Mechanism and Machine Theory 45 (2010) 1096–1107

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmt

A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage

Wen-Yi Lin *

Department of Mechanical Engineering, De Lin Institute of Technology, 1 Lane 380, Qingyan Road, Tucheng, Taipei County, Taiwan, ROC

ARTICLE INFO

Article history:
Received 27 June 2009
Received in revised form 27 October 2009
Accepted 18 March 2010

Keywords:
Four-bar mechanism
Mechanism synthesis
Genetic algorithms
Evolutionary algorithm
Coupler curve
Path generation
Mechanism optimization

ABSTRACT

A new real-coded evolutionary algorithm is proposed for application to path synthesis of a four-bar linkage. This new evolutionary algorithm is obtained by combining differential evolution (DE) with the real-valued genetic algorithm (RGA). We term this the "GA–DE hybrid algorithm." The only difference between the proposed algorithm and RGA is in the content of the crossover. The crossover operation in the RGA is replaced by differential vector perturbation, with the best individual or some excellent individuals as the base vectors. Thus, both the main perturbation of differential vectors and the minor perturbation of mutation are used as genetic operators in the GA–DE hybrid algorithm. The efficiency and accuracy of the proposed method are tested using four cases. Findings show that much more accurate solutions for three cases are obtained with this method than those obtained using other evolutionary methods as discussed in the literature. A moveable stick diagram of the synthesized mechanisms can be obtained using the 2D sketch feature of SolidWorks®. This can be used to check whether the synthesized mechanisms encounter circuit defects or are incapable of motion.

1. Introduction

The path synthesis of a four-bar linkage has been actively studied during the past 50 years. There have been a large number of studies on this topic using a variety of methods. These include analytical [1,2], continuation [3], nonlinear goal programming [4,5], exact gradient [6], coupler-angle function curve [7] and curve curvature methods [8]. The problem of path synthesis of a four-bar linkage is to generate a mechanism whose coupler point can trace the desired trajectory or target points. There is no analytical solution to the general problem of four-bar linkage synthesis for more than five target points. This type of problem may be solved by a variety of numerical methods. For example, it may be considered a mechanism optimization to minimize an objective function. The most common objective function is the so-called position error, defined as the sum of the square of the Euclidean distance between the target points and the obtained coupler points. Following the development of evolutionary algorithms from natural biological evolution and swarm intelligence, the merits of these algorithms are what make them suitable for application to path synthesis of the four-bar linkage. The genetic algorithm (GA) was first introduced by Holland [9], then verified by his student, DeJong [10], as applicable to numerical optimization problems. Differential evolution (DE), first proposed by Storn and Price [11], is another simple yet powerful evolutionary algorithm for real parameter optimization. Particle swarm optimization (PSO), proposed by Kennedy and Eberhart [12], is one of the swarm intelligence optimization methods that operate by imitating the swarming behaviors of natural creatures. These evolutionary methods are simple and easy to implement for solving complicated real-world optimization problems. For these evolutionary methods, there is no need for further demands pertaining to the gradient or other mathematical characteristics of the objective function. Furthermore, if the objective function of an optimization task is not expressed as an explicit function of the design variables, or alternatively, it is too complicated to manipulate, e.g., the

* Tel.: +886 2 22733567; fax: +886 2 22734420.
E-mail address: wylin@dlit.edu.tw.

0094-114X/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechmachtheory.2010.03.011

optimization problem of dimensional synthesis of the five-point double-toggle mold clamping mechanism [13,14], it is more suitable to use these evolutionary algorithms for optimization than the traditional deterministic optimization methods.

Although "genetic algorithm" is used in the title of Ref. [15], Cabrera et al. actually applied a new evolutionary method based on DE to the path synthesis of a four-bar linkage. However, their method is somewhat different than either GA or DE. The main difference between the new evolutionary method and DE is the mutation operation in GA retained in the new evolutionary method. The objective function has two parts. The first part is concerned with the position error. In the second part the constraints are inserted by the penalty function. Three cases were studied. The solutions of two cases were compared with those obtained by Kunjur and Krishnamurty [16] based on the deterministic and genetic methods. Findings show that quite small errors in fast convergence are obtained and therefore the new evolutionary algorithm is relatively effective and efficient. Notice that the final error for case 2 obtained by Cabrera et al. is the same as the position error for their synthesized parameters. However, the values of the final errors for cases 1 and 3 are not the same. Alternatively, the errors for cases 1 and 3 obtained by Cabrera et al. might be incorrect or typing errors. To find the best algorithm among binary-coded GA (BGA), PSO and DE, Acharyya and Mandal [17] applied BGA with multipoint crossover, the PSO with the CFA (constriction factor approach) and the DE with DE/rand/1/exp strategy to the path synthesis of a four-bar linkage. They also introduced a new refinement technique for the generation of initial population. This technique is quite helpful for solving the constrained optimization problem. Three cases were studied for more than five target points. Findings show that the DE method produces a better solution than that obtained previously by Cabrera et al. for case 1, and DE outperforms BGA and PSO for all three cases. However, some of their results for cases 2 and 3 might be unreliable. The error for case 1 obtained by Cabrera et al. has been corrected by Acharyya and Mandal. Notice again that the error employed by Acharyya and Mandal is the square root of the position error. However, this is not clearly stated. The error for the BGA solution for case 2 and the errors for the BGA, PSO and DE solutions for case 3 might be incorrect or alternatively, their synthesized parameters might be typing errors. If the synthesized parameters shown in their paper are correct, the error for the BGA solution for case 2 should be 45.817, not 3.171 (according to their synthesized parameters). The errors for the BGA, PSO and DE solutions for case 3 should be 2.403, 37.083 and 50.423, respectively, not 2.281, 1.971 and 1.952 (according to their synthesized parameters).

The aim of this work is to propose a new real-valued evolutionary algorithm to apply to the path synthesis problems of the four-bar linkage. The new evolutionary algorithm is a combination of the real-valued genetic algorithm (RGA) and the differential evolution. Accordingly, this algorithm will be termed "GA–DE hybrid (evolutionary) algorithm." To verify the efficiency and accuracy of the proposed method, several of the afore-mentioned cases are studied. The only difference between the proposed evolutionary algorithm and the real-valued genetic algorithm is in the content of the crossover. The crossover operation in RGA is replaced by the differential vector perturbation of DE [18], with the best individual or some excellent individuals as the base vectors. Thus, both the main perturbation of the differential vectors and the minor perturbation of the mutation act as genetic operators in the proposed evolutionary algorithm.

It has been pointed out [19] that the synthesized solution may be unusable due to circuit or order defects, or be incapable of moving from one coupler point to another. Although a synthesized solution can be at the traced points, there is no guarantee regarding the behavior of the linkage between those traced points. A circuit defect occurs when the linkage cannot move into a region between those traced points, that is, positions do not exist in such this region for a linkage with a circuit defect. Therefore, a linkage with a circuit defect must be disassembled before the region and reassembled after the region to complete the motion. This is a fatal defect. In this work, a 2D sketch produced by SolidWorks® is used to check whether the synthesized mechanism encounters such defects. One can manipulate rotations of the driving link of the SolidWorks® 2D sketch for the synthesized linkage by using the mouse to drag the driving link to produce the motion. If the driving link cannot be dragged into a region, this indicates the presence of a circuit defect. If one compels the driving link to complete the motion in the 2D sketch, one can observe the phenomenon of jumping between the two limiting positions. Besides, if the defect leading to incapable of movement is not the result of the circuit defect, the driving link cannot be compelled to move in the 2D sketch. The 2D sketch is also helpful to examine the distance between the target point and the synthesized coupler point obtained from the synthesized geometric parameters and the position equations.

2. Problem formulation

2.1. Position equations

Fig. 1 depicts a skeleton drawing and all the geometric parameters of the four-bar linkage, where C is the coupler point. The DOF of the four-bar linkage is one. Angle θ_2 is the input angle in this work and angle θ_3 can be solved by the Freudenstein equation [20]. The position of coupler point C in the world coordinate system OXY can be expressed by

$$\begin{bmatrix} C_X \\ C_Y \end{bmatrix} = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 \\ \sin\theta_0 & \cos\theta_0 \end{bmatrix} \begin{bmatrix} C_{Xr} \\ C_{Yr} \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{1}$$

where

$$\begin{aligned} C_{Xr} &= r_2 \cos\theta_2 + r_{cx}\cos\theta_3 - r_{cy}\sin\theta_3 \\ C_{Yr} &= r_2 \sin\theta_2 + r_{cx}\sin\theta_3 + r_{cy}\cos\theta_3 \end{aligned} \tag{2}$$

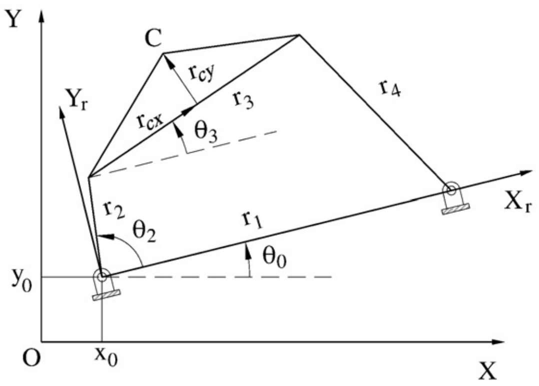

Fig. 1. Four-bar mechanism in the global coordinate system.

2.2. Design parameters

There are at least nine design parameters for path generation with prescribed timing, r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, x_0, y_0 and θ_0 to be optimized. In addition to the nine design parameters, there are input angles θ_2^i $(i=1-N)$ corresponding to target points that need to be optimized for path generation without prescribed timing. Design variable vector $\mathbf{X}$ is given as follows:

$$\mathbf{X} = \left[r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, x_0, y_0, \theta_0, \theta_2^1, \theta_2^2 \ldots \theta_2^N\right] \tag{3}$$

where N is the number of target points to be optimized.

2.3. Design objective

The position error is defined as the sum of the square of the Euclidean distance between each C_d^i and the corresponding C^i, where $\{C_d^i\}$ is the set of the positions of target points (indicated by the designer); and $\{C^i\}$ is the set of the positions of the coupler point of the synthesized mechanism for a set of input angles $\{\theta_2^i\}$. They can be written in the world coordinate system as $\mathbf{C}_d^i = [C_{Xd}^i, C_{Yd}^i]^T$ and $\mathbf{C}^i = [C_X^i, C_Y^i]^T$, respectively.

In this work, the four-bar linkage synthesized problem is to minimize the position error considered as the first part of the objective function, which may be expressed by

$$f_{\text{obj}} = \sum_{i=1}^{N} \left[\left(C_{Xd}^i - C_X^i(\mathbf{X})\right)^2 + \left(C_{Yd}^i - C_Y^i(\mathbf{X})\right)^2\right] \tag{4}$$

For the establishment of a termination criterion, the mean error in distance between the target and coupler points is defined as

$$e_{\text{m}} = \frac{1}{N}\sum_{i=1}^{N} \sqrt{\left(C_{Xd}^i - C_X^i(\mathbf{X})\right)^2 + \left(C_{Yd}^i - C_Y^i(\mathbf{X})\right)^2} \tag{5}$$

2.4. Constraints

In this study, only the following constraints are considered:

1. the Grashof condition is satisfied to allow for the entire turn of at least one link. It may be expressed by

$$2[max(r_1, r_2, r_3, r_4) + min(r_1, r_2, r_3, r_4)] < (r_1 + r_2 + r_3 + r_4). \tag{6}$$

2. The sequence of input angles θ_2 satisfies CW or CCW rotation of the crank.
3. The design parameters are within the specified ranges.
4. The rotation range of the crank.

For all cases discussed here, the crank has 360° of rotation (constraint 4). Apart from the input angles, constraint 3 may be achieved by assigning the value of a design variable within the prescribed range during initialization and mutation operations. Furthermore, if the value obtained by crossover operation is not within the prescribed range, one may repeat this operation until the value falls within the prescribed range. The input angles are assigned values within the range $[0, 2\pi]$ only during initialization and mutation operations. If the obtained input angle exceed its limit, say $2\pi < \theta_2 < 4\pi$, the final value is obtained by subtraction of 2π

from the obtained value. Therefore, it is allowable for the value of the input angle to exceed its limit during the course of evolution, so long as constraint 2 is satisfied in the final evolution. Furthermore, by rearranging the input angles to satisfy CW or CCW rotation during initialization and mutation operations, constraint 2 may be ignored until the end of the evolution. Sound chromosomes may be obtained both by this scheme and the mechanism of evolution. The scheme proposed in this work differs from that proposed by Acharyya and Mandal, in that their scheme is carried out only in the process of initialization and their algorithm always checks constraints 2 in each generation. Lastly constraint 1 is inserted into the objective function as a penalty function as follows:

Minimize

$$f_{\text{obj}} = \sum_{i=1}^{N} \left[\left(C_{Xd}^{i} - C_{X}^{i}(\mathbf{X}) \right)^2 + \left(C_{Yd}^{i} - C_{Y}^{i}(\mathbf{X}) \right)^2 \right] + M_1 h_1(\mathbf{X}) \tag{7}$$

where $h_1(\mathbf{X}) = 0$ indicates that Grashof's condition is true and $h_1(\mathbf{X}) = 1$ indicates that Grashof's condition is false; M_1 is a constant with a very large value that penalizes the objective function when the constraint fails.

3. GA–DE hybrid evolutionary algorithm

The GA–DE hybrid evolutionary algorithm is obtained by combining the real-valued genetic algorithm with the differential evolution. Genetic algorithms are global search techniques based on the concepts of genetics, natural selection and survival of the fittest. The RGA search process is briefly outlined below. A population with randomly generated chromosomes is initialized. Each chromosome (individual) is a candidate solution. The quality of the individual is estimated by the fitness value related to the objective function. Offsprings are generated using genetic operators, crossover and mutation, from either one or two individuals (parents). The parents are randomly selected according to a probability proportional to their fitness. A population evolves from generation to generation by selection pressure and genetic operations.

There are several methods of crossover in RGA, such as arithmetic crossover [21] and blend crossover (BLX-α) [22]. However, these RGA methods do not necessarily work better than those of BGA for the problem of linkage synthesis. There exist a number of other crossover operations, for example, simulated binary crossover [23,24] and fuzzy recombination operations [25]. Beyer and Deb [26] found there to be some similarity in such crossover methods. As pointed out in [27], crossover using such methods is not as meaningful in RGA when contrasted with crossover in BGA. Crossover operators in RGA may be best described as blending operators and may be regarded as perturbations. Most blending operators in RGA are known as crossover operators. Furthermore, although debatable, Deb believes that the distinction between the crossover and mutation operations in RGA lies mainly in the number of individuals used in the perturbation process. We agree with Deb's opinion, so in this work, the crossover is replaced by a differential vector perturbation of DE evolution. Thus, both the main perturbation of differential vectors and the minor perturbation of mutation which act as genetic operators are used in the proposed evolutionary algorithm.

Here, genes x_i $(i = 1-n)$ represent the design parameters encoded in terms of real numbers that fall between their bounds. All genes are grouped in a vector $\mathbf{X}$ that represents a chromosome. That is

$$x_i \in [min(x_i), max(x_i)] \tag{8}$$

$$\mathbf{X} = [x_1, x_2, \ldots, x_n] = \left[r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, x_0, y_0, \theta_0, \theta_2^1, \theta_2^2 \ldots \theta_2^N \right] \tag{9}$$

where n is the number of the design parameters.

3.1. Initialization

An initial population with N_p chromosomes is randomly generated. The gene in each chromosome is given by

$$x_i = min(x_i) + \gamma(max(x_i) - min(x_i)) \tag{10}$$

where γ is a random real number between 0 and 1. Since the Grashof condition fails, it is possible that only a small to moderate amount of individuals may survive. Therefore, during initialization it is natural to reassign r_1, r_2, r_3 and r_4 until the Grashof condition is satisfied. As pointed out by Acharyya and Mandal, the order of randomly generated $\theta_2^i (i = 1-N)$ can be rearranged in CW or CCW; thereafter θ_2^i $(i = 1-N)$ are reassigned in CCW in this work.

3.2. Fitness

The quality of the chromosome is estimated by examining the fitness value. The fitness value is obtained by subtracting the value of the objective function as defined in Eq. (7) from a prescribed large value for the minimization problem. The prescribed large value may be chosen to avoid the fitness of a worse but feasible individual becoming negative and to maintain population diversity. The prescribed large value is 10,000 in this work. The negative fitness value is set to zero.

While the mean error is obtained, a termination criterion is then checked, as given by

$$e_\text{m} \le e_\text{tol} \tag{11}$$

where e_{tol} is a prescribed value for error tolerance. If the termination criterion is not satisfied, the population is updated using perturbation operations to produce the next generation. The procedure is repeated until the termination criterion is satisfied or until the number of generation is equal to the prescribed number of generations G_{max}.

3.3. Selection and pairing

Roulette-wheel selection is employed in this work to allow those individuals with higher fitness to have a higher chance of being selected. Thereafter two non-repeated individuals are paired randomly to be parents.

3.4. Differential vector perturbations

The crossover operation in RGA is replaced by differential vector perturbation of DE, with the best individual or some excellent individuals as the base vectors. Therefore, parents are used as differential vectors, not crossover. All paired parents are randomly divided into k groups corresponding to the top k individuals, denoted by $\mathbf{X}_{top1}, \mathbf{X}_{top2}\ldots\mathbf{X}_{topk}$. These form the base vectors. The value of k is a user-defined integer. In the i-th group, the offspring may be generated by

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{X}_{topi} + F_1(\mathbf{X}_{r1} - \mathbf{X}_{r2}) \\ \mathbf{X}_2 &= \mathbf{X}_{topi} - F_2(\mathbf{X}_{r1} - \mathbf{X}_{r2}) \end{aligned} \tag{12}$$

where $\mathbf{X}_{r1}$ and $\mathbf{X}_{r2}$ are the parents; F_1 and F_2 are random real numbers between 0 and 1 for each design variable.

A main perturbation rate P_{m1} is defined as the ratio of the expected number of offspring generated by differential vector perturbations to the total population size, while replacing crossover rate P_c in RGA.

3.5. Mutation

Mutation may be useful in maintaining population diversity. Here, mutation is performed to replace randomly selected chromosome by random real numbers (that are within the limits of variables). The minor perturbation rate P_{m2}, i.e., mutation rate, is defined as the ratio of the expected number of offspring introduced by mutation to the total population size.

3.6. Elitism

The technique of elitism [28] is employed in this work is defined as follows: the best-so-far individual is always retained, replacing the current worst one, carrying on into the succeeding generation.

3.7. Comparison with DE

There are some differences between the GA–DE hybrid algorithm and DE as follows.

1. For the same number of differential vectors, the range (or the degree of uniform) of differential vector distribution for the GA–DE hybrid algorithm may be better than DE. This is because the definition of disturbing vectors in Eq. (12) and non-repeated individuals $\mathbf{X}_{r1}$ and $\mathbf{X}_{r2}$ are paired randomly. In different strategies of DE, the disturbing vector for the DE/x/y/z with $y = 2$ (x: object to be disturbed, y: number of differential vectors, z: crossover way) is obtained from 4 random individuals. However, the DE/x/1/z is the most widely used strategy, being used in the works of Cabrera et al. and Acharyya and Mandal.
2. The number of the object to be disturbed in the GA–DE hybrid algorithm can be one, two, or more, while the number of the base vector for DE is only one. This might be helpful when searching a better solution for some problems or for other purposes.
3. Both the main perturbation of the differential vectors and the minor perturbation of the mutation are used as genetic operators in the GA–DE hybrid algorithm. However, the frame of this algorithm is the same as in GA. In DE, the main perturbation of the differential vectors (called mutation in DE) is implemented and the crossover operator is retained for population diversity.
4. The ways in which selection is carried out are parent selection and survivor selection (replacement) for the GA–DE hybrid algorithm and DE, respectively.

4. Results and discussion

The efficiency and accuracy of the proposed method are verified by studying four cases (for more than five target points) from the literature. Two sets of user-supplied parameters are used. One is for the purpose of comparison with other solutions discussed in the literature. It includes the following: size of population $N_p = 100$; only the best individual is used as the base vector for the differential vector perturbation, i.e., $k = 1$; number of generations $G_{max} = 1000$ for cases 1-3 and $G_{max} = 50$ for case 4; major perturbation rate $P_{m1} = 0.6$; minor perturbation rate $P_{m2} = 0.01$; error tolerance $e_{tol} = 10^{-4}$. The other is expected to help obtain more satisfactory solutions. It includes the following: size of population $N_p = 400$; top two individuals are used as the base vectors, i.e., $k = 2$; the other user-supplied parameters are the same.

All the synthesized solutions in this work have been validated so that there are no circuit and order defects and they are not incapable of motion. The number (N^o) of evaluations of the objective function and the population size are shown in the tables.

4.1. Case 1: 6 target points and 15 design variables

The first case is a path synthesized problem with six target points arranged in a vertical line and without prescribed timing.

$$\text{Design variables}: \mathbf{X} = \left[r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, x_0, y_0, \theta_0, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6\right]$$

$$\text{Target points}: C_d^i = [(20,20),(20,25),(20,30),(20,35),(20,40),(20,45)]$$

$$\text{Limits of the variables}: r_1, r_2, r_3, r_4 \in [5,60]; r_{cx}, r_{cy}, x_0, y_0 \in [-60,60]; \theta_0, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6 \in [0,2\pi].$$

The synthesized geometric parameters and the corresponding values of the objective function and the mean error are shown in Table 1, together with the synthesized solutions obtained by Cabrera et al. and Acharyya and Mandal. The original value of θ_2^1 obtained for $N_p = 100$ and $k = 1$ is 6.80719 and exceeds 2π. Therefore, its real value should be 0.524005. Although the constraint of the sequence of the input angles during the evolution is ignored in this case, there is no order defect in the best synthesized mechanisms, as can be seen in Table 1. Obviously, the accuracy of the solution for case 1 has been remarkably improved using the present method. The mean error for $N_p = 100$ and $k = 1$ obtained in this work show a decline by about 76% compared with the DE solution obtained by Acharyya and Mandal, and shows a decline by about 86% compared with the solution obtained by Cabrera et al. Clearly, the BGA solution is most unsatisfactory one. As expected, and as shown in Table 1, the solution for $N_p = 400$ and $k = 2$ is more accurate than that for $N_p = 100$ and $k = 1$. For $N_p = 400$ and $k = 2$, we find some solutions for the objective function value on the order of 10^{-6}. However, the order defects are encountered. Fig. 2 shows the six target points and the coupler curve obtained using the proposed method with $N_p = 100$ and $k = 1$ for case 1, together with that obtained by Acharyya and Mandal. Fig. 3 shows the six target points and the coupler curve obtained using the proposed method with $N_p = 400$ and $k = 2$ for case 1.

4.2. Case 2: 6 target points and 9 design variables

This case is a path synthesized problem with six target points arranged in a semi-circular arc and prescribed timing.

$$\text{Design variables}: \mathbf{X} = \left[r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, x_0, y_0, \theta_0\right]$$

$$\text{Target points}: C_d^i = [(0,0),(1.9098,5.8779),(6.9098,9.5106),(13.09,9.5106),(18.09,5.8779),(20,0)] \text{ and } \theta_2^i = \left[\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}, \pi\right]$$

$$\text{Limits of the variables}: r_1, r_2, r_3, r_4 \in [5,50];\ r_{cx}, r_{cy}, x_0, y_0 \in [-50,50]; \theta_0 \in [0,2\pi].$$

The synthesized geometric parameters and the corresponding values of the objective function and the mean error are shown in Table 2, together with the synthesized solutions obtained by Acharyya and Mandal. Obviously, the BGA solution is unsound. The mean

Table 1
Synthesized results for case 1.

	Present		[15]	BGA [17]	DE [17]
	$N_p = 100, k = 1$	$N_p = 400, k = 2$	$N_p = 100$	$N_p = 100$	$N_p = 100$
N^o	100,000	400,000	100,000	200,000	100,000
r_1	33.5959	13.2516	39.46629	28.77133	35.02074
r_2	5.02972	5.94078	8.562912	5.000000	6.404196
r_3	11.1847	58.3118	19.09486	35.36548	31.60722
r_4	28.0878	53.7207	47.83886	59.13681	50.59949
r_{cx}	−24.1755	16.3826	13.38556	0.000000	20.80324
r_{cy}	5.51479	−59.0715	12.21961	14.85037	41.54364
x_0	39.7799	−35.3621	29.72255	29.91329	60.00000
y_0	24.7195	36.7704	23.45454	32.60228	18.07791
θ_0	5.45884	0.196076	6.201627	5.287474	0.000000
θ_2^1	0.524005 *	1.66015	6.119371	6.283185	6.283185
θ_2^2	0.853145	2.04684	0.19304	0.318205	0.264935
θ_2^3	1.16505	2.42811	0.44083	0.638520	0.500377
θ_2^4	1.49253	2.80901	0.684674	0.979950	0.735321
θ_2^5	1.87456	3.19009	0.958351	1.412732	0.996529
θ_2^6	2.44206	3.57379	1.355331	2.076254	1.333549
f_{obj}	7.36984×10^{-4}	1.72075×10^{-5}	3.6279×10^{-2}	1.21216	1.50653×10^{-2}
e_m	9.53563×10^{-3}	1.43295×10^{-3}	6.83347×10^{-2}	4.1825×10^{-1}	3.96175×10^{-2}

* The original value of θ_2^1 in the final evolution is 6.80719.

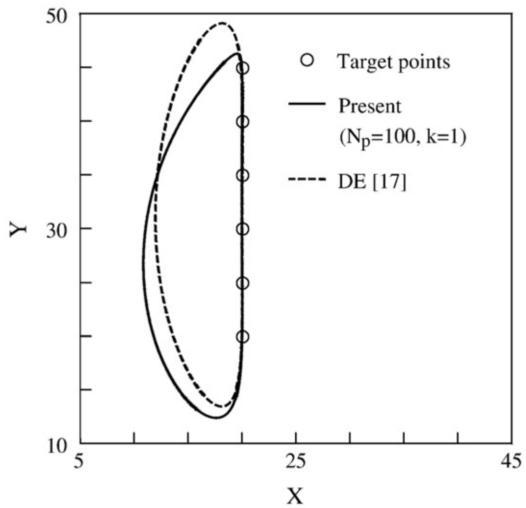

Fig. 2. Six target points and the coupler curve obtained using the proposed method with $N_p = 100$ and $k = 1$ for case 1, together with that obtained by Acharyya and Mandal.

error for $N_p = 100$ and $k = 1$ obtained in this work shows a decline by about 31% as compared with the DE solution. The improvement of the solution when $N_p = 400$ and $k = 2$ is almost negligible compared to that when $N_p = 100$ and $k = 1$, as shown in Table 2. Therefore, this case should be studied further, using other algorithms or design considerations. Fig. 4 shows the six target points and the coupler curve obtained using the proposed method with $N_p = 100$ and $k = 1$ for case 2, together with that obtained by Acharyya and Mandal. The typical evolution of the value of the objective function of the best-so-far individual for case 2 corresponding to the number of generations, obtained with the proposed evolutionary algorithm, is shown in Fig. 5. The convergence rate for the best-so-far individual is fast. The value of the objective function of the best-so-far individual for this case almost converges to a minimum value within only 200 generations; thereafter the improvement of the objective is negligible, which is similar to the result obtained by Acharyya and Mandal.

4.3. Case 3: 10 target points and 19 design variables

The case is a path synthesized problem with 10 target points arranged in an ellipse and without prescribed timing.

$$\text{Design variables}: \mathbf{X} = \left[r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, x_0, y_0, \theta_0, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6, \theta_2^7, \theta_2^8, \theta_2^9, \theta_2^{10}\right]$$

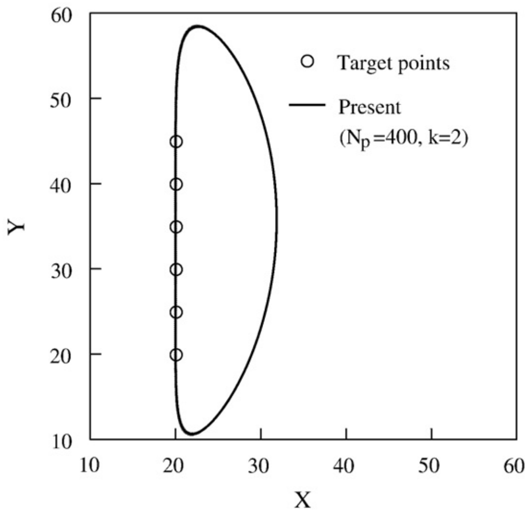

Fig. 3. Six target points and the coupler curve obtained using the proposed method with $N_p = 400$ and $k = 2$ for case 1.

Table 2
Synthesized results for case 2.

	Present		BGA [17]	DE [17]
	$N_p = 100, k = 1$	$N_p = 400, k = 2$	$N_p = 100$	$N_p = 100$
N^o	100,000	400,000	200,000	100,000
r_1	50.0	50.0	50.0	50.0
r_2	5.0	5.0	9.164414	5.0
r_3	6.97009	7.03102	16.858082	5.905345
r_4	48.1993	48.1342	50.0	50.0
r_{cx}	17.045	16.9767	38.458872	18.819312
r_{cy}	12.638	12.952	0.090117	0.0
x_0	12.2377	12.1975	32.328282	14.373772
y_0	−15.8332	−15.9981	29.537054	−12.444295
θ_0	0.0508453	0.0428286	0.877212	0.463633
f_{obj}	2.58286	2.58036	20922.2	5.52069
e_m	0.648358	0.647818	59.0389	0.938757

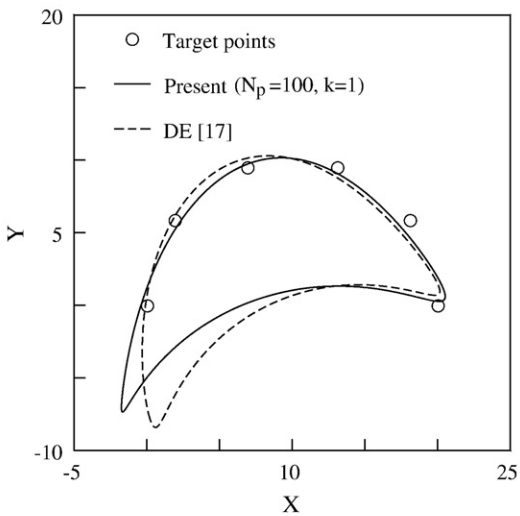

Fig. 4. Six target points and the coupler curve obtained using the proposed method with $N_p = 100$ and $k = 1$ for case 2, together with that obtained by Acharyya and Mandal.

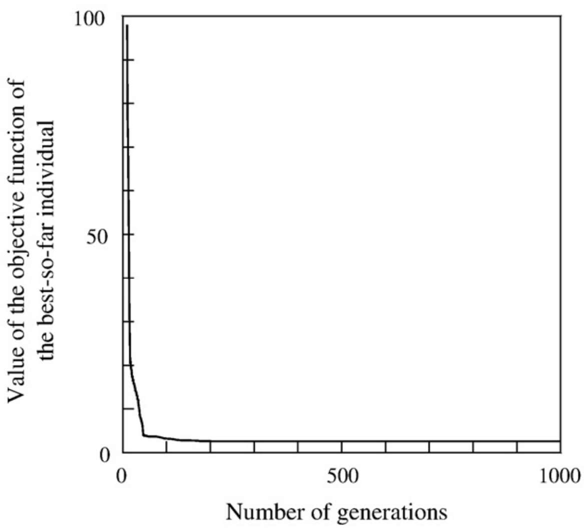

Fig. 5. Typical evolution of the value of the objective function of the best-so-far individual for case 2 corresponding to the number of generations obtained with the proposed evolutionary algorithm.

Table 3
Synthesized results for case 3.

	Present		BGA [17]	DE [17]
	$N_p = 100, k = 1$	$N_p = 400, k = 2$	$N_p = 100$	$N_p = 100$
N^o	100,000	400,000	200,000	100,000
r_1	80.0	80.0	79.981513	54.360893
r_2	8.24689	8.42032	9.109993	8.683351
r_3	45.8968	51.3426	72.936511	34.318634
r_4	58.5404	42.4532	80.0	79.996171
r_{cx}	−6.40389	−9.3741	0.0	0.000187
r_{cy}	−9.12264	5.06098	0.0	1.46525
x_0	6.52409	5.53372	10.155966	10.954397
y_0	20.522	0.477183	10.0	11.074534
θ_0	0.136532	4.28177	0.026149	2.12965
θ_2^1	6.05991	2.0935	6.283185	6.283185
θ_2^2	0.488453	2.81291	0.600745	0.616731
θ_2^3	1.17805	3.51605	1.372812	1.310254
θ_2^4	1.88339	4.20638	2.210575	2.19357
θ_2^5	2.59806	4.89051	2.862639	2.91717
θ_2^6	3.28585	5.57398	3.420547	3.490746
θ_2^7	3.96674	6.26458	4.072611	4.132017
θ_2^8	4.65966	0.676189	4.910373	4.922075
θ_2^9	5.35231	1.38307	5.68244	5.695372
θ_2^{10}	6.06263	2.09348	6.283185	6.28297
f_{obj}	3.11511×10^{-2}	6.02203×10^{-4}	5.77309	2542.25
e_m	4.82473×10^{-2}	6.74935×10^{-3}	7.00454×10^{-1}	15.8445

$$\text{Target points}: C_d^i = \begin{bmatrix} (20,10),(17.66,15.142),(11.736,17.878),(5,16.928),(0.60307,12.736), \\ (0.60307,7.2638),(5,3.0718),(11.736,2.1215),(17.66,4.8577),(20,10) \end{bmatrix}$$

$$\text{Limits of the variables}: r_1,r_2,r_3,r_4 \in [5,80];\ r_{cx},r_{cy},x_0,y_0 \in [-80,80];\ \theta_0,\theta_2^1,\theta_2^2...\theta_2^{10} \in [0,2\pi].$$

The synthesized geometric parameters and the corresponding values of the objective function and the mean error are shown in Table 3, together with the synthesized solutions obtained by Acharyya and Mandal. Although the constraint of the sequence of the input angles during the evolution is ignored in this case, there is no order defect in the best synthesized mechanisms, as can be seen in Table 3. Obviously, the DE solution is unsound. There is remarkable improvement in the accuracy of the solution for case 3 using the present method. The mean error for $N_p = 100$ and $k = 1$ obtained in this work shows a decline by about 93% as compared with the BGA solution. Moreover, the BGA requires many more generations for convergence. As can be seen in Table 3, the solution for $N_p = 400$ and $k = 2$ is much more accurate than that for $N_p = 100$ and $k = 1$. Fig. 6 shows the ten target points and the coupler curve obtained using the proposed method with $N_p = 100$ and $k = 1$ for case 3, together with that obtained by Acharyya and

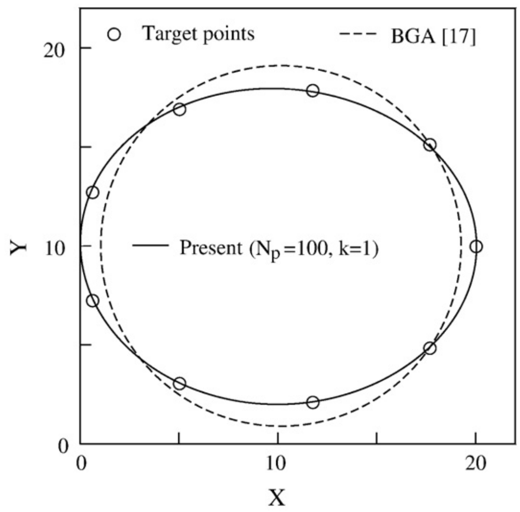

Fig. 6. Ten target points and the coupler curve obtained using the proposed method with $N_p = 100$ and $k = 1$ for case 3, together with that obtained by Acharyya and Mandal.

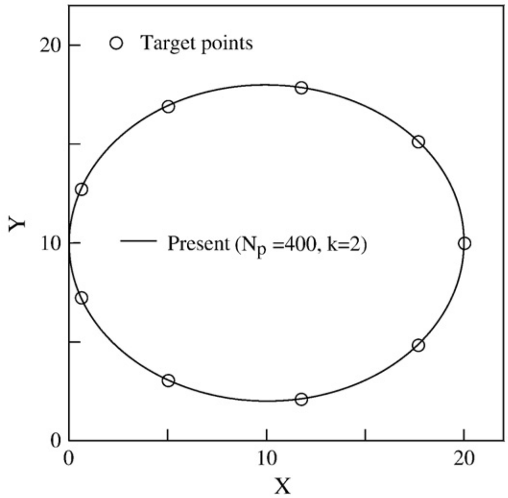

Fig. 7. Ten target points and the coupler curve obtained using the proposed method with $N_p = 400$ and $k = 2$ for case 3.

Mandal. Fig. 7 shows the ten target points and the coupler curve obtained using the proposed method with $N_p = 400$ and $k = 2$ for case 3.

4.4. Case 4: 18 target points and 10 design variables

This case is a path synthesized problem with 18 target points and prescribed timing.

$$\text{Design variables}: \mathbf{X} = \left[r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, x_0, y_0, \theta_0, \theta_2^1\right]$$

$$\text{Target points}: C_d^i = \begin{bmatrix} (0.5,1.1),(0.4,1.1),(0.3,1.1),(0.2,1.0),(0.1,0.9),(0.005,0.75),(0.02,0.6),(0.0,0.5),(0.0,0.4), \\ (0.03,0.3),(0.1,0.25),(0.15,0.2),(0.2,0.3),(0.3,0.4),(0.4,0.5),(0.5,0.7),(0.6,0.9),(0.6,1.0) \end{bmatrix} \text{ and } \theta_2^i = \left[\theta_2^1, \theta_2^1 + 20^\circ * i\right] (i = 1, 2 \ldots 17)$$

$$\text{Limits of the variables}: r_1, r_2, r_3, r_4 \in [0, 50];\ r_{cx}, r_{cy}, x_0, y_0 \in [-50, 50]; \theta_0 \theta_2^i \in [0, 2\pi].$$

The synthesized geometric parameters and the corresponding values of the objective function and the mean error are shown in Table 4, together with the synthesized solutions obtained by Kunjur and Krishnamurty and Cabrera et al. The mean error for $N_p = 100$ and $k = 1$ increases by 14% and 34% compared to the solutions obtained by Cabrera et al. and the exact gradient method, respectively. As expected, and as shown in Table 4, the solution for $N_p = 100$ and $k = 1$ has been improved by using $N_p = 400$ and $k = 2$. Fig. 8 shows the eighteen target points and the coupler curve obtained using the proposed method with $N_p = 100$ and $k = 1$ for case 4, together with the curves obtained by Kunjur and Krishnamurty and Cabrera et al. Fig. 9 shows the eighteen target points and the coupler curve obtained using the proposed method with $N_p = 400$ and $k = 2$ for case 4, together with that obtained by Kunjur and Krishnamurty.

Table 4
Synthesized results for case 4.

	Present		[15] $N_p = 100$	Exact gradient [16]
	$N_p = 100, k = 1$	$N_p = 400, k = 2$		
N^o	5000	400,000	5000	240
r_1	49.9592	47.4379	3.057878	2.85452
r_2	0.218612	0.32477	0.237803	0.36355
r_3	42.4842	0.472857	4.828954	2.91374
r_4	32.7470	47.3093	2.056456	0.49374
r_{cx}	−47.9660	0.118748	0.767038	1.031223
r_{cy}	15.3586	−0.319924	1.850828	1.717471
x_0	44.1758	0.526988	1.776808	0.95928
y_0	−23.9643	0.72393	−0.641991	−1.19645
θ_0	5.37543	3.32029	1.002168	0.76398
θ_2^1	1.88551	3.51233	0.226186	0.51172
f_{obj}	4.61271×10^{-2}	1.08613×10^{-2}	3.48391×10^{-2}	1.09034×10^{-2}
e_m	4.78422×10^{-2}	2.2283×10^{-2}	4.21508×10^{-2}	2.26716×10^{-2}

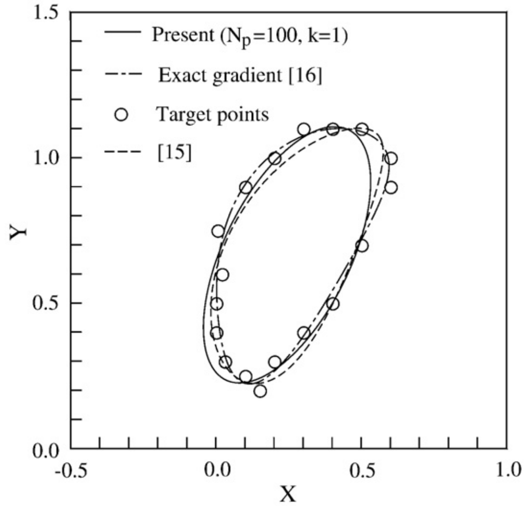

Fig. 8. Eighteen target points and the coupler curve obtained using the proposed method with $N_p = 100$ and $k = 1$ for case 4, together with the curves obtained by Kunjur and Krishnamurty and Cabrera et al.

5. Conclusions

A GA–DE hybrid algorithm is presented to be applied to the problem of four-bar linkage synthesis. The main advantage of this evolutionary algorithm is that it is simple and easy to implement to efficiently solve complicated real-world optimization problems, with no need of deep knowledge of the searching space. The easy 2D moveable sketch method (SolidWorks®) can be used to check whether there are circuit defects or the synthesized mechanism is incapable of motion. The GA–DE hybrid algorithm is produced by combining the real-valued genetic algorithm with the differential evolution. Both the main perturbation of the differential vectors and the minor perturbation of the mutation can be used as genetic operators; the blending crossover operator might thus be unnecessary. For problems without prescribed timing, the constraint of the sequence of the input angles may be ignored by rearranging them to satisfy CW or CCW rotation during initialization and during mutation. The results for cases 1–3 show that much more accurate solutions are obtained with GA–DE hybrid algorithm than with other evolutionary methods discussed in the literature. However, this work cannot lead us to claim that the proposed evolutionary algorithm outperforms other state-of-the-art evolutionary algorithms in all possible real-world optimization problems. The conclusion we draw is that the proper combination between/among evolutionary algorithms has potential to solve the optimization problem more effectively. The present work may be extended to solve other mechanism design problems. Improvement of the proposed evolutionary algorithm might be another direction for future work.

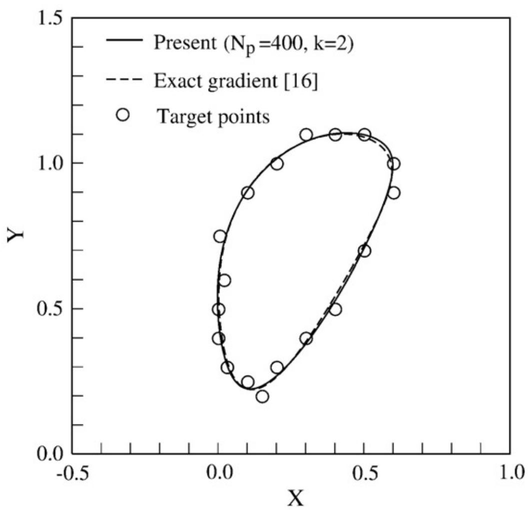

Fig. 9. Eighteen target points and the coupler curve obtained using the proposed method with $N_p = 400$ and $k = 2$ for case 4, together with that obtained by Kunjur and Krishnamurty.

Acknowledgements

This work was supported by the National Science Council of the Republic of China under the contract NSC 98-2221-E-237-004.

References

[1] G.N. Sandor, A General Complex Number Method for Plane Kinematic Synthesis with Application, Ph.D. Thesis, Columbia University, New York, 1959.
[2] A.G. Erdman, G.N. Sandor, Advanced Mechanism Design: Analysis and Synthesis, Vol. 2, Prentice-Hall, Upper Saddle River, NJ, 1984.
[3] A.P. Morgan, C.W. Wampler, Solving a planar fourbar design problem using continuation, Journal of Mechanical Design 112 (4) (1990) 544.
[4] C. Han, A general method for the optimum design of mechanisms, Journal of Mechanisms 1 (1966) 301–313.
[5] S. Krishnamurty, D.A. Turcic, Optimal synthesis of mechanisms using nonlinear goal programming techniques, Mechanisms and Machine Theory 27 (5) (1992) 599–612.
[6] J. Mariappan, S. Krishnamurty, A generalized exact gradient method for mechanism synthesis, Mechanism and Machine Theory 31 (1996) 413–421.
[7] Y. Hongying, T. Dewei, W. Zhixing, Study on a new computer path synthesis method of a four-bar linkage, Mechanism and Machine Theory 42 (2002) 383–392.
[8] J. Buskiewicz, R. Starosta, T. Walczak, On the application of the curve curvature in path synthesis, Mechanism and Machine Theory 44 (2009) 1223–1239.
[9] J.H. Holland, Adaptation in Natural and Artificial System, The University of Michigan press, Michigan, 1975.
[10] K. A. DeJong, An Analysis of the Behavior of a Class of Genetic Adaptive System, Ph.D. Thesis, University of Michigan, Michigan, 1975.
[11] R. Storn, K. Price, Differential evolution. A simple and efficient heuristic scheme for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.
[12] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks (1995) 1942–1948.
[13] W.Y. Lin, K.M. Hsiao, Study on improvements of the five-point double-toggle mould clamping mechanism, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 218 (2004) 761–774.
[14] W.Y. Lin, C.L. Shen, K.M. Hsiao, A case study of the five-point double-toggle mould clamping mechanism, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 220 (2006) 527–535.
[15] J.A. Cabrera, A. Simon, M. Prado, Optimal synthesis of mechanisms with genetic algorithms, Mechanism and Machanism and Machine Theory 37 (2002) 1165–1175.
[16] A. Kunjur, S. Krishnamurty, Genetic algorithms in mechanical synthesis, Journal of Applied Mechamisms and Robotics 4 (2) (1997) 18–24.
[17] S.K. Acharyya, M. Mandal, Performance of EAs for four-bar linkage synthesis, Mechanism and Machine Theory 44 (2009) 1784–1794.
[18] K. Price, R. Storn, J. Lampinen, Differential Evolution — a Practical Approach to Global Optimization, Springer-Verlag, Berlin Heidelberg, 2005.
[19] R.L. Norton, Design of Machinery, third ed.McGraw-Hill, New York, 2002.
[20] F. Fredenstein, An analytical approach to the design of four-link mechanisms, Transactions of the ASME 76 (1954) 483–492.
[21] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution programs, 2nd ed.Springer-Verlag, New York, 1994.
[22] L.J. Eshelman, D.J. Shaffer, Real-coded genetic algorithms and interval-schemata, in: D.L. Whitley (Ed.), Fundations of Genetic Algorithms 2, San Mateo, CA, Morgan Kaufman, 1993, pp. 187–202.
[23] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 9 (2) (1995) 115–148.
[24] K. Deb, A. Kumar, Real-coded genetic algorithms with simulated binary crossover: studies on multi-modal and multi-objective problems, Complex Systems 9 (6) (1995) 431–454.
[25] H.M. Voigt, H. Muhlenbein, D. Cvetkovic, Fuzzy recombination for the breeder genetic algorithm, Proceedings of 6th International Conference on Genetic Algorithms (1995) 104–111.
[26] H.G. Beyer, K. Deb, On the desired behavior of self-adaptive evolutionary algorithms, Parallel Problem Solving from Nature VI, 2000, pp. 59–68.
[27] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, West Sussex, 2001.
[28] D.E. Goldberg, Genetic algorithms in search, Optimization and Machine Learning, Addison Wesley, Massachusetts, 1989.