection process. A strong selective pressure may lead to premature convergence (i.e., converge to a local optimum), while a weak selective pressure tends to reduce the convergence rate of a GA.

A.2.3.1 Proportionate selection

Proportionate selection is the best-known selection scheme, which is sometimes called “roulette wheel selection”. This method is identical to the process for generating discrete random variables with Monte Carlo Simulation. It simply assigns to each solution a sector of a roulette wheel whose size is proportional to the fitness value of the solution, and then generates a random position on the wheel to determine which individual is being selected. In other words, the selection probability of each individual is proportional to its fitness value. For a maximization problem, the probability is simply the fitness value divided by the total fitness value of all individuals in the population. For a minimization problem, however, the selection probability may be difficult to determine.

It is also noted that the selective pressure in this approach is heavily dependent on the distribution of the fitness values of the population. For example, if the fitness values range within $[99990,100000]$, even the worst individual has a good change of being selected. If their fitness values are in the range $[0,10]$, then the most poorly adapted chromosomes have almost no chance of being selected. Moreover, in early stages, there may be a few super chromosomes that dominate the selection process, while in later generations when all individuals resemble each other, GAs do nothing but just a random search. To mitigate these problems, some researchers have developed scaling mechanisms to map the fitness values to positive real values, and the survival probability for each chromosome is thus determined according to these values.

A.2.3.2 Ordinal-based selection

An ordinal-based scheme is also proposed to alleviate the problem embedded in the selection process based on the raw fitness values. It selects individuals not based on their raw fitness, but on their ranks within the population. The selective pressure in this scheme is independent of the fitness distribution of the population, and solely based upon the relative ordering of the population. Also it can be easily applied to a minimization problem, where the individual with the lowest fitness value is ranked as 1. Since the selection scheme is based on the rank of the individuals, the selective pressure will be consistent throughout the entire generations.

A.2.4 Genetic operators

In classical GAs, offspring are generated from their parents by two typical types of genetic operators: mutation and crossover.

A.2.4.1 Crossover

Crossover is the process of generating offspring from two parents by swapping their genes at some randomly chosen locus of the chromosomes. The intuition behind the applicability of the operator is information exchange between potential solutions. The mechanism is similar to sexual mating in nature. The crossover operator is supposed to help in exploiting the information of the better individuals in the population.

There are many kinds of crossovers. The following figure is an illustration of single point crossover, which is the simplest crossover operator in GAs.

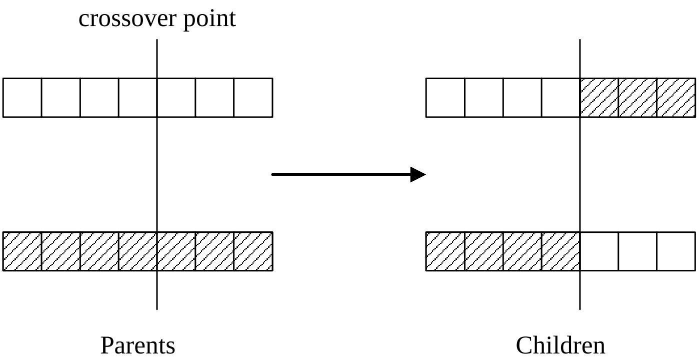


Figure A.2: A single-point crossover.

A.2.4.2 Mutation

A mutation operator arbitrarily alters one or more genes of a selected chromosome, by a random change with a probability called a mutation rate. If the rate is too low, many genes that might have been useful are never tried. On the other hand, if it is too high, there will be much random perturbation, the offspring will start losing their resemblance to the parents, and GAs will lose the ability to gain knowledge from the history of the search.

Mutation is supposed to help in exploring the entire search space, and is often seen as a background operator to maintain genetic diversity in the population. Figure A.3 is a simple example of mutation operator applying to a chromosome, where we assume each gene is represented by a binary variable.

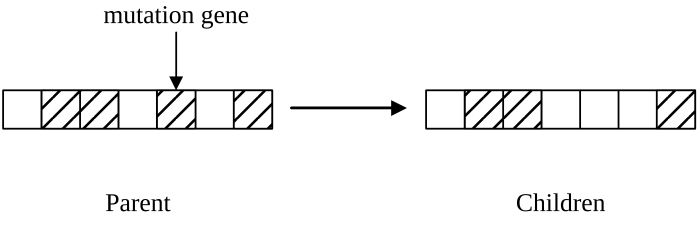


Figure A.3: A simple mutation.

A.2.5 Replacement

Once offspring are produced, we must determine which of the current members of the population should be replaced by the new offspring. Replacement is strongly related to the selection process, where we decide which of the current members of the population are going to reproduce offspring. There are many kinds of classifications of replacements. From the sampling space point of view, we can basically categorize them as either regular sampling space or enlarged sampling space.

A.2.5.1 Regular sampling space

In a regular sampling space, once an offspring is born, an individual in the current population is selected to die. The dying individual can be chosen randomly, or selected as the worst (or the oldest) member of the current population, or the one who most closely resembled the new offspring. Obviously, the sampling space of this strategy is the population size. Note that in regular sampling space, individuals in the current population can be replaced soon after their offspring are produced, or at the next generation.

A.2.5.2 Enlarged sampling space

An enlarged sampling space means that the sampling space for replacement is larger than the population size. A typical strategy is that both parents and offspring have the same chance of competing for survival. The sampling space in this case is the population size plus the size of new offspring. Another scheme is to produce a set of offspring whose size is larger than the population size. Then the first best required number (i.e., population size) of offspring are selected as parents at the next generation. Both strategies for replacements are based on generations (i.e., the offspring replace the members of the current population at the next generation).

Note that it is not guaranteed that the newly born offspring will dominate their parents, and that the best chromosome in the current generation will not be selected to die. An elitism model is thus developed for preventing the best individual from dying off. In this policy, the best chromosome is always passed on to the next generation.

A.2.6 Convergence

If a GA has been correctly implemented, the population will evolve over successive generations so that it will converge toward the global optimum. Unfortunately, GAs cannot be expected to stop spontaneously, nor guaranteed to find the global optimum. The evolution has to be stopped at some point according to a pre-determined criterion. The simplest way is to terminate the evolution after a fixed number of iterations. A better solution is to continue the iteration for as long as the improvements are noticeable. Note that if GAs stop too early or a strong selective pressure is employed, the population may converge to a local optimum. This failure is usually called premature convergence. To overcome this problem, additional generations with a higher mutation rate and lower selective pressure help in avoiding premature convergence. However, this takes too long to converge toward the optimal solution. Therefore, a good compromise is required so that GAs can find good solutions in a relatively short time.

A.3 Why do genetic algorithms work?

There is no rigorous proof that GA results converge toward the global optimum. Nevertheless, several theorems and hypotheses have been developed to theoretically explain the effectiveness of GAs. In fact, GAs work quite successfully in many practical applications.

The theoretical foundations of GAs rely on a binary string representation of solutions, and the notation of a schema. A schema (plural, schemata) is a string of symbols taken from the alphabet $\{0,1,*\}$, where * is a “don’t care” symbol, which can be 0 or 1. The number of 0’s and 1’s in a chromosome is called the order. The distance between the positions of the first and the last non-* symbol in a chromosome is called the defining length.

The schema theorem and the building block hypothesis explain the power of GAs in terms of how schemata are processed. The schema theorem states that schemata with short defining length, low order, and better fitness (called building blocks) receive exponentially increasing trials in subsequent generations of a GA. The building block hypothesis says that a GA seeks near-optimal performance through the juxtaposition of building blocks by genetic operators. Detailed descriptions of the schema theorem and building block hypothesis can be found in Goldberg (1989) and Michalewicz (1996), and will not be further discussed here.

A.4 Comparisons with other optimization techniques

A number of general techniques have been developed to solve search and optimization problems. The main differences that distinguish Genetic Algorithms from other conventional optimization techniques are:

(1) GAs use an encoded representation of the decision variables rather than the decision variables themselves
(2) GAs search from a population rather than a single point
(3) GAs only exploit the values of the objective function, and do not require other auxiliary information such as the derivatives
(4) GAs use stochastic operators, rather than deterministic rules

Detailed comparisons between GAs and other optimization techniques have been presented in many textbooks and papers listed in the references. The disadvantages and limitations of some alternative optimization techniques are summarized in Table A.2.

An efficient optimization algorithm must satisfy two requirements for finding solutions very near the global optimum: exploring the search space efficiently and exploiting the knowledge gained at the previously visited points. Unlike other optimization methods, which have either one of the above features, GAs use both exploration and exploitation for search. Moreover, GAs work on a population rather than a single point, and use probabilistic transition rules. These properties give GAs a great "potential" for finding the global optimum.

Note that at each generation, GAs evaluate the fitness function several times, depending on crossover and mutation rates. This implies that GAs take longer than other traditional direct search methods at each iteration. However, for a noisy objective function, GAs generally outperform traditional direct search methods because GAs not only exploit the knowledge from the previous search, but also explore the search space. It is also noted that GAs are not guaranteed to find the global optimum solution to a problem. They are satisfied with finding "acceptably good" solutions to problems "relatively quickly", especially when the objective function is quite noisy. It is also important to mention that GAs do not always dominate other optimization methods in all aspects. For example if the objective function is strictly convex and differentiable, then calculus-based search may be more appropriate to the problem. While GAs are general tools, for different problems, we may have to devise an appropriate genetic encoding of the decision variables as well as genetic operators that are suitable for the problem structure. These are generally not easy tasks and may require many trials.

Table A.2: Disadvantages and limitations of some optimization techniques.

Method	Limitations and Disadvantages
Calculus-based search & gradient search	- Require differentiable objective function - Tendency to get trapped in local optimum
Enumeration	- Inefficient
Random search	- Inefficient
Simulated annealing	- Only deals with a single point - Make no use of the knowledge from previous moves
Dynamic programming	- Requires independence between subproblems

This page intentionally left blank

Appendix B

Overview of geographic information system applications in transportation

A Geographic Information System (GIS) can have a significant role in transportation because a GIS can help capture, store, analyze, and display geographical information based on its location character. According to Hanson (1995), transportation-related legislation such as the Intermodal Surface Transportation Efficiency Act of 1991 (ISTEA), the Clean Air Act Amendments of 1990 (CAAA), and the Americans with Disabilities Act of 1990 (ADA) all encourage a more systematic and comprehensive approach to transportation planning, management, and implementation. A GIS is particularly useful in transportation since it is an effective way to integrate the information needed to support the many facets of transportation analysis.

A GIS for transportation, usually known as GIS-T has four dimensions (Hanson, 1995):

(1) Purpose: address complex tasks in policy making, planning, design, construction, maintenance, management, operations, and evaluation activities.

(2) Functional capabilities: provide database management for extending human memory, spatial analysis for rigorous computation, and map display for visualization of large amounts of information.

(3) The implied work flow for a GIS analyst: expedite a sequence of work activities involving problem scoping, data collection/storage, analysis, and display (plus iteration).

(4) The impact of a GIS on organizational/institutional arrangements: promote linkages within and between organizations that enable or constrain the work flow for effecting desired goals.

Since a highway alignment optimization model relies on various geographic data such as topography, soil conditions, floodplains, wetlands, area and unit costs of land parcels, location of structures, type of land uses, and location of lakes and streams, it can greatly benefit from a GIS. Furthermore, the spatial characteristics of a GIS can be exploited to analyze multiple effects of geographic characteristics. For example, effects of soil conditions and wetlands can be simultaneously examined. The spatial features can also help estimate socio-economic costs.

There are no studies available in which a GIS is used for highway alignment optimization. The first phase in any highway location study is the examination of all available data of the area in which the road is to be constructed. These data can be obtained from a good GIS. The type and amount of data collected and examined depend on the type of highway being considered, but in general, data should be obtained on the following characteristics of the area:

- Engineering, including topography, geology, climate, and traffic volumes
- Social and demographic, including land use and zoning patterns
- Environmental, including types of wildlife; location of recreational, historic, and archeological sites; and the possible effects of air, noise, and water pollution
- Economic, including unit costs for construction and the trend of agricultural, commercial, and industrial activities

In Maryland a GIS-based desktop electronic property map called MDProperty View was created by the office of planning (Lettre *et al*, 1997) in which scanned property maps are stored with their unit cost, area, and zoning type. In addition, the Maryland State Highway Administration has GIS layers available for soil characteristics, 100 and 500-year floodplains, tidal and non-tidal wetlands, watersheds, streams, and existing roads, which can be used in highway alignment optimization. One of the deficiencies of MDProperty View maps is that they are scanned images and the boundaries of geographic features are represented by simple lines. This may require converting the maps to digital format before they can be used for spatial computations.

The *spatial characteristics* of *geographic entities* stored in a GIS allow precise calculations of areas or volumes (Laurini and Thompson, 1992). Spatial characteristics are conditions that use positional information, such as shape, areal extent, or volume. A geographic entity called a phenomenon cannot be subdivided into similar sub-units. For example, a house is not divisible into houses, but can be split into rooms. An *attribute* is a defined characteristic of an entity, such as the type of a bridge, the size of a lake, or the wetness of soil.

One of the challenges in highway alignment optimization problems is to extract GIS data at program run-time, which can be used by the optimization algorithm. Most of the GIS applications in the literature discuss various facets of GIS without considering the automation necessary for alignment optimization applications. Simkowitz (1989) discussed the data handling capacity, mapping of data, and tying the data to a uniform geographical coordinate system.

Applications in FHWA and among state agencies were considered. While the study was a good application of the spatial capabilities of GIS, the process was purely manual and of no benefit to highway alignment optimization problems. Abkowitz *et al* (1990) discussed the application of GIS in highway management. The focus of the study was to establish guidelines that would help highway agencies to adapt the way in which they collect, store, and utilize data as well as undergo fundamental changes in how they carry out technical responsibilities. The study presented a good GIS application with greater promise for future applications in transportation policy analyses. Several other interesting GIS-T applications were also reported (Evans *et al*, 1993; Hammad *et al*, 1993; Warwick and Haness, 1993; Lamm *et al*, 1994; Sarasua, 1994; Lee and Clover, 1995; Bartlett, 1996; Olivera and Maidment, 1998; Versenyi *et al*, 1994; Jourquin and Beuthe, 1996; Miller and Storm, 1996; and Vandermark and Corbley, 1996). The existing GIS studies in Transportation can thus be classified into the following two categories, which are shown in Table B.1:

- Those exploiting spatial characteristics and database management; and
- Those building integrated or expert systems by interfacing with other applications.

A great advantage of integrating GIS with expert systems or other models is that some otherwise very laborious tasks can be automated. For example, some Genetic Algorithms (GAs) are very good search tools that can be applied to a complex and continuous search space to obtain minimum cost highway alternatives. For practical applications however, the input data such as land cost and terrain elevation should be made available in an automated fashion during the optimization process. A recent study (Gilbrook, 1999) applied a GIS to obtain an optimum roadway corridor for highway construction. The method reported was based on overlaying a number of GIS features to arrive at the optimum corridor. However, selection of the optimum corridor was based on simple averaging and did not involve any mathematical optimization.

Table B.1: Classification of GIS-T Applications.

Application Type	**Studies**
Spatial representation and database management	Simkowitz (1989), Abkowitz *et al* (1990), Evans *et al* (1993), Hammad *et al* (1993), Warwick and Haness (1993), Lamm *et al* (1994), Sarasua (1994), Lee and Clover (1995), Bartlett (1996), Olivera and Maidment (1998)
Building expert systems by interfacing with other applications	Evans *et al* (1993), Sarasua (1994), Lee and Clover (1995), Bartlett (1996)

A GIS can be exploited to develop algorithms based on the spatial relations. For highway design optimization, such algorithms may be developed to precisely compute right-of-way costs. This was not found in the literature. Additionally, environmental analysis using a GIS can also be performed and incorporated in highway design optimization. Recent applications of GIS in alignment optimization can be found in Jha and Schonfeld (2000a&b, 2003), Jha (2001, 2002, and 2003), Jha *et al* (2001), Kim *et al* (2001, 2004a&b), and Jha and Schonfeld (2004).

Bibliography

[1] AASHTO, *Acquisition for Right-of-Way*, American Association of State Highway and Transportation Officials, 1962.

[2] AASHTO, *Roadside Design Guide*, American Association of State Highway and Transportation Officials, 1988.

[3] AASHTO, *A Manual on User Benefit Analysis of Highway and Bus-Transit Improvements*, American Association of State Highway and Transportation Officials, Washington, D. C., 1977.

[4] AASHTO, *A Policy on Geometric Design of Highways and Streets*, American Association of State Highway and Transportation Officials, Washington, D. C., 2001.

[5] AASHTO, *AASHTO Maintenance Manual*, American Association of State Highway and Transportation Officials, Washington, D. C., 1987.

[6] Abd El Halim, A. O., Easa, S. M., and Hassan, Y., Highway Alignment: Three-Dimensional Problem and Three-Dimensional Solution, *Transportation Research Record* 1612, TRB, National Research Council, Washington, D.C., pp. 17-25, 1998.

[7] Abkowitz, M., Walsh, S., Hauser, E., and Minor, L., Adaptation of Geographic Information Systems to Highway Management, *ASEC Journal of Transportation Engineering,* Vol. 116, No. 3, pp. 310-327, 1990.

[8] Akcelik, R., Bayley, C., Bowyer, D. P., and Biggs, D. C., A Hierarchy of Vehicle Fuel Consumption Models, *Traffic Engineering and Control,* Vol. 24, No. 10, pp 491-495, 1983.

[9] Ashford, N. and Wright, P. H., *Airport Engineering*, 3rd Edition, John Wiley & Sons, Inc., 1962.

[10] Athanassoulis, G. C. and Calogero, V., Optimal Location of a New Highway from A to B – A Computer Technique for Route Planning, *In PTRC Seminar Proceedings on Cost Models and Optimisation in Highways (Session L9)*, London, 1973.

[11] Barker, R. M. and Puckett, J. A., *Design of Highway Bridges*, John Wiley & Sons, 1997.

[12] Barnett, J., *Highway Intersections, Design and Application*. American Association of State Highway and Transportation Officials, pp. 26-45, 1939.

[13] Barrett, D. E., Traffic-Noise Impact Study for Least Bell's Vireo Habitat Along California State Route 83, *Transportation Research Record* 1559, TRB, National Research Council, Washington, D.C., pp. 3-7, 1996.

[14] Bartlett, R.E., GIS-CAD and the New Urban Planning Universe, *ITE Journal*, Vol. 66, No. 1, pp. 47-49, 1996.

[15] Bauer, C. S., Some Energy Considerations in Traffic Signal Timing, *Traffic Engineering*, Vol. 45, No. 2, pp. 19-25, 1975.

[16] Beasley, D., Bull, D. R., Martin, R. R., An Overview of Genetic Algorithms: Part 1, Fundamentals, *University Computing*, 15, pp. 58-69, 1993a.

[17] Beasley, D., Bull, D. R., Martin, R. R., An Overview of Genetic Algorithms: Part 2, Research Topics, *University Computing*, 15, pp. 170-181, 1993b.

[18] Ben-Akiva, M., Hirsh, M., and Prashker, J., Probabilistic and Economic Factors in Highway Geometric Design, *Transportation Science*, Vol. 19, No. 1, pp. 38-57, 1985.

[19] Bendelius, A. G., Tunnel Ventilation, Chapter 20 of *Tunnel Engineering Handbook*, 2nd Edition by Bickel, J. O., Kuesel, T. R. and King, E. H., An International Thompson Publishing Company, 1996.

[20] Biggs, D. C., *ARFCOM-Models for Estimating Light to Heavy Vehicle Fuel Consumption*, Research Report ARR 152, Australian Road Research Board, 1988.

[21] Boubel, R. W., D. L. Fox, D. B. Turner, and A. C. Stern, *Fundamentals of Air Pollution*, 3rd Edition, Academic Press, 1994.

[22] Bowyer, D. P., Akcelik, R., and Biggs, D. C., Fuel Consumption Analyses for Urban Traffic Management, *ITE Journal*, Vol. 56, No. 12, pp. 31-34, 1986.

[23] Bureau of Economic Analysis (BEA) Homepage, *National Income and Product Accounts*, August 8, 2001, http://www.bea.doc.gov/bea/dn1.htm

[24] Chapra, S. C. and Canale, R. P., *Numerical Methods for Engineers*, McGraw-Hall, Inc., New York, 1988.

[25] Chesher, A. and Harrison, R., *Vehicle Operating Costs — Evidence from Developing Countries*, The International Bank for Reconstruction and Development / The World Bank, Washington, D.C., 1987.

[26] Chew, E. P., Goh, C. J., and Fwa, T. F., Simultaneous Optimization of Horizontal and Vertical Alignments for Highways, *Transportation Research Part B*, Vol. 23B. No. 5, pp.315-329, 1989.

[27] Cormen, T. H., Leiserson, C. E., and Rivest, R. L., *Introduction to Algorithms*, 6th Printing, The Massachusetts Institute of Technology Press, Cambridge, Massachusetts, 1996.

[28] Courage, K. G. and Parapar, S. M., Delay and Fuel Consumption at Traffic Signals, *Traffic Engineering*, Vol. 45, No. 11, pp. 23-27, 1975.

[29] DeCorla-Souza, P. and Jensen-Fisher, R., Comparing Multimodal Alternatives in Major Travel Corridors, *Transportation Research Record* 1429, TRB, National Research Council, Washington, D.C., pp. 15-23, 1994.

[30] Easa, S. M., Model for Sight-Distance Analysis of Uncontrolled Intersections, *ASEC Journal of Transportation Engineering*, Vol. 124, Issue 2, pp 156-162, 1998.

[31] Easa, S. M. and Hassan, Y., Development of Transitioned Vertical Curve I Properties, *Transportation Research Part A*, Vol. 34, pp. 481-496, 2000.

[32] Easa, S. M., Selection of Roadway Grades that Minimize Earthwork Cost Using Linear Programming, *Transportation Research Part A*, Vol. 22A, No. 2, pp. 121-136, 1988.

[33] Easa, S.M., Halim, A.O., and Hassan, Y., Sight Distance Evaluation on Complex Highway Vertical Alignments, *Canadian Journal of Civil Engineering*, Vol. 23, pp. 577-586, 1996.

[34] ESRI, *Programming with Avenue*, Environmental Systems Research Institute, Redlands, CA, 1999.

[35] Evans, L., Herman, R., and Lam, T., Multivariate Analysis of Traffic Factors Related to Fuel Consumption in Urban Driving, *Transportation Science*, Vol. 10, No. 2, pp. 205-215, 1976.

[36] Evans, T.A., Djokic, D., and Maidment, D.R., Development and Application of Expert Geographic Information System, *Journal of Computing in Civil Engineering*, Vol. 7, No. 3, pp. 339-353, 1993.

[37] FHWA, *TRANSYT-7F Student Workbook*, U. S. Department of Transportation, 1984.

[38] FHWA, *Manual on Uniform Traffic Control Devices (MUTCD) 2000*, June 12, 2001, http://mutcd.fhwa.dot.gov/.

[39] Fwa, T. F., Highway Vertical Alignment Analysis by Dynamic Programming, *Transportation Research Record* 1239, TRB, National Research Council, Washington, D.C., pp. 1-9, 1989.

[40] Garber, N. J. and Hoel, L. A., *Traffic and Highway Engineering*, PWS Publishing Company, Massachusetts, 1996.

[41] Gen, M. and Cheng, R. , *Genetic Algorithms and Engineering Design*, John Wiley & Sons, Inc., New York, 1997

[42] Gilbrook, M.J., GIS Paves the Way, *Civil Engineering*, Vol. 69, No. 11, pp. 34-39, 1999.

[43] Glendening, P. N. , *Consolidated Transportation Program - 1997 State Report on Transportation*, Maryland Department of Transportation, 1997

[44] Goh, C. J. and Teo K. L., Control Parameterization: A Unified Approach to Optimal Control Problems with General Constraints, *Automatica*, Vol. 24, No.1, pp. 3-18, 1988.

[45] Goh, C. J., Chew, E. P., and Fwa, T. F., Discrete and Continuous Models for Computation of Optimal Vertical Highway Alignment, *Transportation Research Part B*, Vol. 22B, No 6, pp. 399-409, 1988.

[46] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., Massachusetts, 1989.

[47] Haling, D. and Cohen, H., Residential Noise Damage Costs Caused by Motor Vehicles, *Transportation Research Record* 1559, TRB, National Research Council, Washington, D.C., pp. 84-93, 1996.

[48] Halvorsen, R., and Ruby, M. G., *Benefit-Cost Analysis of Air-Pollution Control*, Lexington Books, 1981.

[49] Hammad, A., Itoh, Y., and Nishido, T., Bridge Planning using GIS and Expert System Approach, *Journal of Computing in Civil Engineering*, Vol. 7, No. 3, pp. 278-295, 1993.

[50] Hanson, S., *The Geography of Urban Transportation*, The Guilford Press, New York, 1995.

[51] Harwood, D. W., Wood, R. M., Newman, T. R., Potts, I. B., and Antonucci, N., Framework for an Expert System for Design Review of Intersections on Rural Two-Lane Highways, *2nd International Symposium on Highway Geometric Design*, Road and Transportation Research Association, Cologne, Germany, 2000.

[52] Hassan, Y., Easa, S. M., and Abd El Halim, A. O., Design Considerations for Combined Highway Alignments, *ASCE Journal of Transportation Engineering*, Vol. 123, No. 1, pp. 60-68, 1997.

[53] Hayman, R, W., Optimization of Vertical Alignment for Highways through Mathematical Programming, *Highway Research Record*, Highway Research Board, No. 306, pp. 1-9, 1970.

[54] Herman, R. and Ardekani, S., The Influence of Stops on Vehicle Fuel Consumption in Urban Traffic, *Transportation Science,* Vol. 19, No. 1, pp. 1-12, 1985.

[55] Hickerson, T. F., *Route Location and Design*, 5th Edition. McGraw-Hill Book Company, 1964.

[56] Hogan, J. D., Experience with OPTLOC – Optimum Location of Highways by Computer, *In PTRC Seminar Proceedings on Cost Models and Optimisation in Highways (Session L10),* London, 1973.

[57] Hokanson, B., Minkioff, M., Nichols, S., and Cowart, S., *Measures of Noise Damage Costs Attributable to Motor Vehicle Travel,* Institute of Urban and Regional Research, University of Iowa, Iowa City, 1981.

[58] Holzmann, F. D. and Marek, M. A., Interchange Study and Selection Process, *Transportation Research Record* 1385, TRB, National Research Council, Washington, D.C., pp. 90-99, 1993.

[59] Howard, B. E., Bramnick, Z., and Shaw, J. F. B., Optimum Curvature Principle in Highway Routing, *Journal of the Highway Division*, ASCE, Vol. 94, No. HW1, Proc. Paper 5987, pp. 61-82, 1968.

[60] ILAC, *Uniform Appraisal Standards for Federal Land Acquisitions*, Intergency Land Acquisition Conference, Washington, D.C., 1992.

[61] Institute of Transportation Engineers, *Manual of Traffic Signal Design, Second Edition*, Englewood Cliffs, NJ, USA, Prentice Hall, 1991.

[62] IRWA, *Principles of Right of Way*, International Right of Way Association, Gardena, California, 1990.

[63] Jha, M.K., *A Geographic Information Systems-Based Model for Highway Design Optimization*, Ph. D. Dissertation, University of Maryland, College Park, 2000.

[64] Jha, M.K., Using a GIS for Automated Decision-Making in Highway Cost Analysis, *Transportation Research Record* 1768, TRB, National Research Council, Washington, D.C., pp. 260-267, 2001.

[65] Jha, M.K., Optimizing Highway Networks: A Genetic Algorithm and Swarm Intelligence Based Approach, In *Computing in Civil Engineering*, pp. 76-89, A. Songer and J. Miles (eds.), American Society of Civil Engineers (ASCE), Reston, VA, 2002.

[66] Jha, M.K., Criteria-Based Decision Support System for Selecting Highway Alignments, *ASCE Journal of Transportation Engineering.* Vol. 129, No. 1, pp. 33-41, 2003.

[67] Jha, M.K. and Schonfeld, P., Geographic Information System-Based Analysis of Right-of-Way Cost for Highway Optimization, *Transportation Research Record* 1719, TRB, National Research Council, Washington, D.C., pp. 241-249, 2000a.

[68] Jha, M.K. and Schonfeld, P., Integrating Genetic Algorithms and GIS to Optimize Highway Alignments, *Transportation Research Record* 1719, TRB, National Research Council, Washington, D.C., pp. 233-240, 2000b.

[69] Jha, M.K. and Schonfeld, P., Tradeoffs between Initial and Maintenance Costs of Highways in Cross-Slopes, *Journal of Infrastructure Systems.* 9(1), pp. 16-25, 2003.

[70] Jha, M.K. and Schonfeld, P., A Highway Alignment Optimization Model using Geographic Information Systems, *Transportation Research, Part A*, 38, pp. 455-481, 2004.

[71] Jha, M.K., McCall, C., and Schonfeld, P., Using GIS, Genetic Algorithms, and Computer Visualization in Highway Development, *Computer-Aided Civil and Infrastructure Engineering*, Vol. 16, No. 6, pp. 399-414, 2001.

[72] Jong, J. C. and Schonfeld, P., An Evolutionary Program for Simultaneously Optimizing 3-Dimensional Highway Alignments, *Transportation Research Part B*, Vol.37, No. 2, pp. 1-22, 2003.

[73] Jong, J. C. and Schonfeld, P., Cost Functions for Optimizing Highway Alignments, *Transportation Research Record* 1659, TRB, National Research Council, Washington, D.C., pp. 58-67, 1999.

[74] Jong, J. C. and Schonfeld, P., Optimizing Vertical Alignments with Genetic Algorithms, *Transportation Research Board Annual Meeting*, 1999.

[75] Jong, J. C., *Optimizing Highway Alignments with Genetic Algorithms*. University of Maryland, College Park, Ph. D. Dissertation, 1998.

[76] Jong, J. C., Schonfeld, P., and Jha, M. K., Preliminary Highway Design with Genetic Algorithms and Geographic Information System, *Computer-Aided Civil and Infrastructure Engineering*, Vol. 15, No. 4, pp. 261-271, 2000.

[77] Jourquin, B. and Beuthe, M., Transportation Policy Analysis with a Geographic Information System: The Virtual Network of Freight Transportation in Europe, *Transportation Research Part C*, Vol. 4, No. 6, pp. 359-371, 1996.

[78] Khan, S., Shanmugam, R., and Hoeschen, B., Injury, Fatal, and Property Damage Accident Models for Highway Corridors, *Transportation Research Record* 1665, TRB, National Research Council, Washington, D.C., pp. 84-92, 1999.

[79] Kim, E., Jha, M.K., and Son, B., Improving the Computational Efficiency of Highway Alignment Optimization Models through a Stepwise Genetic Algorithms Approach, *Transportation Research, Part B*, Vol. 39, No. 4, pp. 339-60, 2005.

[80] Kim, E., Jha, M.K., and Schonfeld, P., Intersection Construction Cost Functions for Alignment Optimization, *ASCE Journal of Transportation Engineering*, Vol. 130, No. 2, pp. 194-203, 2004.

[81] Kim, E., Jha, M.K., Lovell, D.J., and Schonfeld, P., Intersection Cost Modeling for Highway Alignment Optimization, *Computer-Aided Civil and Infrastructure Engineering*, Vol. 19, No. 2, pp.136-146, 2004.

[82] Kim, E., Jha, M.K., Kim, K., and Son, B., Application of Planar Interpolation for Estimating Alignment Earthwork Costs with GIS, *Journal of the Korean Society of Civil Engineers*, 22(3-D), pp. 463-470, 2002.

[83] Kim, E., *Modeling Intersections and Other Structures in Highway Alignment Optimization*, Ph. D. Dissertation, University of Maryland, College Park, 2001.

[84] King, E. H. and Kuesel, T. R. (1996), An Introduction to Tunnel Engineering, Chapter 1 of *Tunnel Engineering Handbook*, 2nd Edition by Bickel, J. O., Kuesel, T. R. and King, E. H., An International Thompson Publishing Company.

[85] Krammes, R. A., Rao, K. S., and Oh, H., Highway Geometric Design Consistency Evaluation Software, *Transportation Research Record* 1500, TRB, National Research Council, Washington, D.C., pp. 19-24, 1995.

[86] Lamm, R. and Choueiri, E. M., Recommendations for Evaluating Horizontal Design Consistency Based on Investigations in the State of New York, *Transportation Research Record* 1122, TRB, National Research Council, Washington, D.C., pp. 68-78, 1987.

[87] Lamm, R., Guenther, A. K., and Grunwald, B., Environmental Impact on Highway Geometric Design in Western Europe Based on a Geographical Information System, *Transportation Research Record* 1445, TRB, National Research Council, Washington, D.C., pp. 54-63, 1994.

[88] Lau, M. Y-K. and May, Jr, A., Injury Accident Prediction Models for Signalized Intersections, *Transportation Research Record* 1172, TRB, National Research Council, Washington, D.C., pp. 58-67, 1988.

[89] Laurini, R. and Thompson, D., *Fundamentals of Spatial Information Systems*, The APIC Series, Academic Press, San Diego, 1992.

[90] Law, A. M. and Kelton, W. D., *Simulation Modeling & Analysis*, 2nd ed., McGraw-Hill, New York, 1991.

[91] Lay, D. C., *Linear Algebra and Its Applications*, 2nd Edition, Addison-Wesley Longman, Inc., 1997.

[92] Lee, H. and Clover, P., GIS-based Highway Design Review System to Improve Constructability of Design, *Journal of Advanced Transportation*, Vol. 29, No. 3, pp. 375-388, 1995.

[93] Leisch, J. P., Grade-Separated Intersections, *Transportation Research Record* 1385, TRB, National Research Council, Washington, D.C., pp. 48-50, 1993.

[94] Lettre, M., Garber, J., Dadd, B., and Baker, T., MDProperty View-Desktop Electronic Property Map and Parcel Data Analysis, *Document Management*, pp. 18-19, 1997.

[95] Liao, T-Y. and Machemehl, R. B., Development of an Aggregate Fuel Consumption Model for Signalized Intersections, *Transportation Research Record* 1641, TRB, National Research Council, Washington, D.C., pp. 9-18, 1998.

[96] Lovell, D. J., Automated calculations of sight distance from horizontal geometry, *ASEC Journal of Transportation Engineering*, Vol. 125, No.4, pp. 297-304, 1999.

[97] Lovell, D. J., Jong, J. C., and Chang, P. C., Improvements to a Sight Distance Algorithm, *ASCE Journal of Transportation Engineering*, Vol. 127, No. 4, pp. 283-288, 2001.

[98] McDonald Jr., D. R., Traffic Signal Warrants: Two Agencies' Preferences, *ITE Journal*, Vol. 71, No. 1, January, ITE, Washington, D.C., pp. 36-44, 2001.

[99] McShane, W. R. and Roess, R. P., *Traffic Engineering*, Prince-Hall Inc., Englewood Cliffs, New Jersey, 1990.

[100] MDOT, *Consolidated Transportation Program*, Maryland Department of Transportation, Baltimore, Maryland,1999.

[101] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd Edition, Springer-Verlag, New York, 1996.

[102] Miller, H.J. and Storm, J.D., Geographic Information System Design for Network Equilibrium based Travel Demand Models, *Transportation Research Part C*, Vol. 4, No. 6, pp. 373-389, 1996.

[103] Moavenzadeh, F., Becker, M., and Parody, T., *Highway Cost Model: Operating Instructions and Program Documentations*, Report on Contract DOT-OS-00096, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1973.

[104] Monchak, T. M. and Eaton, C. K., Arizona Department of Transportation and Environmental Protection Agency Cooperative Superfund Site Cleanup Effort for Red Mountain Freeway, *Transportation Research Record* 1475, TRB, National Research Council, Washington, D.C., pp. 85-91, 1996.

[105] Mortenson, M. E., *Geometric Modeling*, 2nd edition, John Wiley & Sons, Inc., New York, 1997.

[106] Mowczan, P. A., Tunnel Lighting, Chapter 21 of *Tunnel Engineering Handbook*, 2nd Edition by Bickel, J. O., Kuesel, T. R. and King, E. H., An International Thompson Publishing Company, 1996.

[107] Murchland, J. D., Methods of Vertical Profile Optimisation for an Improvement to an Existing Road, in *PTRC Seminar Proceedings on Cost Models and Optimisation in Highways (Session L12)*, London, 1973.

[108] NASA's Cost Models, August 21, 2001, www.jsc.nasa.gov/bu2/inflateGDP.html

[109] Nash, S. G. and Sofer, A., *Linear and Nonlinear Programming*, McGraw-Hill Companies, Inc., New York, 1996.

[110] Neter, J., Wasserman, W., and Whitemore, G. A., *Applied Statistics*, 2nd ed., Allyn and Bacon, Inc., Boston, 1982.

[111] Nicholson, A. J., Elms, D. G., and Williman, A., A Variational Approach to Optimal Route Location, *Highway Engineers*, Vol. 23, pp. 22-25, 1976.

[112] Nyerges, T. L., Geographical Information System Support For Urban/Regional Transportation Analysis, Chapter 10 of *The Geography of Urban Transportation*, edited by Hanson, S., The Guilford Press, New York, 1995.

[113] O'Connor, C., *Design of Bridge Superstructures*, John Wiley & Sons, Inc., 1971.

[114] OECD, *Optimisation of Road Alignment by the Use of Computers*, Organisation of Economic Co-operation and Development, Paris, 1973.

[115] Olivera, F. and Maidment, D., Geographic Information System Use for Hydrologic Data Development for Design of Highway Drainage Facilities, *Transportation Research Record* 1625, TRB, National Research Council, Washington, D.C., pp. 131-138, 1998.

[116] Park, B., *Development of Genetic Algorithm-Based Signal Optimization Program for Oversaturated Intersections*, Ph. D. Dissertation, Department of Civil Engineering, Texas A&M University, 1998.

[117] Parker, N. A., Rural Highway Route Corridor Selection, *Transportation Planning and Technology*, Vol. 3, pp. 247-256, 1977.

[118] Pearman, A. D., Investigation of the Objective Function in Problems of Optimising Highway Vertical Alignment, *In PTRC Seminar Proceedings on Cost Models and Optimisation in Highways (Session L18)*, London, 1973.

[119] Pline, J. L., *Traffic Engineering Handbook*, 4th edition, Institute of Transportation Engineers (ITE), Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1992.

[120] Polus, A. and Eck, R. W., Operational and Economic Considerations in an Evaluation of Geometric Design, *Transportation Research Record* 1122, TRB, National Research Council, Washington, D.C., pp. 12-17, 1987.

[121] Polus, A. Livneh, M., and Craus, J., Effect of Traffic and Geometric Measures on Highway Average Running Speed, *Transportation Research Record* 960, TRB, National Research Council, Washington, D.C., pp. 34-39, 1984.

[122] Ponnuswamy, S. and Victor, D. J., *Transportation Tunnels*, A. A. Balkema Publishers, Vermont, 1996.

[123] Puy Huarte, J., OPYGAR: Optimisation and Automatic Design of Highway Profiles, *In PTRC Seminar Proceedings on Cost Models and Optimisation in Highways (Session L13)*, London, 1973.

[124] ReVelle, C. S., Whitlatch, E. E., and Wright, J. R., *Civil and Environmental Systems Engineering*, Prentice Hall, New Jersey, 1997.

[125] Robertson, D. I., *TRANSYT: A Traffic Network Study Tool*, Road Research Laboratory Report 253, Crowthorne, Berkshire, Great Britain, 1969.

[126] Robinson, R., Automatic Design of the Road Vertical Alignment, in *PTRC Seminar Proceedings on Cost Models and Optimisation in Highways (Session L19)*, London, 1973.

[127] Sarasua, W. A., A GIS-based Traffic Signal Coordination and Information Management System, *Microcomputers in Civil Engineering*, Vol. 9, pp 235-250, 1994.

[128] Sayed, T. and Rodriguez, F., Accident Prediction Models for Urban Unsignalized Intersections in British Columbia, *Transportation Research Record* 1665, TRB, National Research Council, Washington, D.C., pp. 93-99, 1999.

[129] Schoon, J. G., *Geometric Design Projects for Highways: An Introduction*, American Society of Civil Engineers, New York, 1993.

[130] Shaw, J. F. B. and Howard, B. E., Expressway Route Optimization by OCP, *Transportation Engineering Journal of ASCE*, Proceedings of the American Society of Civil Engineers, ASCE, Vol. 108, No. TE3, pp. 227-243, 1982.

[131] Shaw, J. F. B., and Howard, B. E., Comparison of Two Integration Methods in Transportation Routing, *Transportation Research Record* 806, TRB, National Research Council, Washington, D.C., pp. 8-13, 1981.

[132] Simkowitz, H. J., GIS: Technology for Transportation, *Civil Engineering*, pp. 72-75, 1989.

[133] Sthapit, N. and Mori, H., Model to Estimate Highway Earthwork Cost in Nepal, *Journal of Transportation Engineering*, Vol. 120, No.2, pp. 498-504, 1994.

[134] Swokowski, E. W., *Calculus with Analytic Geometry*, 4th printing, Prindle, Weber & Schmidt, Boston, Massachusetts, 1999.

[135] Taber, J. T., *Multi-Objective Optimization of Intersection and Roadway Access Design*, Mountain-Plains Consortium (MPC) Report No. 98-90, Utah DOT, 1998.

[136] Teets, M. K., *Highway Statistics 1996*, U. S. DOT, Federal Highway Administration, Office of Highway Information Management, Washington D.C, 1997.

[137] Teo K. L. and Goh C. J., A Simple Computational Procedure for Optimization Problems with Functional Inequality Constraints, *IEEE Transactions on Automatic Control*, Vol. 32, No.10, pp. 940-941, 1987.

[138] Thomson, N. R. and Sykes, J. F., Route Selection through a Dynamic Ice Field Using the Maximum Principle, *Transportation Research Part B*, Vol. 22B, No. 5, pp. 339-356, 1988.

[139] TRB, *Highway Capacity Manual: Special Report* 209, National Research Council, Washington, D. C., 2000.

[140] Trietsch D., Comprehensive Design of Highway Networks, *Transportation Science*, Vol. 21, No. 1, pp. 26-35, 1987b.

[141] Trietsch, D., A Family of Methods for Preliminary Highway Alignment, *Transportation Science,* Vol. 21, No. 1, pp. 17-25, 1987a.

[142] Turner, A, K., A Decade of Experience in Computer Aided Route Selection, *Photogrammetric Engineering and Remote Sensing,* Vol. 44, pp. 1561-1576, 1978.

[143] Turner, A. K. and Miles, R. D., A Computer-Assisted Method of Regional Route Location, *Highway Research Record* 348, pp. 1-15, 1971.

[144] Turner, S., Weaver, C., and Reale, M., *Cost and Emissions Benefits of Selected Air Pollution Control Measures for Santiago, Chile: Final Report*, Engine, Fuel, and Emissions Engineering, Inc., Sacramento, California, 1993.

[145] Underwood, R. T., *The Geometric Design of Roads*, The Macmillan Company of Australia pty ltd, Australia, 1991.

[146] United States Steel Corporation, *Short Span Steel Bridges (Load Factor Design),* Pittsburgh, 1978.

[147] Vandermark, B. and Corbley, K., Accurate Basemap Development, *Public Works,* pp. 53-54, 1996.

[148] Versenyi, J.H., Holdstock, D.A., and Fischer, T., Considering the Alternatives: GIS Identifies Nine Possibilities for Highway Connection, *Geo Info Systems,* Vol. 4, No. 9, pp. 43-45, 1994.

[149] Vogt, A. and Bared, J., Accident Models for Two-Lane Rural Segments and Intersections, *Transportation Research Record* 1635, TRB, National Research Council, Washington, D.C., pp. 18-29, 1998a.

[150] Vogt, A. and Bared, J. G., *Accident Models for Two-Lane Rural Roads: Segments and Intersections,* Report FHWA-RD-98-133, FHWA, U. S. Department of Transportation, 1998b.

[151] Walker, R. J., Coordination of Basic Intersection Design Elements: An Overview, *Transportation Research Record* 1385, TRB, National Research Council, Washington, D.C., pp. 51-59, 1993.

[152] Wallace, C. E., Courage, K. G., and Hadi, M. A., *TRANSYT-7F User's Guide*, Transportation Research Center, University of Florida, Gainesville, Florida, 1991.

[153] Wallace, C. E., Courage, K. G., Hadi, M. A., and Gan, A. C., *TRANSYT-7F User's Guide*, Transportation Research Center, University of Florida, Gainesville, Florida, 1998.

[154] Walton, D. J. and Meek, D. S., Computer-Aided Design for Horizontal Alignment, *ASCE Journal of Transportation Engineering*, Vol. 115, No. 4, pp. 411-423, 1989.

[155] Wan, F. Y. M., *Introduction to the Calculus of Variations and its Applications*, Chapman & Hall, New York, 1995.

[156] Warwick, J. J. and Haness, S. J., Accurate Polygon Centroid Computation using ARC/INFO GIS, *Journal of Computing in Civil Engineering*, Vol. 7, No. 3, pp. 388-392, 1993.

[157] Watanatada, T., Dhareshwar, A. M., and Rezende-Lima, P. R. S., *Vehicle Speeds and Operating Costs — Models for Road Planning and Management*, The International Bank for Reconstruction and Development / The World Bank, Washington, D.C, 1987.

[158] Webster, F. V., Traffic Signal Settings, *Road Research Technical Paper*, No. 39, Her Majesty's Stationary Office, London, 1958.

[159] Winfrey, R., *Economic Analysis for Highways*, International Textbook, Scranton, PA, 1968.

[160] Winston, W. L., *Operations Research, Applications and Algorithms*, 3rd edition, International Thomson Publishing, Boston, 1994.

[161] Wright, P. H., *Highway Engineering*, John Wiley & Sons, Inc., New York, 1996.

[162] Xanthakos, P. P., *Theory and Design of Bridges*, John Wiley & Sons, 1994.

[163] Zaniewski, J. P., Fuel Consumption Related to Roadway Characteristics, *Transportation Research Record* 901, TRB, National Research Council, Washington, D.C., pp. 18-29, 1983.

[164] Zegeer, C. V., Hummer, J., Herf, L., Reinfurt, D., and Hunter, W., *Safety Effects of Cross-Section Design for Two-Lane Roads*, Report FHWA-RD-86-125, FHWA, U. S. Department of Transportation, 1986.

[165] Zegeer, C. V., Stewart, R., Council, F. M., Reinfurt, D. W., and Hamilton, E., Safety Effects of Geometric Improvements on Horizontal Curves, *Transportation Research Record* 1356, TRB, National Research Council, Washington, D.C., pp. 11-19, 1992.

[166] Zegras, C. and Litman, T. Cost Estimation of Transport Air Pollution in Santiago, Chile, *Transportation Research Record* 1587, TRB, National Research Council, Washington, D.C., pp. 106-112, 1997.

This page intentionally left blank

Index

WIT*PRESS*

How to Make Two-Lane Rural Roads Safer

Scientific Background and Guide for Practical Application

R. LAMM, University of Karlsruhe, Germany, A. BECK, University of Karlsruhe, Germany, T. RUSCHER, University of Karlsruhe, Germany, T. MAILAENDER, Mailaender Ingenieur Consult GmbH, Germany, S. CAFISO, University of Catania, Italy, G. LA CAVA, University of Catania, Italy

In most countries two-lane rural roads make up about 90 percent of rural networks and account for about 60 percent or more of highway fatalities worldwide - 500,000 people per year. Based on new research and the demands of many design professionals this book provides an understandable scientific framework for the application of quantitative safety evaluation processes on two-lane rural roads.

The methodology described will support the achievement of quantified measures of 1) design consistency, 2) operating speed consistency, and 3) driving dynamic consistency. All three criteria are evaluated in three ranges described as "good", "fair" and "poor". It has been proved that the results of these criteria coincide with the actual accident situation prevailing on two-lane rural roads. By using the "good" ranges sound alignments in plan and profile, which match the expected driving behaviour of motorists, can be achieved. The safety criteria are then combined into an overall safety module for a simplified general overview of the safety evaluation process. The authors also encourage the coordination of safety concerns with important economic, environmental and aesthetic considerations. A CD-ROM with practical applications accompanies the text.

This book will be an invaluable aid to educators, students, consultants, highway engineers and administrators, as well as scientists in the fields of highway design and traffic safety engineering.

ISBN: 1-84564-156-6 2006
apx 104pp+CD-ROM
apx £59.00/US$94.00/€88.50

Hybrid Vehicle Propulsion

C.M. JEFFERSON, University of the West of England, UK and R.H. BARNARD, University of Hertfordshire, UK

In this book, the authors review recent progress in the development of a range of hybrid vehicles and describe the results of field trials and operational experience. Numerous tables, graphs and photographs are included together with clear references.

The volume will be of great interest to engineering and technical staff working in the road and rail vehicle industries, and final year undergraduates and postgraduates studying mechanical and automotive engineering.

Series: Advances in Transport, Vol 10

ISBN: 1-85312-887-2 2002 176pp
£69.00/US$107.00/€103.50

Innovations in Freight Transport

Editors: E. TANIGUCHI, Kyoto University, Japan and R.G. THOMPSON, University of Melbourne, Australia

Highlighting new ideas and best practice, this book examines innovations in modern freight transport systems.

Partial Contents: Intelligent Transport Systems; Vehicle Routing and Scheduling; Logistics Terminals; Intermodal Freight Transport; Underground Freight Transport Systems; E-Commerce and the Consequences for Freight Transport; Future Perspectives.

Series: Advances in Transport, Vol 11

ISBN: 1-85312-894-5 2002 216pp
£76.00/US$118.00/€114.00

All prices correct at time of going to press but subject to change.
WIT Press books are available through your bookseller or direct from the publisher.

This page intentionally left blank

This page intentionally left blank

This page intentionally left blank