# Advanced Algorithms

Esmaeil Nourani

WWW.Nurani.Ir
info@nurani.ir



## Course Outline

- **Local search Algorithms**
  - Gradient-Based local optimization methods
  - Random Search
  - Hill climbing, stochastic Hill clim., Iterative Hill clim.
  - Simulated Annealing
  - Tabu Search

- Hw#1 Stochastic HC , S.A (1.5 pts)
- Hw#2 T.S (1.5 pts)

# Course Outline(Cont.)

- **Meta-Heuristics**
  - Introduction to Population based Algorithm
  - EA/ES/Genetic Algorithms
  - EA and TSP
  - Constrained Evolutionary Optimization
  - Ant Colony Optimization
  - Particle Swarm Optimization
  - Bee Algorithm

- Hw#3, Genetic Algorithm (1.5 pts)
- Hw#4, ACO (1.5 pts)
- Hw#5, PSO (1.5 pts)



# Text Books

1. **Essentials of Metaheuristics, A Set of Undergraduate Lecture Notes by Sean Luke, George Mason University, Second Edition 2013**

2. **Metaheuristics for Hard Optimization, Methods and Case Studies Dréo, J., Pétrowski, A., Siarry, P., Taillard, E., 2003**

# Gradient-Based local optimization methods and Random Search

Lecture # 1



If we can really understand the problem
the answer will come out of it,
because the answer is not separate from
the problem

Krishnamurti

# Metaheuristics

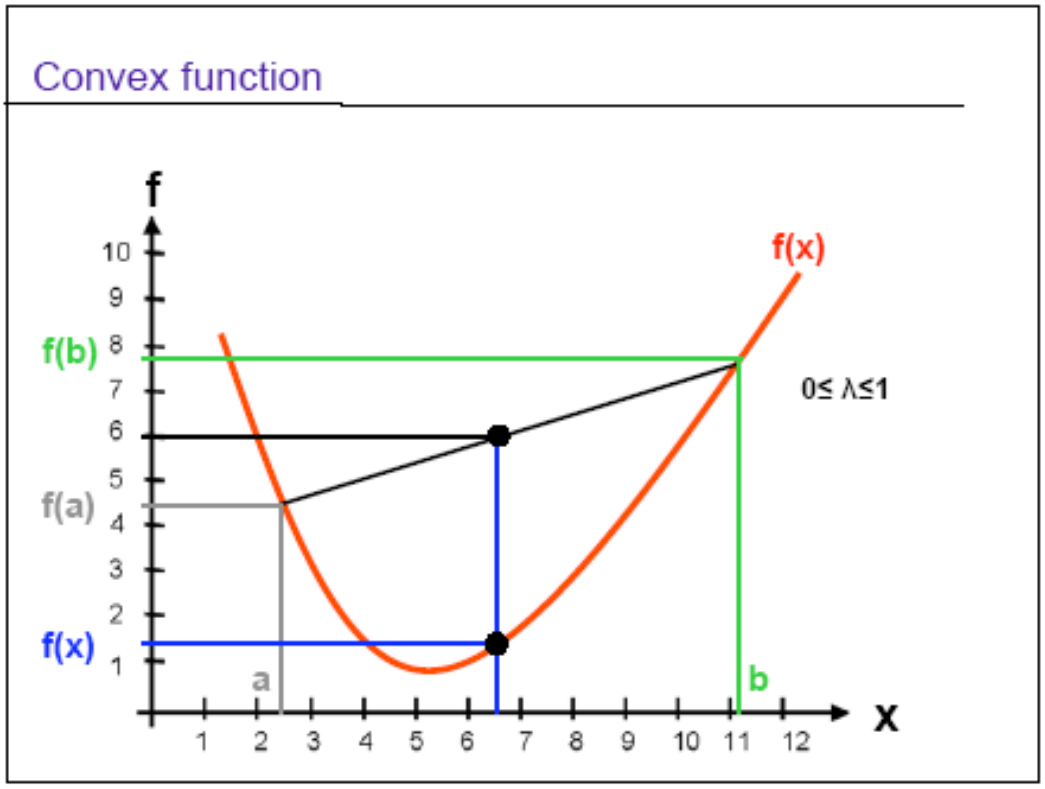
- Stochastic optimization is the general class of algorithms and techniques which employ some degree of randomness to find optimal (or as optimal as possible) solutions to hard problems.
- Metaheuristics are the most general of these kinds of algorithms, and are applied to a very wide range of problems.



# Gradient Optimization

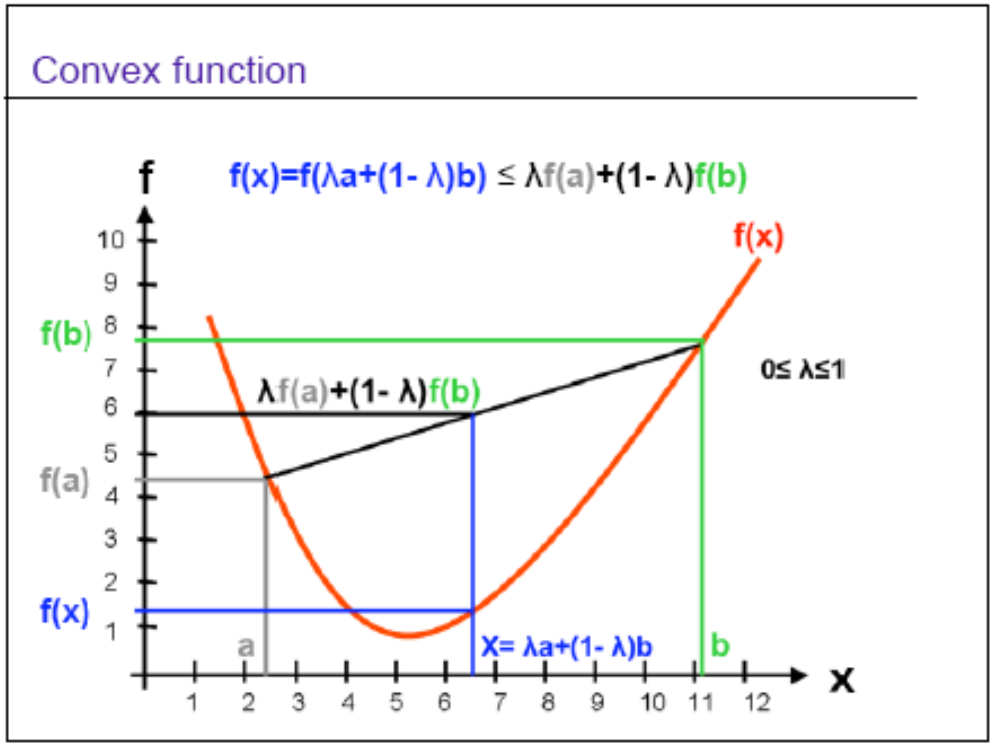
- Convex functions
- Convex sets
- Reminder: derivative
- Gradient descent
- Problems of gradient descent

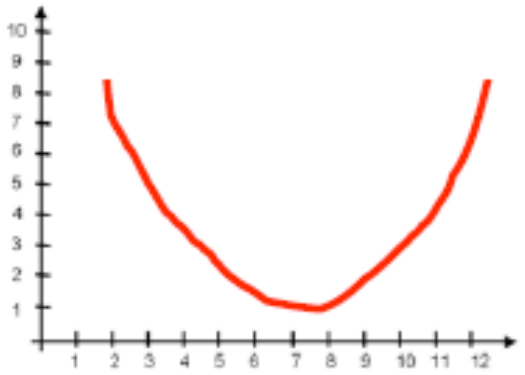
# Convex function



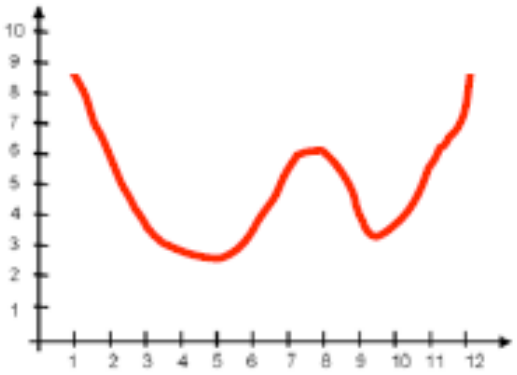



# Convex function





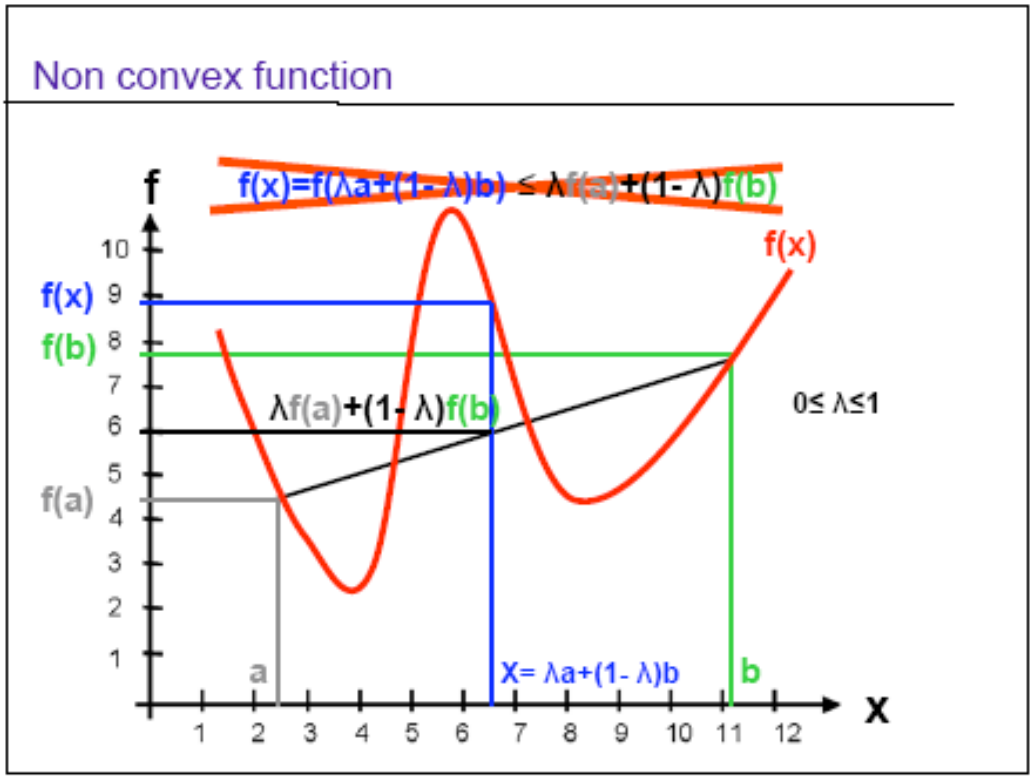

Convex function




Non convex function

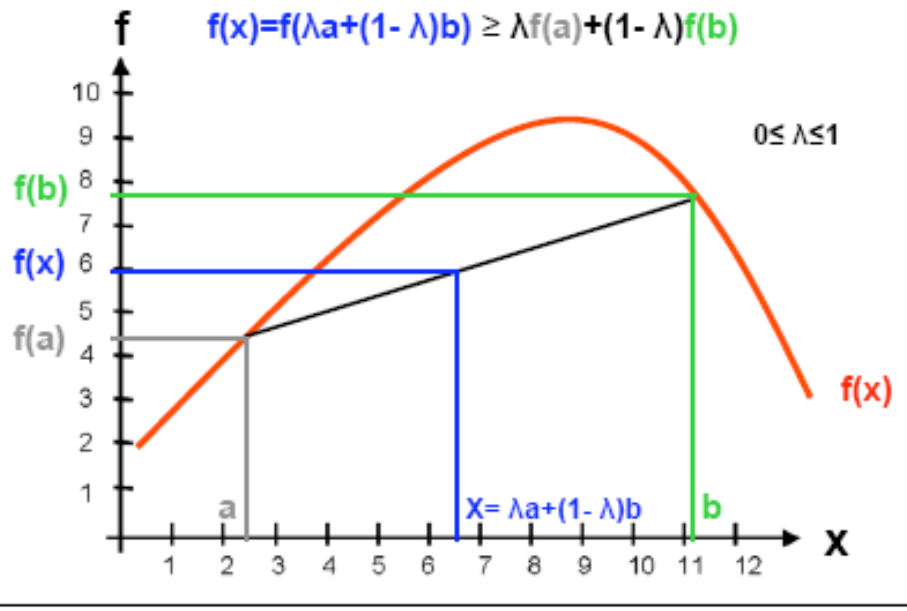


# Non convex function



# Concave function

## Concave function






# Convexity

## Definitions / facts, common mistakes

Definitions/facts

- If f is convex, -f is concave
- If f is concave, -f is convex

Common mistakes

- A nonconvex function is not necessarily concave
- A nonconcave function is not necessarily convex
- A function can be neither concave neither convex
- A function can be concave and convex

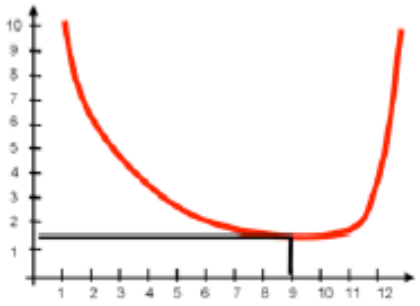
## Convex optimization programs

$$\begin{aligned} \textbf{min:} \quad & f(x) \\ \textbf{s.t.} \quad & g(x) \leq 0 \end{aligned}$$

f and g are convex functions, defined on convex sets

Convex optimization programs are "easy" problems, compared to general optimization programs

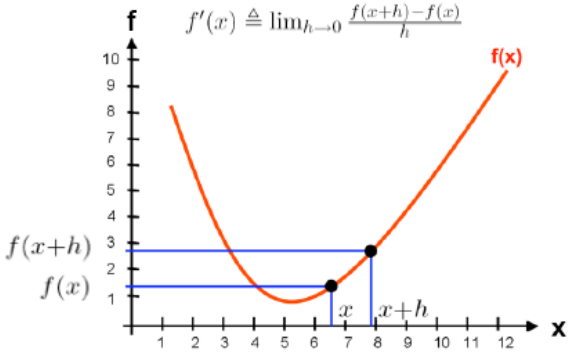
Convex function



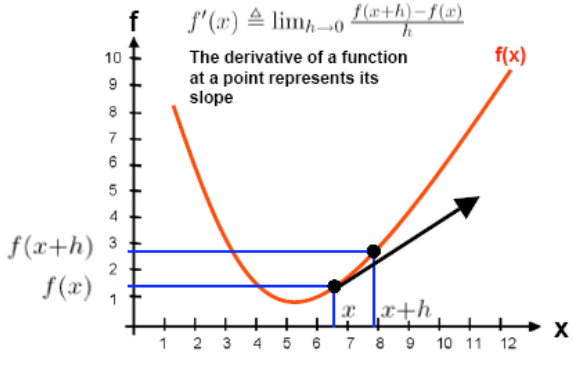

Local minimum is a global minimum



## Reminder: derivative




## Reminder: derivative



## Gradient descent (conceptual description)

**You want to find the minimum of a function, starting from a guess, assuming that you cannot depict the graph of the function, for example the following function**

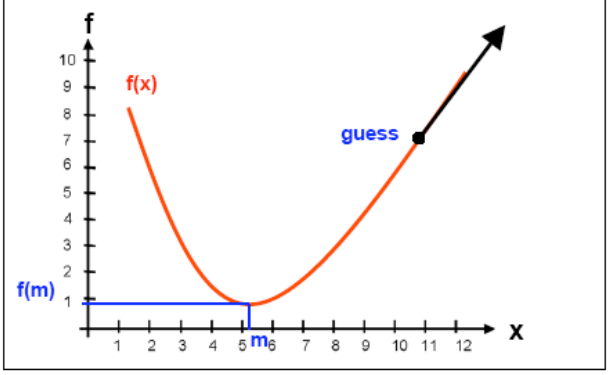
$$f(x) = \exp(\sin(x^2)) + \sqrt{x^4+3}\sin(\exp(-\frac{1}{(1+e|x|)}))$$

**Idea:**

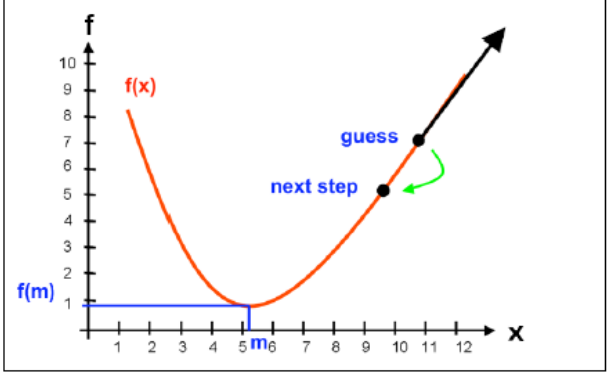
1) **Make a guess**
2) **Compute the derivative at this point (i.e. the slope)**
3) **Follow the direction of the slope (i.e. descend)**
4) **Stop when the slope is zero, i.e. it does not go downhill**



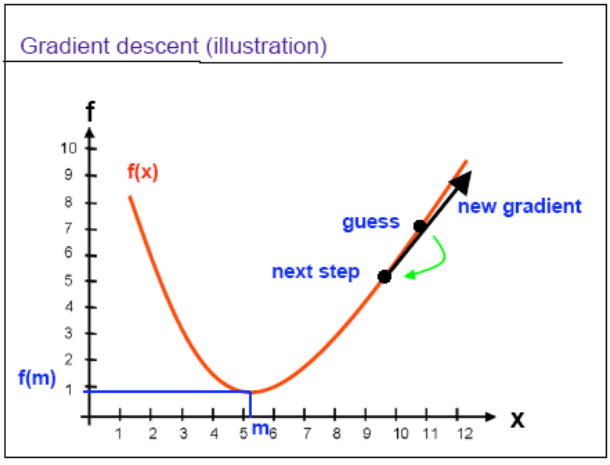
## Gradient descent (illustration)



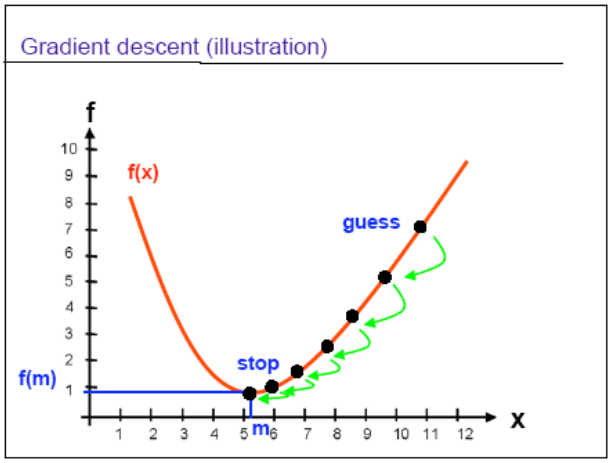

## Gradient descent (illustration)




## Gradient descent (illustration)




## Gradient descent (illustration)



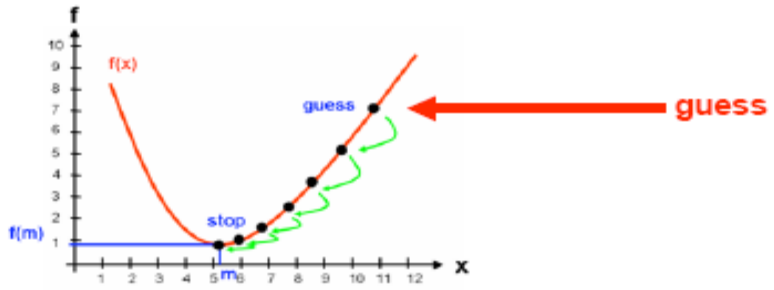
## Gradient descent: algorithm

**Start with a point (guess)**

**Repeat**

- Determine a descent direction
- Choose a step
- Update

**Until stopping criterion is satisfied**





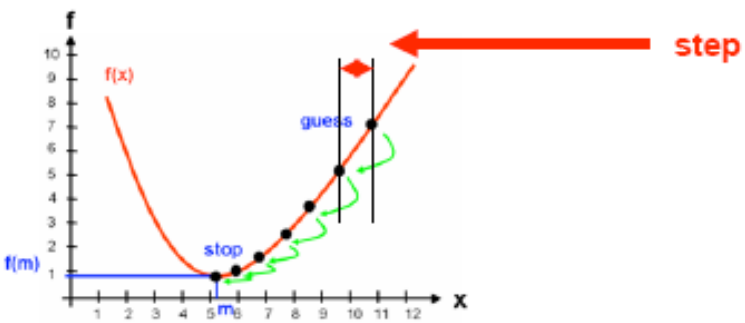

## Gradient descent: algorithm

**Start with a point (guess)**

**Repeat**

- Determine a descent direction
- Choose a step
- Update

**Until stopping criterion is satisfied**



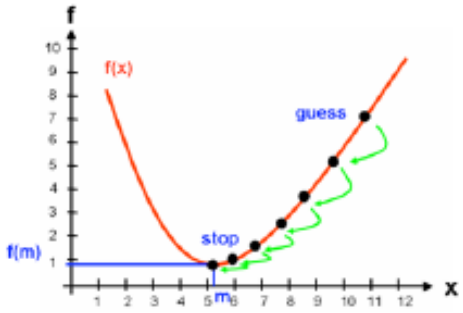
## Gradient descent: algorithm

**Start with a point (guess)**
**Repeat**
- Determine a descent direction
- Choose a step
- Update

**Until stopping criterion is satisfied**

**guess = x**

**direction = -f'(x)**
**step = h > 0**
**x:=x–hf'(x)**
**f'(x)~0**






## Example of 2D gradient: pic of the MATLAB demo

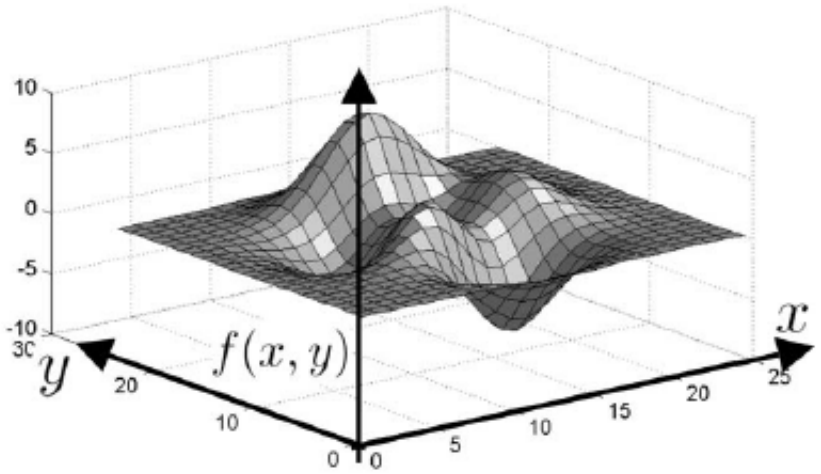
Definition of the gradient in 2D

$$\nabla f(x,y) = \begin{pmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{pmatrix}$$

This is just a genaralization of the derivative in two dimensions.
This can be generalized to any dimension.

## Example of 2D gradient: pic of the MATLAB demo

Illustration of the gradient in 2D





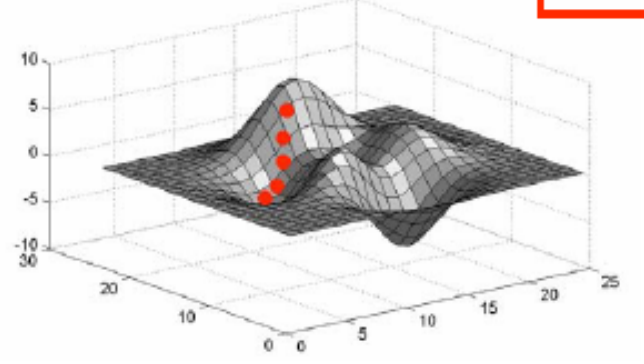

## Generalization to multiple dimensions

**Start with a point (guess)**
**Repeat**
- Determine a descent direction
- Choose a step
- Update

**Until stopping criterion is satisfied**

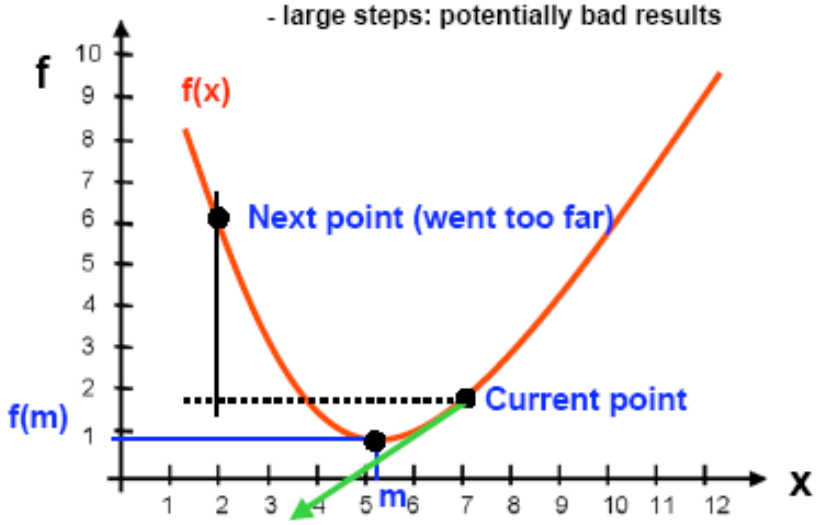
**guess = x**

**direction = -f(x)**
**step = h > 0**
**x:=x–h ∇f'(x)**
**∇f'(x)~0**



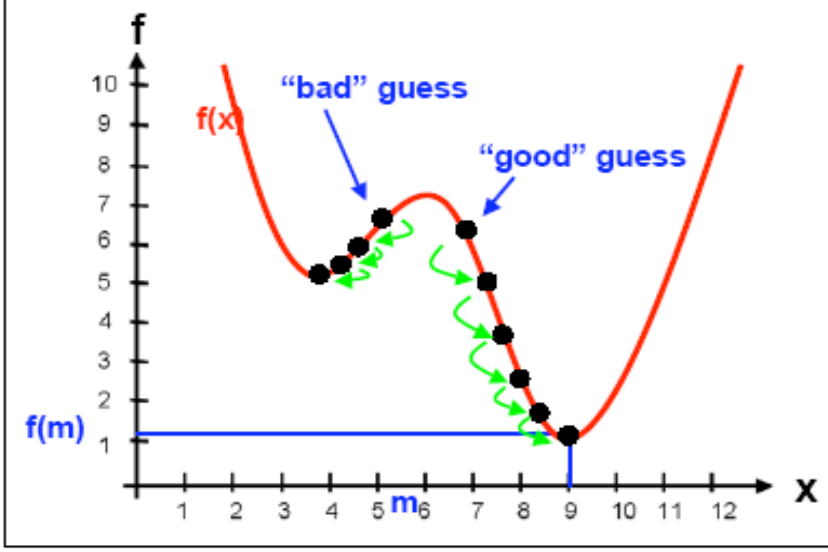
## Problem 1: choice of the step

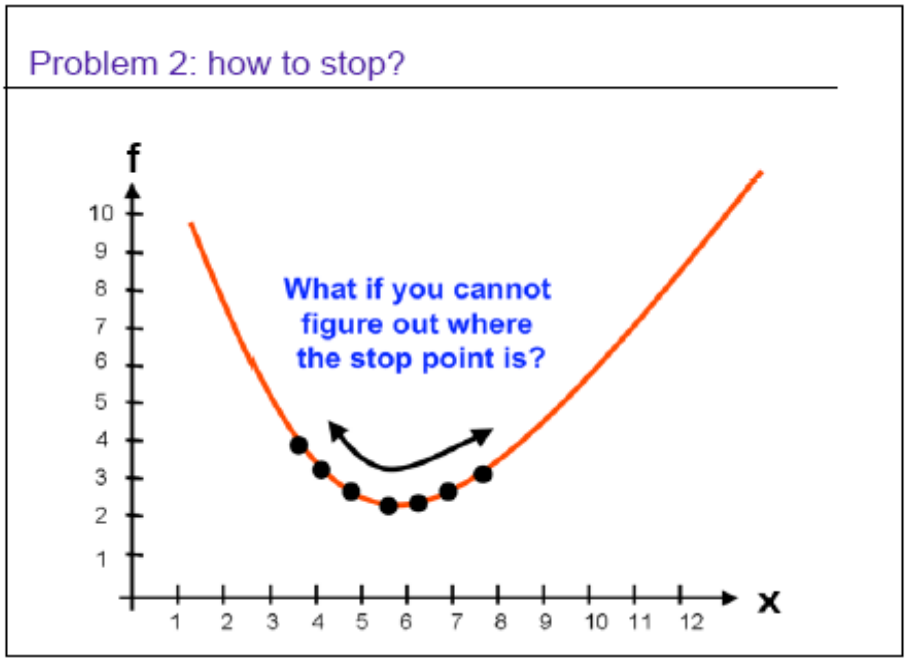
When updating the current computation:
- small steps: inefficient
- large steps: potentially bad results






## Problem 1: non convex function








# Randomized Algorithms

Randomized (probabilistic) algorithms can be

- Nondeterministic ---they can make random but correct decisions : the same algorithm may behave differently when it is applied twice to the same instance of a problem.

- Not very precise sometimes ---usually the more time is given, the better precision can be obtained

# Randomized Algorithm

- **A randomized algorithm is one that makes random choices during the execution**



# What

What is a random search algorithm?

# What

What is a random search algorithm?

A random search algorithm refers to an algorithm that uses some kind of randomness or probability (typically in the form of a pseudo-random number generator) in the definition of the method, and in the literature, may be called a Monte Carlo method or a stochastic algorithm.



# What

- The term metaheuristic is also commonly associated with random search algorithms.
- Simulated annealing, tabu search, genetic algorithms, evolutionary programming, particle swarm optimization, ant colony optimization, multi-start, clustering algorithms, and other random search methods are being widely applied to global optimization problems,

# When

- When you use Random Search Algorithm?



# When

- When you use Random Search Algorithm?

Random search algorithms are useful for ill-structured global optimization problems, where the objective function may be non-convex, non-differentiable, and possibly discontinuous over a continuous, discrete, or mixed continuous-discrete domain.

# Convergence

- In contrast to deterministic methods (such as Dynamic Programming which typically guarantee convergence to the optimum, random search algorithms ensure convergence in probability.



# Why

The tradeoff is in terms of computational effort. Random search algorithms are popular because they can provide a relatively good solution quickly and easily.

# Why

- When an algorithm is confronted by a choice, it is sometimes preferable to choose a course of action at random, rather than spending time to work out which alternative is the best.
- Sometimes we do not have a better method than making random choices
- One advantage of this approach is that if there is more than one correct answer, several different ones may be obtained by running the probabilistic algorithm more than once.



# Large Scale Problem

Random search methods have been shown to have a potential to solve large-scale problems

efficiently in a way that is not possible for deterministic algorithms.

Deterministic method for global optimization is NP-hard, there is evidence that a stochastic algorithm can be executed in polynomial time, on the average

# Advantage

Another advantage of random search methods is that they are relatively easy to implement on complex problems with “black-box” function evaluations. Because the methods typically only rely on function evaluations, rather than gradient , they can be coded quickly, and applied to a broad class of global optimization problems.



# Disadvantage

disadvantage of these methods is that they are currently customized to each specific problem

largely through trial and error.

However:

A common experience is that random search algorithms perform well and are “robust” in the sense that they give useful information quickly for ill-structured global optimization problems.

# Categorization type

- global search phase
- local search phase

The global phase can be viewed as an exploration phase aimed at exploring the entire feasible region, while the local phase can be viewed as an exploitation phase aimed at exploiting local information (e.g. gradient).



# Definition

The general global optimization problem (P) used here is defined as,

$$(P) \min_{x \epsilon S} f(x)$$

where x is a vector of n decision variables, S is an n-dimensional feasible region and assumed to be nonempty, and f is a real-valued function defined over S. The goal is to find a value for x contained in S that minimizes f.

# Local Optimum

- **Definition 1 (Local Maximum). A (local) maximum $x_l \in X$ of one (objective) function $f : X \rightarrow R$ is an input element with $f(x_l) \geq f(x)$ for all x neighboring $x_l$.**

  **If $X \subseteq R^n$, we can write:**

  $$\forall x_l \; \exists \varepsilon > 0 : f(x_l) \geq f(x) \; \forall x \in X, \; |x - x_l| < \varepsilon$$

- **Definition 2 (Local Minimum). A (local) minimum $x_l \in X$ of one (objective) function $f : X \rightarrow R$ is an input element with $f(x_l) \leq f(x)$ for all x neighboring $x_l$.**

  **If $X \subseteq R^n$, we can write:**

  $$\forall x_l \; \exists \varepsilon > 0 : f(x_l) \leq f(x) \; \forall x \in X, \; |x - x_l| < \varepsilon$$

- **Definition 3(Local Optimum). A (local) optimum $x_l \in X$ of one (objective) function $f : X \rightarrow R$ is either a local maximum or a local minimum.**



# Global Optimum

- **Definition 4(Global Maximum). A global maximum $x^* \in x$ of one (objective) function $f : X \rightarrow R$ is an input element with $f(x^*) \geq f(x) \; \forall x \in X$.**

- **Definition 5(Global Minimum). A global minimum $x^* \in X$ of one (objective) function $f : X \rightarrow R$ is an input element with $f(x^*) \leq f(x) \; \forall x \in X$.**

- **Definition 6(Global Optimum). A global optimum $x\star \in X$ of one (objective) function $f : X \rightarrow R$ is either a global maximum or a global minimum.**

# The Structure Of Optimization

- **Definition 7 (Problem Space). The problem space X of an optimization problem is the set containing all elements x which could be its solution.**

  **The problem space X is often restricted by:**

  **1. logical constraints**
  **2. practical constraints**

  **with the Java programming language, we can only use 64 bit floating point numbers. With these 64 bit, it is only possible to express numbers up to a certain precision and we cannot have more than 15 or so decimals.**



# The Search Operations

- **Definition 8 (Search Space). The search space G of an optimization problem is the set of all elements g which can be processed by the search operations.**

- **Definition 9 (Search Operations). The search operations are used by optimization algorithms in order to explore the search space G.**

# Puzzle

The are six matches on the table and the task is to construct four equilateral triangles where the length of each side is equal to the length of a match.



If you start with the wrong search space, you will never find the right answer!

# Generic Random Search Algorithm:

- Step 0. Initialize algorithm parameters 0, initial points X(0)ϵ S and iteration index k = 0.
- Step 1. Generate a collection of candidate points V(k+1) ϵ S according to a specific generator.
- Step 2. Update X(k+1) based on the candidate points V(k+1), previous iterates and algorithmic parameters. Also update algorithm parameters k+1.
- Step 3. If a stopping criterion is met, stop. Otherwise increment k and return to Step 1.



# Single-point Generators

Many random search algorithms maintain and generate a single point at each iteration.

# Multiple-point Generators

- Population-based random search algorithms use a collection of current points to generate another collection of candidate points.

  Many of these algorithms are motivated by biological processes, and include genetic algorithms, evolutionary programming, particle swarm optimization and ant colony optimization.



# Update Procedure

- Update Procedure After a candidate point is generated, Step 2 of the generic random search algorithm specifies a procedure to update the current point and algorithm parameters. Algorithms that are strictly improving have a simple procedure, update the current point only if the candidate point is improving but in the most of the case the update is done even if the new solution in not better than the last one.

# Convergence

- Convergence in Probability to the global minimum for general step size algorithms is equal to one. But =>

  with conditions on the method of generating the step length and direction. Essentially, as long as the generator does not consistently ignore any region, then the algorithm will converge with probability one



# Pure Random Search(PRS)

- The simplest and most obvious random search method is a “blind search” or called pure random search.
- Pure random search was first defined in 1958 by Brooks
- Pure random search (PRS) samples repeatedly from the feasible region S, typically according to a uniform sampling distribution. In the context of the generic random search algorithm, each candidate point is generated independently from a uniform distribution on S, and Xk+1 is updated only if the candidate point is improving.

- It can be shown that pure random search converges to within an epsilon distance of the global optimum with probability one

# TSP Pure Random Search

- Consider the TSP with N cities and subsequently (N − 1)! possible points in the domain.
- If there is a unique minimum, then

$$p(y) = 1/(N-1)!$$

and the expected number of PRS iterations to first sample the minimum is (N − 1)!, which explodes in N.



# Pure Adaptive Search(PAS)

- In contrast to pure random search, where each sample is independent and identically distributed, we next consider pure adaptive search, where each sample depends on the one immediately before it.

- Pure adaptive search (PAS) was introduced for convex programs and later analyzed for global optimization problems

# Pure Adaptive Search(PAS)

- PAS by definition, generates a candidate point uniformly in the subset of the domain that is strictly improving. It is an idealized algorithm to show potential performance for a random search algorithm.

- Whereas sampling from a uniform distribution in PRS over S is typically very easy sampling from a uniform distribution in PAS over is very difficult in general.

- However, the analysis shows the value of being able to find points in the improving level set; the number of iterations of PAS is an exponential improvement over the number of iterations in PRS



# Conclusion

- Gradient based optimization

  you can compute the first , second … derivative of your domain
- Random Based optimization

  you do not have any idea of the structure of your search domain.

# Ref

- Slides adapted from Advanced Algorithms course, presented by kourosh ziarati