Evolutionary Algorithm

- Aspect & Prospect

Dr. Md. Aminul Haque Akhand
Dept. of CSE, KUET

Evolutionary Algorithm (EA) Basics

Evolutionary
(Search and Optimization)
Algorithm

Evolutionary Algorithm
for
Search and Optimization

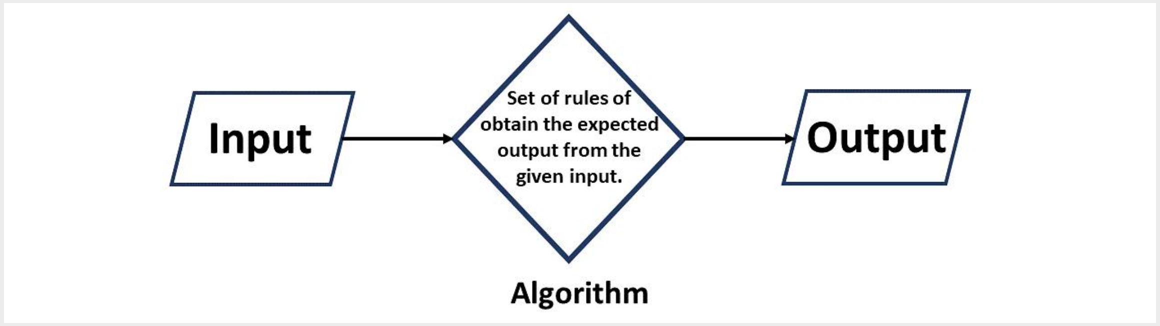
An Evolutionary Algorithm
is a subset of
Evolutionary Computation

What is Algorithm?

- An algorithm is a set of commands that must be followed for a computer to perform calculations or other problem-solving operations.

- According to its formal definition, an algorithm is a finite set of instructions carried out in a specific order to perform a particular task.

- It is not the entire program or code; it is simple logic to a problem represented as an informal description in the form of a flowchart or pseudocode.


Input
Set of rules of obtain the expected output from the given input.
Output
Algorithm

https://www.simplilearn.com/tutorials/data-structure-tutorial/what-is-an-algorithm#what_is_an_algorithm

What is Algorithm?

https://en.wikipedia.org/wiki/Algorithm

- In mathematics and computer science, an algorithm is a finite sequence of rigorous instructions, typically used to solve a class of specific problems or to perform a computation.

- Algorithms are used as specifications for performing calculations and data processing.

- By making use of artificial intelligence, algorithms can perform automated deductions (referred to as automated reasoning) and use mathematical and logical tests to divert the code execution through various routes (referred to as automated decision-making).

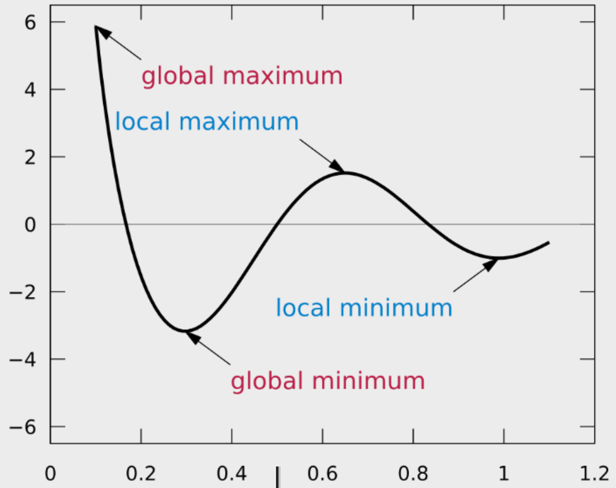
Task: Search and Optimization

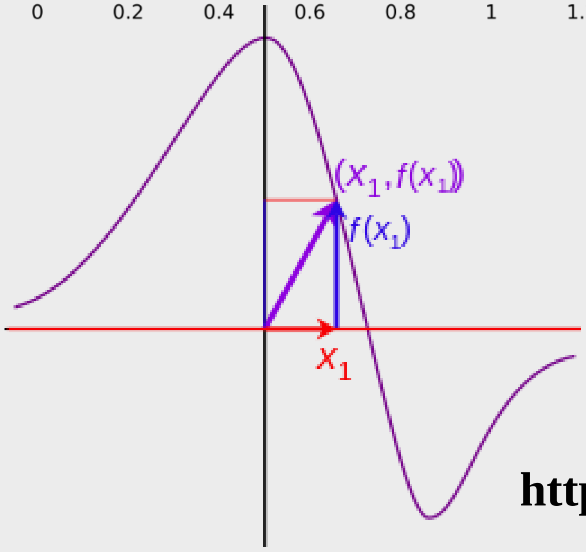
Search (General Definition)
https://www.dictionary.com/browse/search

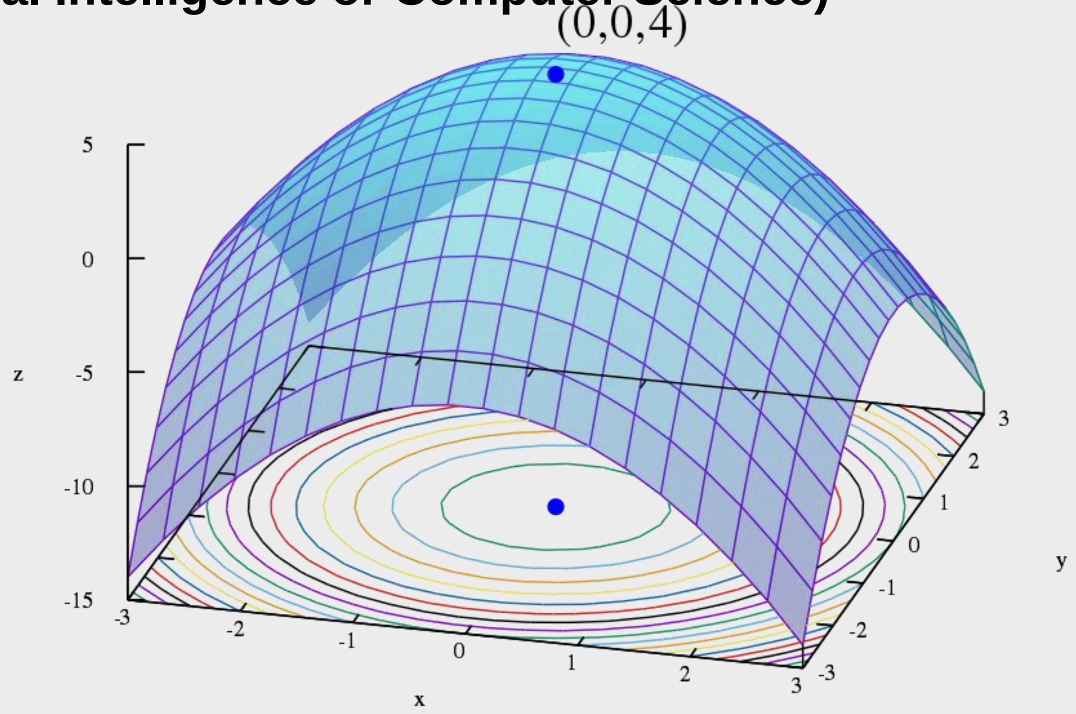
- to go or look through (a place, area, etc.) carefully in order to find something missing or lost: They searched the woods for the missing child. I searched the desk for the letter.
- to look at or examine (a person, object, etc.) carefully in order to find something concealed: The police searched the suspect for weapons.
- to explore or examine in order to discover: They searched the hills for gold.

Task: Search and Optimization

Search (Computational Intelligence or Computer Science)







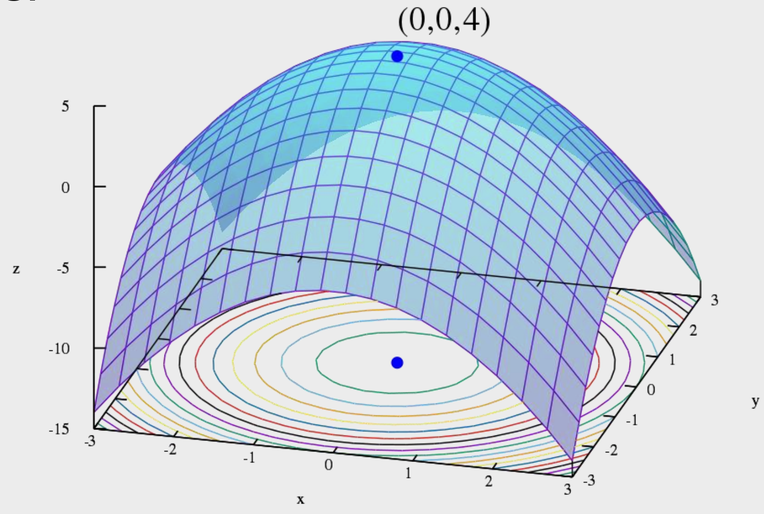
Graph of a given by $z = f(x, y) = -(x^2 + y^2) + 4$. The global maximum at $(x, y, z) = (0, 0, 4)$ is indicated by a blue dot.

https://en.wikipedia.org/wiki/Test_functions_for_optimization

Task: Search and Optimization

Optimization (General) https://www.dictionary.com/browse/optimization

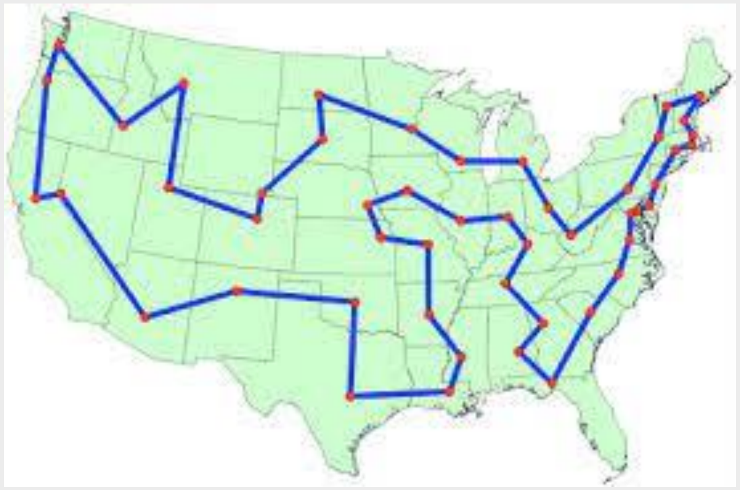
- the fact of optimizing; making the best of anything.
- the condition of being optimized.
- Mathematics: A mathematical technique for finding a maximum or minimum value of a function of several variables subject to a set of constraints, as linear programming or systems analysis.



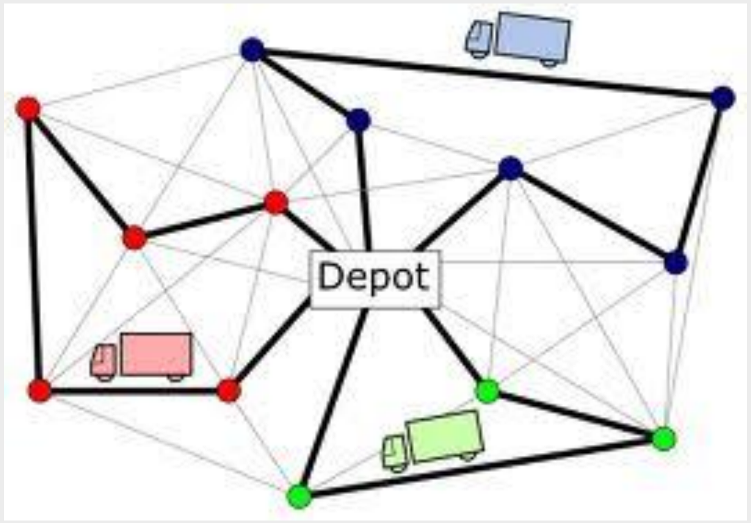
Task: Search and Optimization

Optimization (Computational Intelligence or Computer Science)

Traveling Salesman Problem (TSP)



Vehicle Routing Problem (VRP)



http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html

https://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/

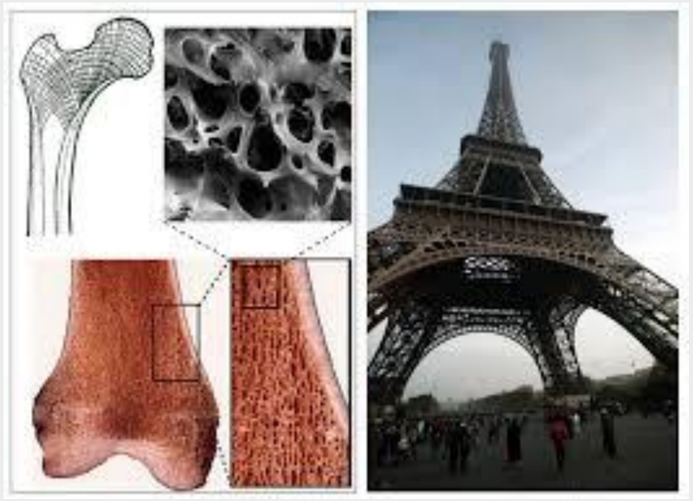
Task: Search and Optimization

Search and Optimization in Real-Life Scenarios

Solving Method: Evolutionary Approach

Techniques taking idea from Natural Phenomena

Biomimicry / Biomimetic: Technology is inspired by nature


Eiffel tower designed on the femur (the human thighbone)

https://biomimicry.org/examples/

The giant water lily that inspired the London Crystal Palace

https://www.youtube.com/watch?v=HppE6ezLDqI

Our Course Concern: Computing Techniques based on Natural Phenomena

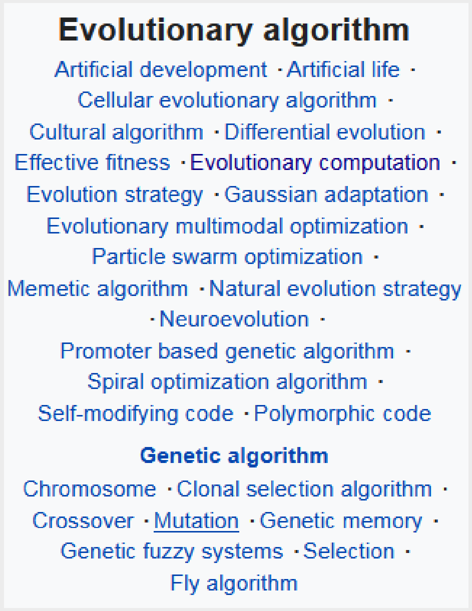
Evolutionary Algorithm

https://en.wikipedia.org/wiki/Evolutionary_algorithm

In computational intelligence (CI), an evolutionary algorithm (EA) is a subset of evolutionary computation, a generic population-based metaheuristic optimization algorithm. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection.

Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions (see also loss function).

Evolution of the population then takes place after the repeated application of the above operators



Evolutionary algorithm

Artificial development · Artificial life · Cellular evolutionary algorithm · Cultural algorithm · Differential evolution · Effective fitness · Evolutionary computation · Evolution strategy · Gaussian adaptation · Evolutionary multimodal optimization · Particle swarm optimization · Memetic algorithm · Natural evolution strategy · Neuroevolution · Promoter based genetic algorithm · Spiral optimization algorithm · Self-modifying code · Polymorphic code

Genetic algorithm

Chromosome · Clonal selection algorithm · Crossover · Mutation · Genetic memory · Genetic fuzzy systems · Selection · Fly algorithm

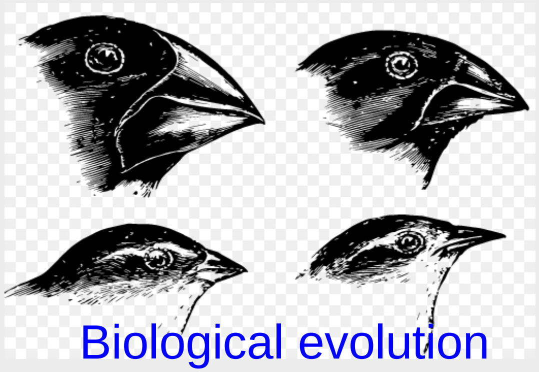
Evolutionary Computation

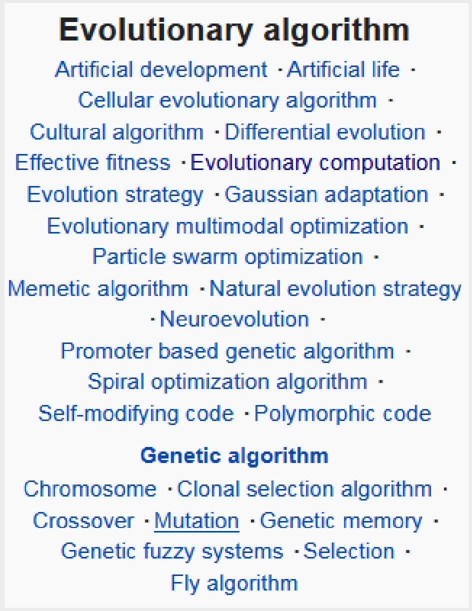
https://en.wikipedia.org/wiki/Evolutionary_computation

In computer science, evolutionary computation is a family of algorithms for global optimization inspired by biological evolution, and the subfield of artificial intelligence and soft computing studying these algorithms. In technical terms, they are a family of population-based trial and error problem solvers with a metaheuristic or stochastic optimization character.

In evolutionary computation, an initial set of candidate solutions is generated and iteratively updated. Each new generation is produced by stochastically removing less desired solutions, and introducing small random changes. In biological terminology, a population of solutions is subjected to natural selection (or artificial selection) and mutation. As a result, the population will gradually evolve to increase in fitness, in this case the chosen fitness function of the algorithm.

Evolutionary computation techniques can produce highly optimized solutions in a wide range of problem settings, making them popular in computer science. Many variants and extensions exist, suited to more specific families of problems and data structures.


Biological evolution

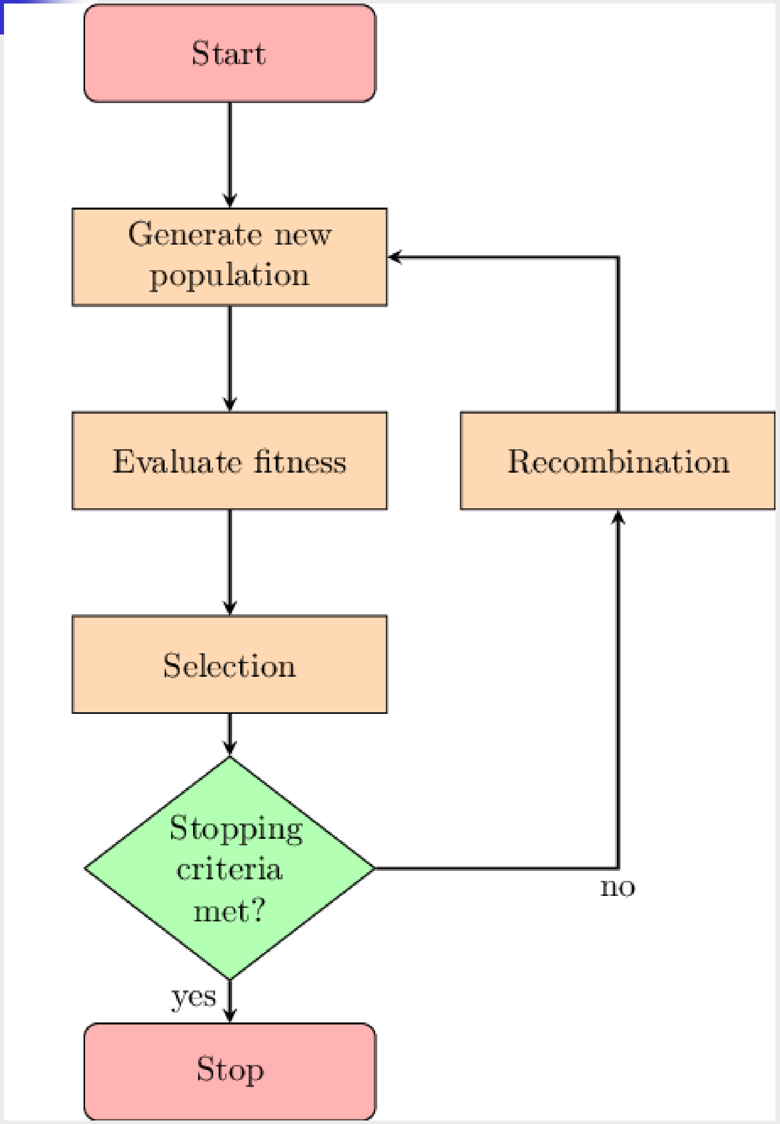

Evolutionary algorithm

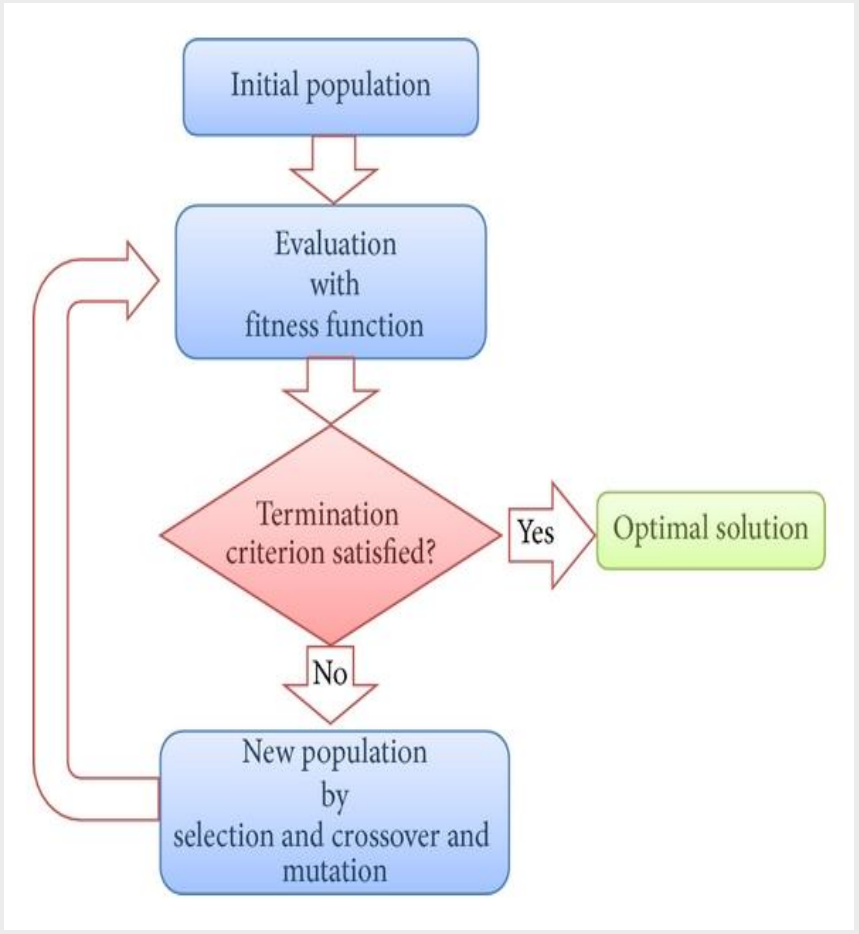
Artificial development · Artificial life · Cellular evolutionary algorithm · Cultural algorithm · Differential evolution · Effective fitness · Evolutionary computation · Evolution strategy · Gaussian adaptation · Evolutionary multimodal optimization · Particle swarm optimization · Memetic algorithm · Natural evolution strategy · Neuroevolution · Promoter based genetic algorithm · Spiral optimization algorithm · Self-modifying code · Polymorphic code

Genetic algorithm

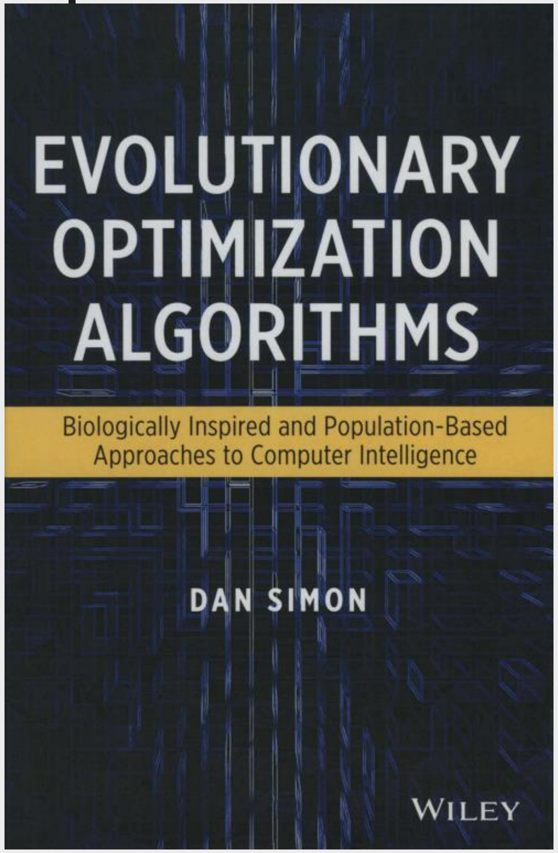
Chromosome · Clonal selection algorithm · Crossover · Mutation · Genetic memory · Genetic fuzzy systems · Selection · Fly algorithm

Evolutionary Algorithm Overview




Initial population
Evaluation with fitness function
Termination criterion satisfied?
Yes
Optimal solution
No
New population by selection and crossover and mutation

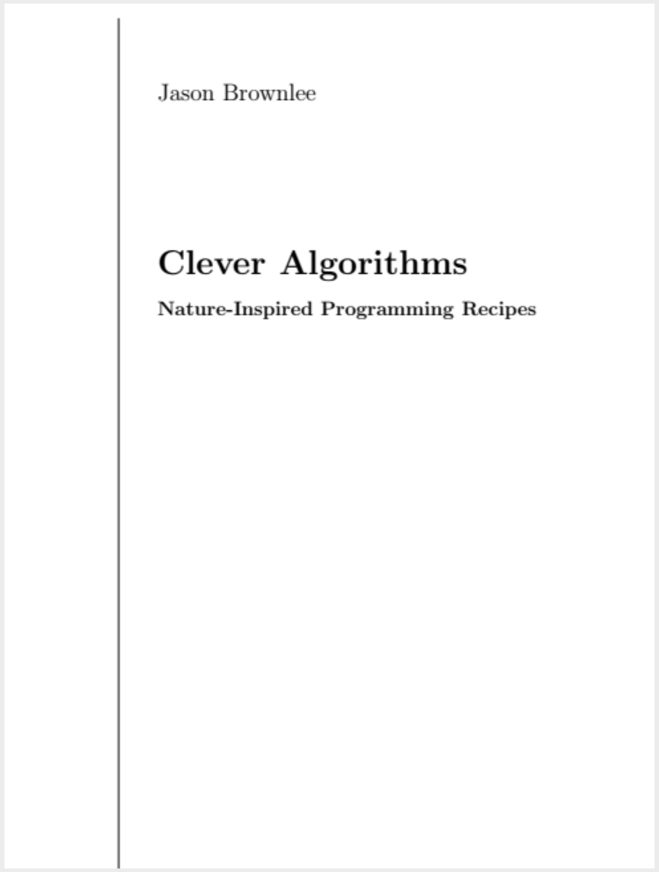
Textbooks


EVOLUTIONARY OPTIMIZATION ALGORITHMS

Biologically Inspired and Population-Based Approaches to Computer Intelligence

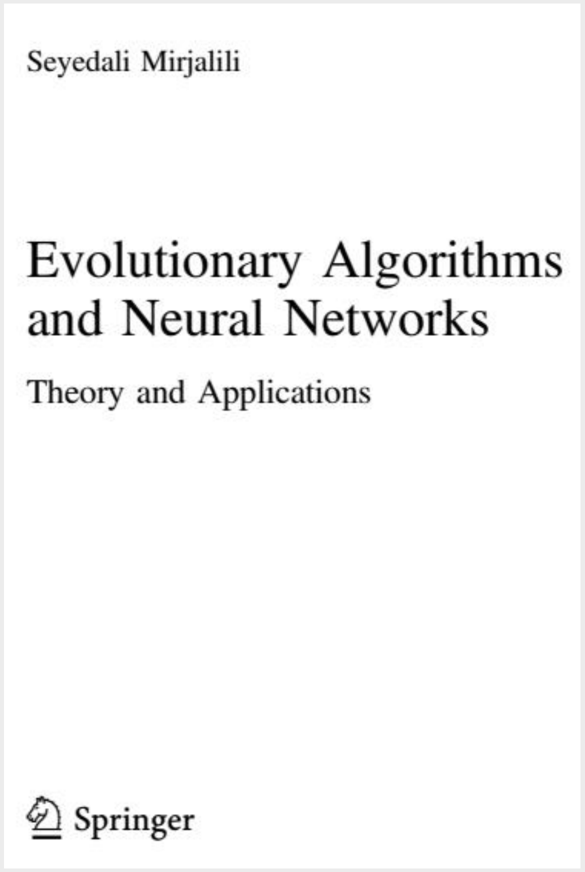
DAN SIMON

WILEY


Jason Brownlee

Clever Algorithms

Nature-Inspired Programming Recipes


Seyedali Mirjalili

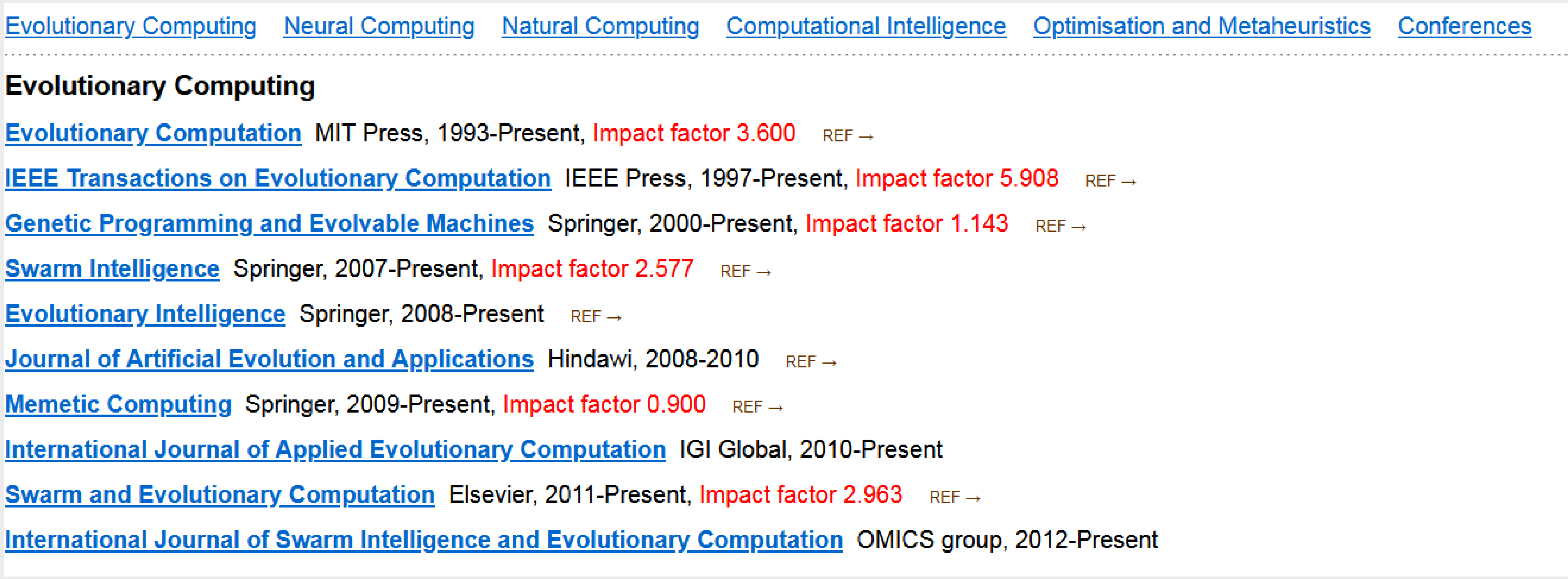
Evolutionary Algorithms and Neural Networks

Theory and Applications

Springer

Research Resources

http://www.macs.hw.ac.uk/~ml355/journals.htm



Evolutionary Computing   Neural Computing   Natural Computing   Computational Intelligence   Optimisation and Metaheuristics   Conferences

Evolutionary Computing

Evolutionary Computation MIT Press, 1993-Present, Impact factor 3.600 REF →

IEEE Transactions on Evolutionary Computation IEEE Press, 1997-Present, Impact factor 5.908 REF →

Genetic Programming and Evolvable Machines Springer, 2000-Present, Impact factor 1.143 REF →

Swarm Intelligence Springer, 2007-Present, Impact factor 2.577 REF →

Evolutionary Intelligence Springer, 2008-Present REF →

Journal of Artificial Evolution and Applications Hindawi, 2008-2010 REF →

Memetic Computing Springer, 2009-Present, Impact factor 0.900 REF →

International Journal of Applied Evolutionary Computation IGI Global, 2010-Present

Swarm and Evolutionary Computation Elsevier, 2011-Present, Impact factor 2.963 REF →

International Journal of Swarm Intelligence and Evolutionary Computation OMICS group, 2012-Present

Open Discussion

Optimization

Dr. Md. Aminul Haque Akhand
Dept. of CSE, KUET

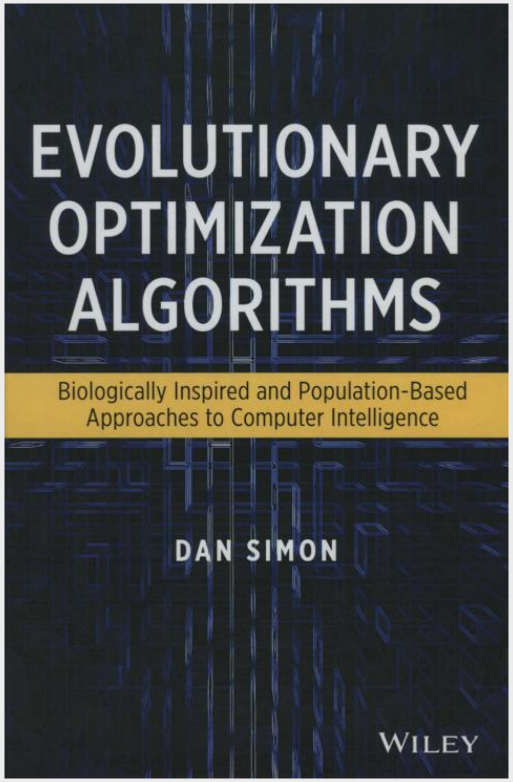
Contents

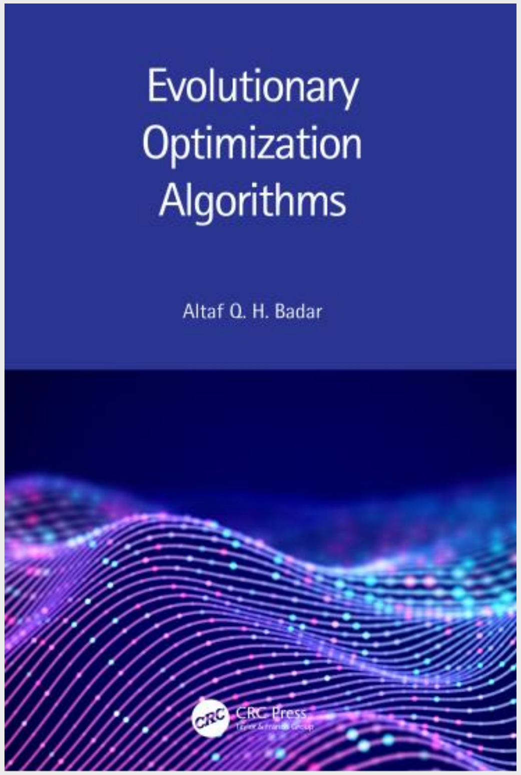
- Optimization Basics
- Unconstrained, Multi-Objective, Multimodal, Combinatorial Optimizations
- Hill Climbing
- Intelligence

Optimization Basics

Optimization saturates what we do and drives almost every aspect of engineering.

—Dennis Bernstein [Bernstein, 2006]


EVOLUTIONARY OPTIMIZATION ALGORITHMS
Biologically Inspired and Population-Based Approaches to Computer Intelligence
DAN SIMON
WILEY



Unconstrained Optimization

- Optimization is applicable to virtually all areas of life. Optimization algorithms can be applied to everything from aardvark breeding to zygote research.
- The possible applications of EAs are limited only by the engineer's imagination, which is why EAs have become so widely researched and applied in the past few decades.

An optimization problem can be written as a minimization problem or as a maximization problem. [Sometimes we try to minimize a function and sometimes we try to maximize a function.]

These two problems are easily converted to the other form:

$$\begin{aligned} \min_x f(x) \quad &\Longleftrightarrow \quad \max_x[-f(x)] \\ \max_x f(x) \quad &\Longleftrightarrow \quad \min_x[-f(x)]. \end{aligned}$$

The number of elements in *x* is called the dimension of the problem. The function *f(x)* is called the objective function, and the vector *x* is called the independent variable, or decision variable.

$$\begin{aligned} \min_x f(x) \quad &\Rightarrow \quad f(x) \text{ is called ``cost'' or ``objective''} \\ \max_x f(x) \quad &\Rightarrow \quad f(x) \text{ is called ``fitness'' or ``objective.''} \end{aligned}$$

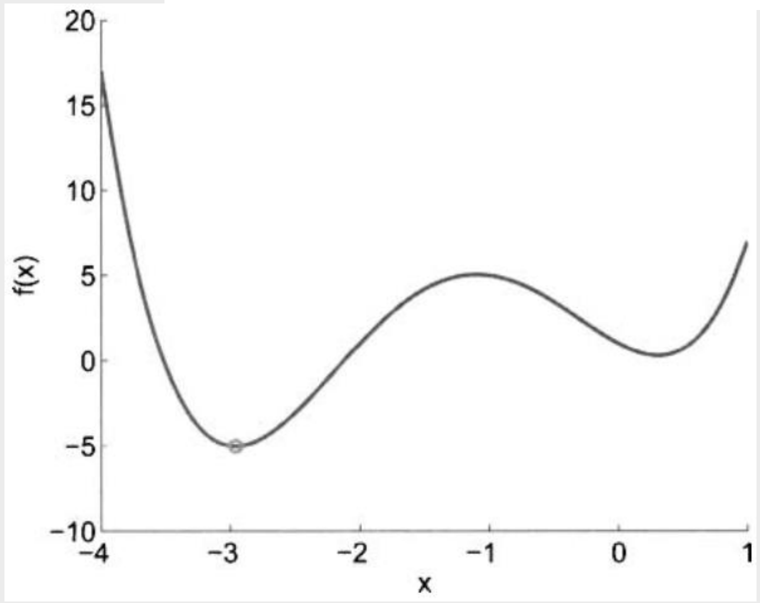
Unconstrained Optimization (Example)

$$f(x, y, z) = (x - 1)^2 + (y + 2)^2 + (z - 5)^2 + 3. \quad (2.3)$$

- *f*(x, *y, z)* is called the objective function or the cost function.
- We can change the problem to a maximization problem by defining *g(x,y,z)* = -*f(x,y,z)* and trying to maximize *g(x,y,z)*. The function *g(x,y,z)* is called the objective function or the fitness function.
- The solution to the problem *min f*(*x,y,z)* is the same as the solution to the problem *max g(x,y,z),* and is *x* = 1, *y* =-*2,* and *z* = 5.
- However, the optimal value of *f*(x,*y,z)* is the negative of the optimal value of *g(x,y,z).*

Unconstrained Optimization (Example)

$$\min_x f(x), \text{ where } f(x) = x^4 + 5x^3 + 4x^2 - 4x + 1. \qquad (2.4)$$



$$f'(x) = 4x^3 + 15x^2 + 8x - 4,$$

$f'(x) = 0$ at $x = -2.96$, $x = -1.10$, and $x = 0.31$.

$$f''(x) = 12x^2 + 30x + 8 = \begin{cases} 24.33, & x = -2.96 \\ -10.48, & x = -1.10 \\ 18.45, & x = 0.31 \end{cases}$$

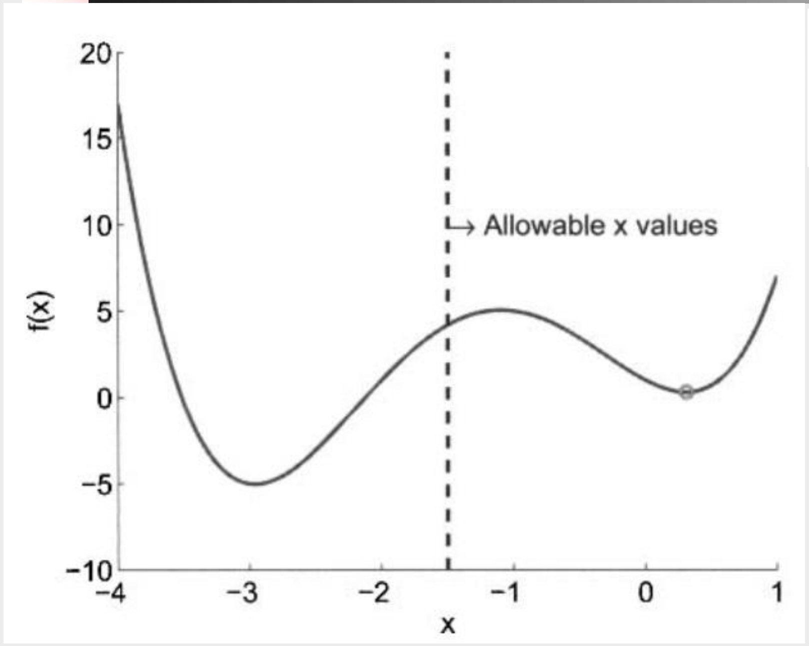
Figure 2.1 Example 2.2: A simple minimization problem. $f(x)$ has two local minima and one global minimum, which occurs at $x = -2.96$.

Recall that the second derivative of a function at a local minimum is positive, and the second derivative of a function at a local maximum is negative. The values of $f''(x)$ at the stationary points therefore confirm that $x = -2.96$ is a local minimum, $x = -1.10$ is a local maximum, and $x = 0.31$ is another local minimum.

A local minimum x^* can be defined as $f(x^*) < f(x)$ for all x such that $||x - x^*|| < \epsilon$

A global minimum x^* can be defined as $f(x^*) \leq f(x)$ for all x.

Constrained Optimization



$$\min_x f(x) \quad \text{where} \quad f(x) = x^4 + 5x^3 + 4x^2 - 4x + 1$$
$$\text{and} \quad x \geq -1.5.$$

$$f(x) = \begin{cases} 4.19 & \text{for } x = -1.50 \\ 0.30 & \text{for } x = 0.31 \end{cases}.$$

$f'(x) = 0$ at $x = -2.96$, $x = -1.10$, and $x = 0.31$.

Figure 2.2 Example 2.3: A simple constrained minimization problem. The constrained minimum occurs at $x = 0.31$.

Multi-Objective Optimization

$$\min_x [f(x) \text{ and } g(x)], \quad \text{where} \quad f(x) = x^4 + 5x^3 + 4x^2 - 4x + 1$$
$$\text{and} \quad g(x) = 2(x+1)^2.$$

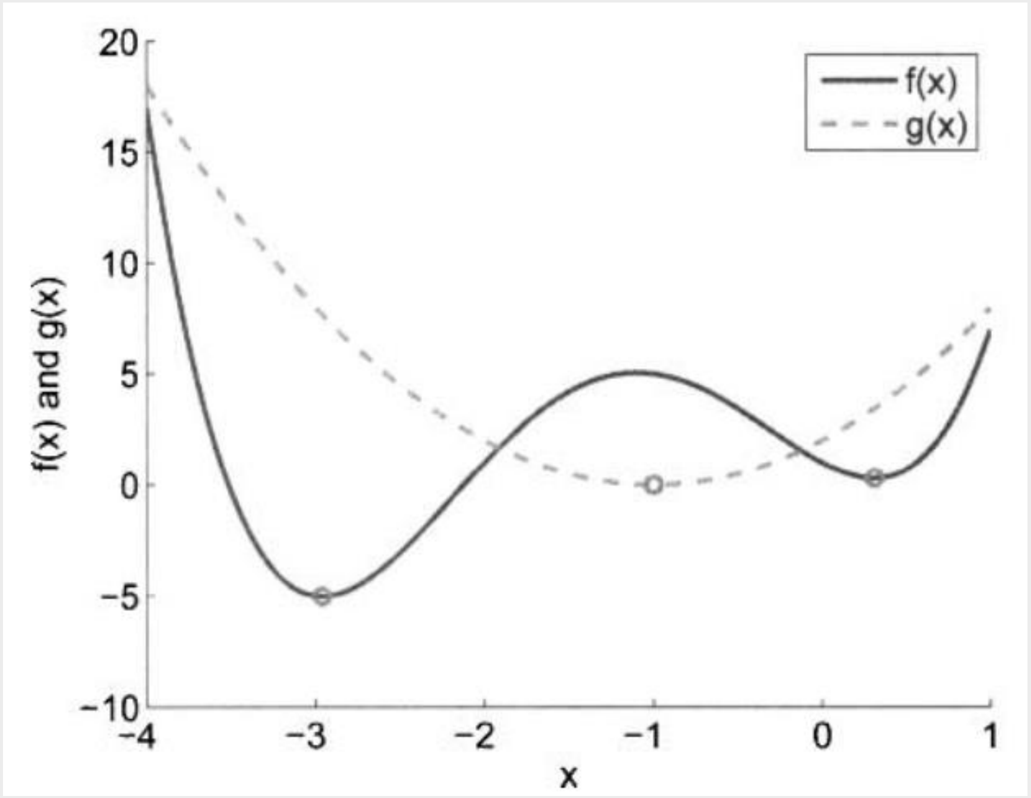
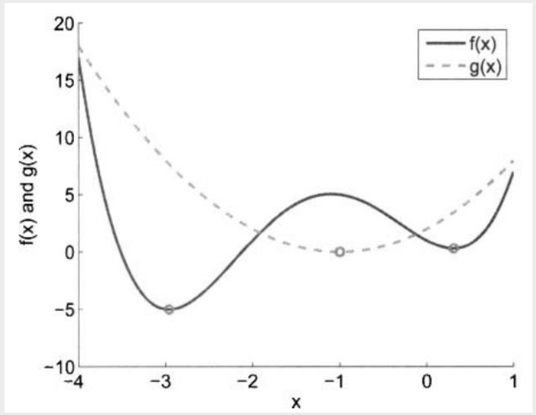


Figure 2.3 Example 2.4: A simple multi-objective minimization problem. $f(x)$ has two minima and $g(x)$ has one minimum. The two objectives conflict.

Multi-Objective Optimization (Cont.)

$$\min_x [f(x) \text{ and } g(x)], \quad \text{where} \quad f(x) = x^4 + 5x^3 + 4x^2 - 4x + 1$$
$$\text{and} \quad g(x) = 2(x+1)^2.$$



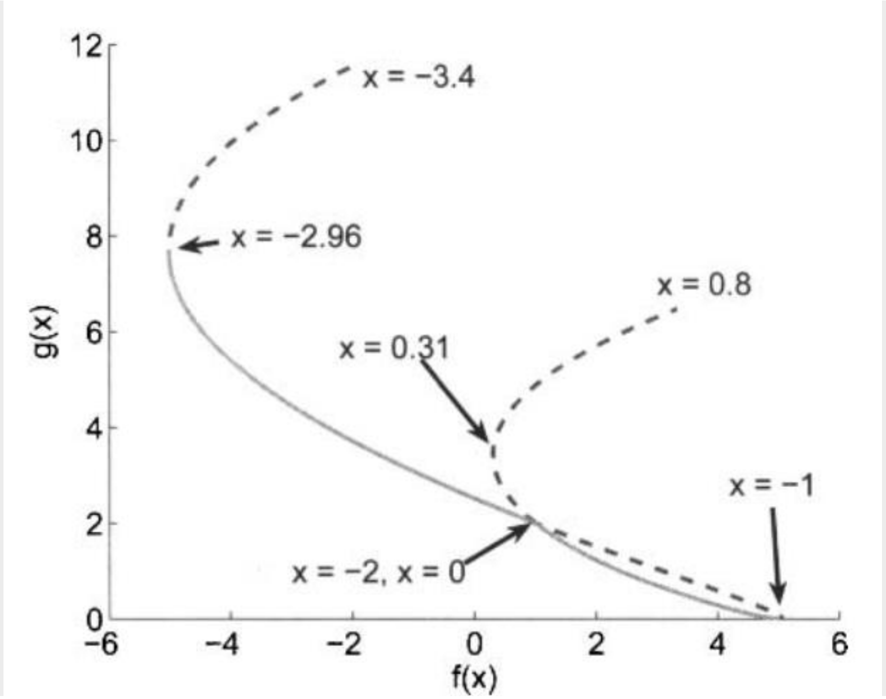


Figure 2.4 Example 2.4: This figure shows $g(x)$ as a function of $f(x)$ for a simple multi-objective minimization problem as x varies from -3.4 to 0.8. The solid line is the Pareto front.

The Pareto front gives a set of reasonable choices, but any choice of *x* from the Pareto set still entails a tradeoff between the two objectives.

- A typical real-world optimization problem involves many more objectives, and Pareto front is difficult to obtain.
- Even if we could obtain the Pareto front, we would not be able to visualize it because of its high dimensionality.

Multimodal Optimization

A multimodal optimization problem is a problem that has more than one local minimum.

$$\min_{x,y} f(x,y), \text{ where}$$

$$f(x,y) = e - 20\exp\left(-0.2\sqrt{\frac{x^2+y^2}{2}}\right) - \exp\left(\frac{\cos(2\pi x)+\cos(2\pi y)}{2}\right)$$

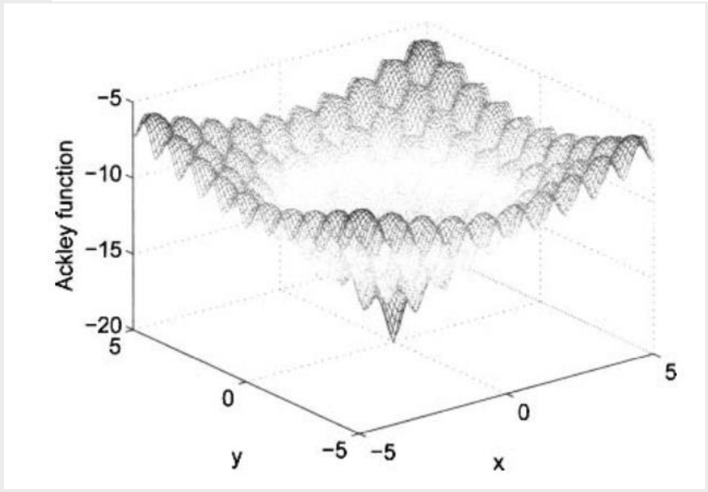
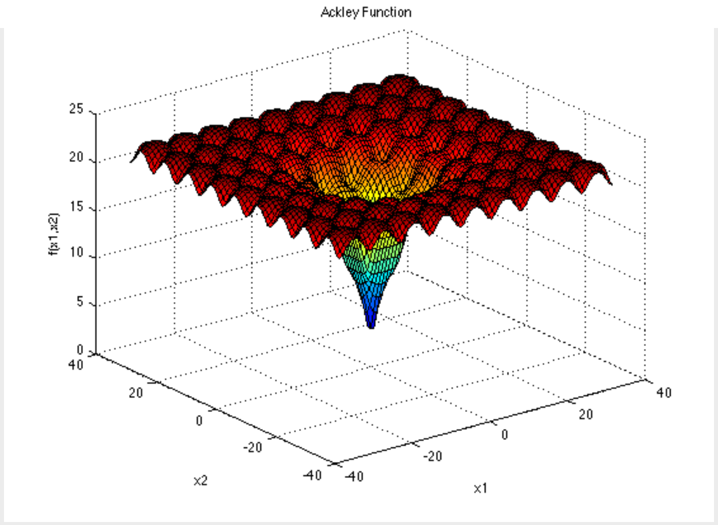


Figure 2.5 Example 2.5: The two-dimension Ackley function.



https://www.sfu.ca/~ssurjano/ackley.html

- We were able to solve shown previous examples using graphical methods or calculus
- But many real-world problems are more like this with more independent variables, with multiple objectives, and with constraints.
- Evolutionary Algorithms (EAs) are a good choice for real-world problems.

Combinatorial Optimization : Traveling Salesman Problem (TSP)

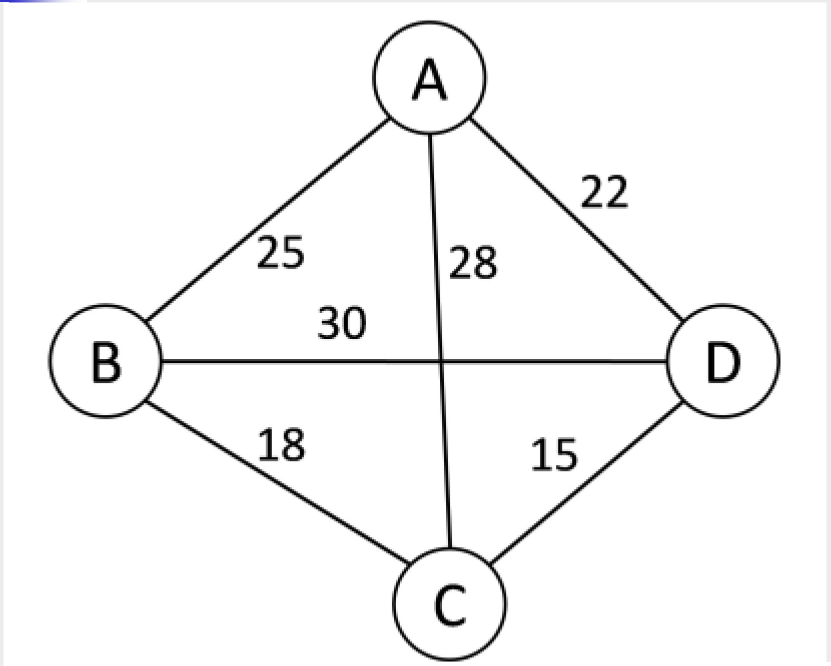


Figure 2.7: Simple Traveling Salesman Problem

There are four cities, A, B, C and D. Thus, the different possible routes would be (A as the hometown):
[A - B - C - D - A], [A - B - D - C - A],
[A - C - B - D - A], [A - C - D - B - A],
[A - D - C - B - A], [A - D - B - C - A]

If the salesman has to visit *n* cities there would be (*n*-1)! possibilities

- This number grows very rapidly, and for modest values of *n* it is not possible to calculate all possible solutions.
- Suppose the business person needs to visit one city in each of the 50 states in the USA. The number of possible solutions is 49! = 6.1 x 10^{62}.

Hill Climbing

Hill climbing (HC) is a simple optimization algorithm; actually, it is a family of algorithms with many variations.

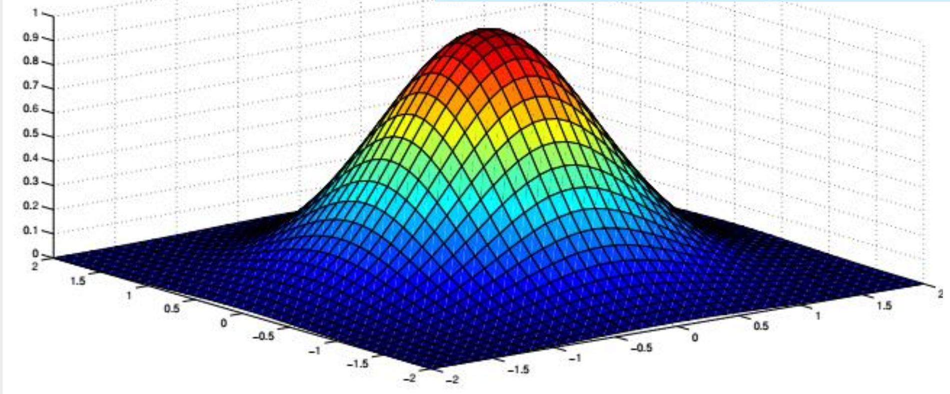


Figure 2.8: Simple Hill Climbing Problem

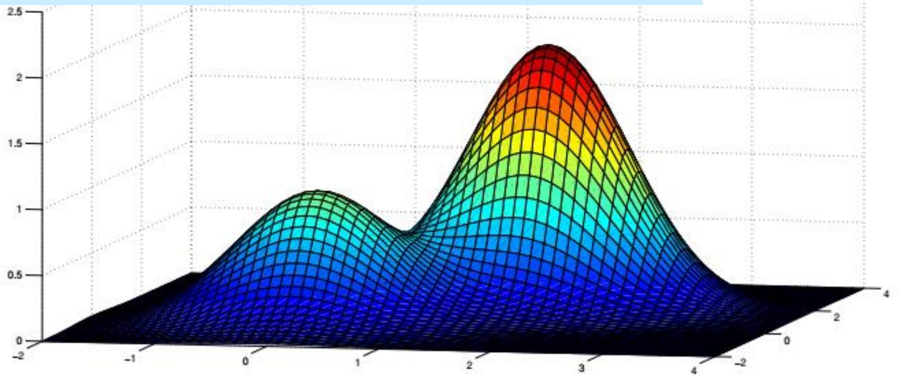


Figure 2.9: Multiple Hills in an Hill Climbing Problem

Figure 2.8 shows a simple hill-climbing surface and is the realization of the equation $e^{-(x2+y2)}$, where the value of x and y vary from -2 to 2.

Fig 2.9 is given by $e^{-(x^2+y^2)} + 2e^{-((x-1.7)^2+(y-1.7)^2)}$ in the range of -2 to 4.

- Some researchers consider HC to be a simple EA, while others consider it a non-evolutionary algorithm.
- HC is often a good first choice for solving a new optimization problem because it is simple, surprisingly effective
- The idea of HC is so straightforward that it must have been invented many times, and long ago, so it is difficult to determine its origin.

Hill Climbing

The different Hill Climbing solution techniques are:

1. **Simple Hill Climbing**: In a simple hill climbing algorithm, neighboring nodes are examined one after another. The first node, which gives a better objective function value than the existing position, is selected.

2. **Steepest Ascent Hill Climbing**: In the steepest ascent hill climbing technique all the neighboring positions are evaluated. The position which gives the largest improvement in the objective function value is selected as the next node.

3. **Iterative Hill Climbing**: In this technique, the process starts with a simple random guess. Increments are done for one input variable of the random solution.

4. **Stochastic Hill Climbing**: This method chooses a neighboring position randomly and evaluates its objective function value.

5. **Random Restart/Shotgun Hill Climbing**: In the random restart hill climbing method, the hill climbing is started with a random initial position.

Intelligence

- Computers were very good at some things that humans did poorly, like calculating the trajectory of a ballistic missile.
- But computers were (and still are) not effective at tasks that humans can do well, like recognizing a face.
- This led to attempts to mimic biological behavior in an effort to make computers better at such tasks.
- These efforts resulted in technologies like fuzzy systems, neural networks, genetic algorithms, and other EAs. EAs are therefore considered to be a part of the general category of computer intelligence.
- EAs should be *intelligent.* But what does it mean to be intelligent? Does it mean that our EAs can score high on an IQ test?
- Characteristics of Intelligence: adaptation, randomness, communication, feedback, exploration, and exploitation.

Intelligence Characteristics

- **Adaptation:**

  Adaptation to changing environments is a feature of intelligence.

- **Randomness:**
  - ✓ Some degree of randomness is a necessary component of intelligence.
  - ✓ Too much randomness will be counterproductive.
  - ✓ So randomness is a feature of intelligence, but only within limitations.

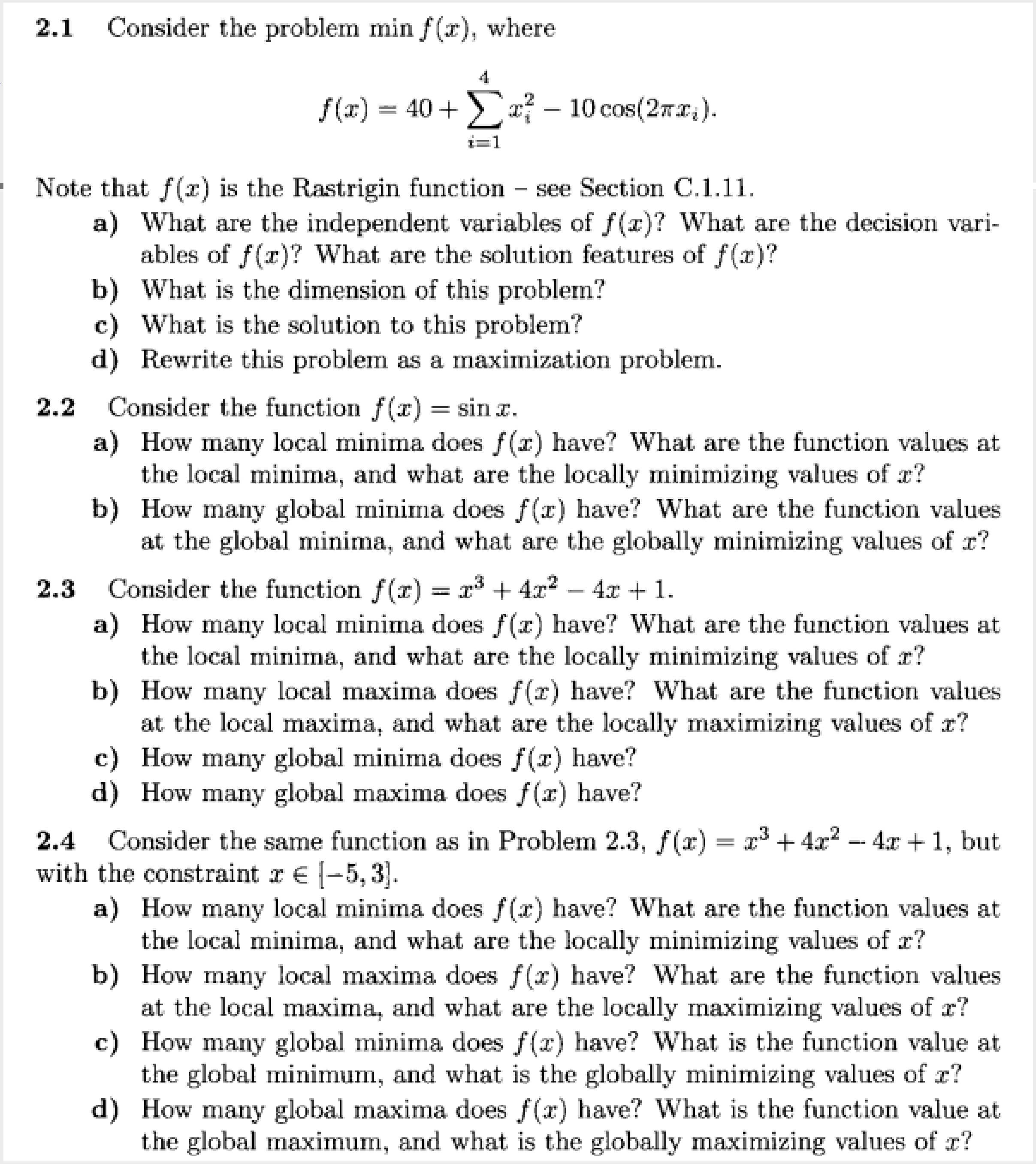
- **Communication:**
  - ✓ Communication is a feature of intelligence.
  - ✓ Intelligence not only involves communication, but it is also emergent. A single individual cannot be intelligent.
  - ✓ Communication is required to develop intelligence, and intelligence is required to communicate.
  - ✓ Intelligence and communication form a positive feedback loop.

Intelligence Characteristics

- **Feedback:**
  - ✓ Feedback is a fundamental characteristic of intelligence.
  - ✓ When we make mistakes, we change so that we don't repeat those mistakes. Feedback is also the basis for many natural phenomena.
  - ✓ Designed EA will have incorporate positive and negative feedback. An EA without feedback will not be very effective, but an EA with feedback has satisfied this necessary condition for intelligence.

- **Exploration and Exploitation:**
  - ✓ Exploration is the search for new ideas or new strategies.
  - ✓ Exploitation is the use of existing ideas and strategies that have proven successful in the past. Exploration is high-risk; a lot of new ideas waste time and lead to dead ends. However, exploration can also be high-return; a lot of new ideas pay off in ways that we could not have imagined.
  - ✓ Intelligence includes the proper balance of exploration and exploitation.

Exercises



2.1 Consider the problem $\min f(x)$, where

$$f(x) = 40 + \sum_{i=1}^{4} x_i^2 - 10\cos(2\pi x_i).$$

Note that $f(x)$ is the Rastrigin function – see Section C.1.11.

a) What are the independent variables of $f(x)$? What are the decision variables of $f(x)$? What are the solution features of $f(x)$?

b) What is the dimension of this problem?

c) What is the solution to this problem?

d) Rewrite this problem as a maximization problem.

2.2 Consider the function $f(x) = \sin x$.

a) How many local minima does $f(x)$ have? What are the function values at the local minima, and what are the locally minimizing values of x?

b) How many global minima does $f(x)$ have? What are the function values at the global minima, and what are the globally minimizing values of x?

2.3 Consider the function $f(x) = x^3 + 4x^2 - 4x + 1$.

a) How many local minima does $f(x)$ have? What are the function values at the local minima, and what are the locally minimizing values of x?

b) How many local maxima does $f(x)$ have? What are the function values at the local maxima, and what are the locally maximizing values of x?

c) How many global minima does $f(x)$ have?

d) How many global maxima does $f(x)$ have?

2.4 Consider the same function as in Problem 2.3, $f(x) = x^3 + 4x^2 - 4x + 1$, but with the constraint $x \in [-5, 3]$.

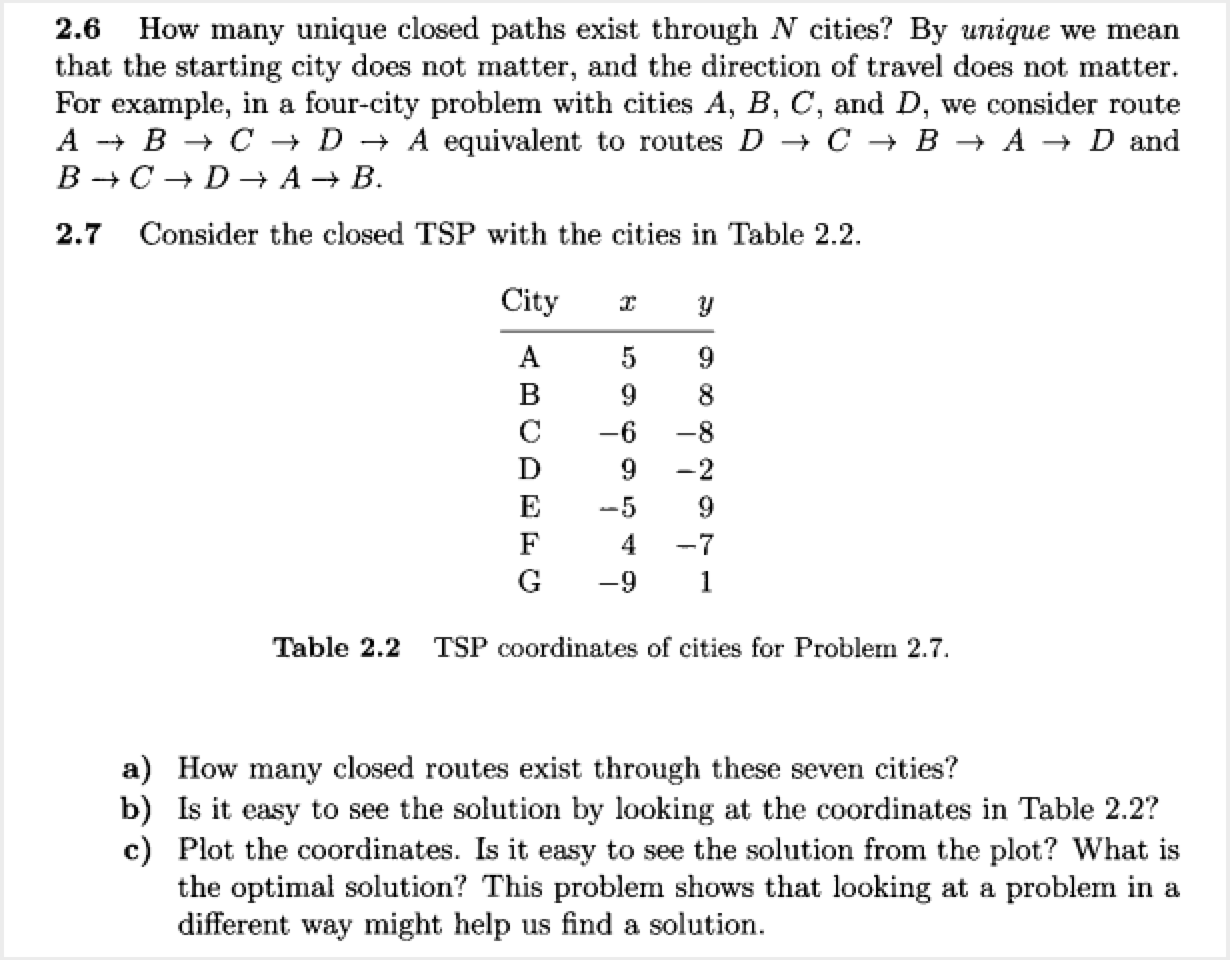
a) How many local minima does $f(x)$ have? What are the function values at the local minima, and what are the locally minimizing values of x?

b) How many local maxima does $f(x)$ have? What are the function values at the local maxima, and what are the locally maximizing values of x?

c) How many global minima does $f(x)$ have? What is the function value at the global minimum, and what is the globally minimizing values of x?

d) How many global maxima does $f(x)$ have? What is the function value at the global maximum, and what is the globally maximizing values of x?

Exercises



2.6 How many unique closed paths exist through N cities? By *unique* we mean that the starting city does not matter, and the direction of travel does not matter. For example, in a four-city problem with cities A, B, C, and D, we consider route $A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$ equivalent to routes $D \rightarrow C \rightarrow B \rightarrow A \rightarrow D$ and $B \rightarrow C \rightarrow D \rightarrow A \rightarrow B$.

2.7 Consider the closed TSP with the cities in Table 2.2.

City	x	y
A	5	9
B	9	8
C	−6	−8
D	9	−2
E	−5	9
F	4	−7
G	−9	1

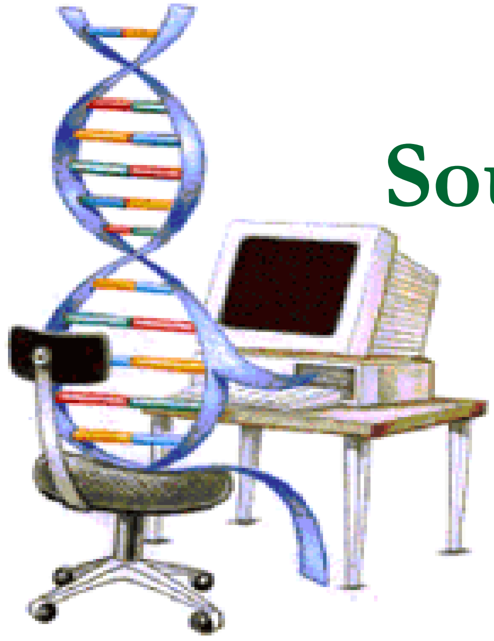
Table 2.2 TSP coordinates of cities for Problem 2.7.

a) How many closed routes exist through these seven cities?
b) Is it easy to see the solution by looking at the coordinates in Table 2.2?
c) Plot the coordinates. Is it easy to see the solution from the plot? What is the optimal solution? This problem shows that looking at a problem in a different way might help us find a solution.

Open Discussion

Genetic Algorithms

Prof. Dr. Md. Aminul Haque Akhand
Dept. of CSE, KUET



Source

1. Presentation of Dr. Chhavi Kashyap
2. Presentation of Dr. Son Kuswadi
3. Others

Overview

- Introduction To Genetic Algorithms (GAs)
- GA Operators and Parameters
- Application of Genetic Algorithms
- Summary

References

- D. E. Goldberg, ‘Genetic Algorithm In Search, Optimization And Machine Learning’, New York: Addison – Wesley (1989)
- John H. Holland ‘Genetic Algorithms’, Scientific American Journal, July 1992.
- Kalyanmoy Deb, ‘An Introduction To Genetic Algorithms’, Sadhana, Vol. 24 Parts 4 And 5.
- D. Whitley, et al, ‘Traveling Salesman And Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination’, Handbook Of Genetic Algorithms, New York
- Jason Brownlee, “Clever Algorithms - Nature-Inspired Programming Recipes” 2011.

WEBSITES

www.iitk.ac.in/kangal www.genetic-programming.com

Introduction To Genetic Algorithms (GAs)

History Of Genetic Algorithms

- "Evolutionary Computing" was introduced in the 1960s by **I. Rechenberg**.

- John Holland wrote the first book on Genetic Algorithms 'Adaptation in Natural and Artificial Systems' in 1975.

- In 1992 **John Koza** used genetic algorithm to evolve programs to perform certain tasks. He called his method "Genetic Programming".

"It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change."

-------- Charles Darwin

(12/02/1809 – 19/04/1882)

Basic Idea Of Principle Of Natural Selection

"Select The Best, Discard The Rest"

Darwin's Principle Of Natural Selection

- IF there are organisms that reproduce, and
- IF offsprings inherit traits from their progenitors, and
- IF there is variability of traits, and
- IF the environment cannot support all members of a growing population,
- THEN those members of the population with less-adaptive traits (determined by the environment) will die out, and
- THEN those members with more-adaptive traits (determined by the environment) will thrive

The result is the evolution of species.

An Example of Natural Selection

- **Giraffes have long necks.**

Giraffes with slightly longer necks could feed on leaves of higher branches when all lower ones had been eaten off.

→ They had a better chance of survival.

→ Favorable characteristic propagated through generations of giraffes.

→ Now, evolved species has long necks.

NOTE: Longer necks may have been a deviant characteristic (mutation) initially but since it was favorable, was propagated over generations. Now an established trait.

So, some mutations are beneficial.

Evolution Through Natural Selection

$$\begin{tikzcd}
\text{Initial Population Of Animals} \arrow[d] \\
\text{Struggle For Existence-Survival Of the Fittest} \arrow[d] \\
\text{Surviving Individuals Reproduce, Propagate Favorable Characteristics} \arrow[d] \arrow[loop left, "\text{Millions Of Years}"] \\
\text{Evolved Species}
\end{tikzcd}$$

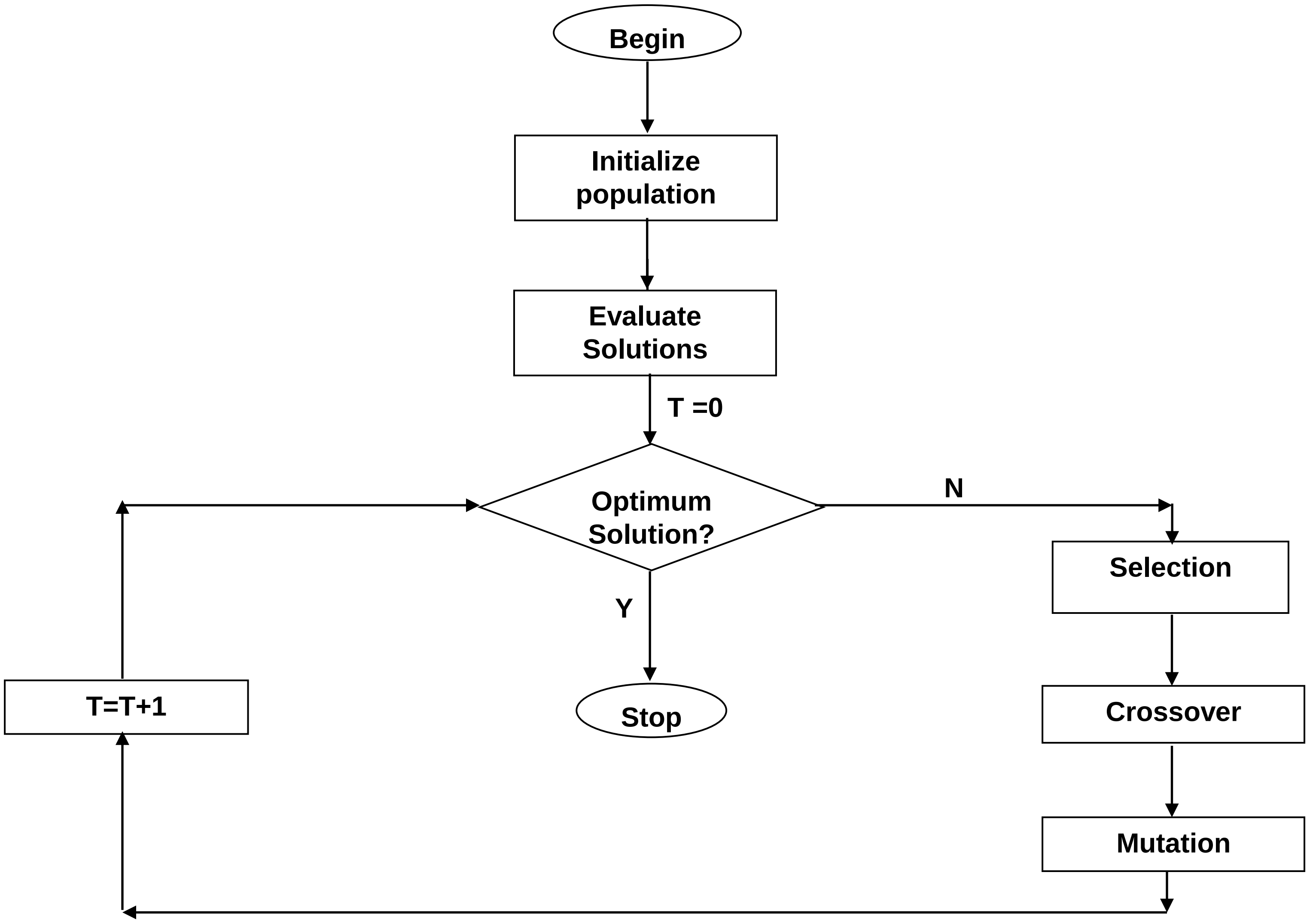
(Favorable Characteristic Now A Trait Of Species)

What Are Genetic Algorithms (GAs)?

Genetic Algorithms are *search* and *optimization* techniques based on Darwin’s Principle of *Natural Selection*.

Genetic Algorithms Implement Optimization Strategies By Simulating Evolution Of Species Through Natural Selection.

Working Mechanism Of GAs



Simple Genetic Algorithm

```
Simple_Genetic_Algorithm()
     {
     Initialize the Population;
     Calculate Fitness Function;

     While(Fitness Value != Optimal Value)
          {
          Selection;//Natural Selection, Survival
  Of Fittest
          Crossover;//Reproduction, Propagate
  favorable characteristics

          Mutation;//Mutation
          Calculate Fitness Function;
          }
     }
```

Nature to Computer Mapping

Nature	Computer
Population	Set of solutions.
Individual	Solution to a problem.
Fitness	Quality of a solution.
Chromosome	Encoding for a Solution.
Gene	Part of the encoding of a solution.
Reproduction	Crossover

GA Operators and Parameters

Encoding

*The process of representing the solution in the form of a **string** that conveys the necessary information.*

- Just as in a chromosome, each gene controls a particular characteristic of the individual, similarly, each bit in the string represents a characteristic of the solution.

Encoding Methods

- **Binary Encoding –** Most common method of encoding. Chromosomes are strings of 1s and 0s and each position in the chromosome represents a particular characteristic of the problem.

Chromosome A	10110010110011100101
Chromosome B	11111110000000011111

Encoding Methods (contd.)

- **Permutation Encoding –** Useful in ordering problems such as the Traveling Salesman Problem (TSP). Example. In TSP, every chromosome is a string of numbers, each of which represents a city to be visited.

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Encoding Methods (contd.)

- **Value Encoding –** Used in problems where complicated values, such as real numbers, are used and where binary encoding would not suffice.

  Good for some problems, but *often necessary to develop some specific crossover and mutation techniques for these chromosomes.*

Chromosome A	1.235 5.323 0.454 2.321 2.454
Chromosome B	(left), (back), (left), (right), (forward)

Fitness Function

A fitness function quantifies the optimality of a solution (chromosome) so that that particular solution may be ranked against all the other solutions.

- A fitness value is assigned to each solution depending on how close it actually is to solving the problem.
- Ideal fitness function correlates closely to goal + quickly computable.
- Example. In TSP, f(x) is sum of distances between the cities in solution. The lesser the value, the fitter the solution is.

Recombination

The process that determines which solutions are to be preserved and allowed to reproduce and which ones deserve to die out.

- The primary objective of the recombination operator is to emphasize the good solutions and eliminate the bad solutions in a population, while keeping the population size constant.
- “Selects The Best, Discards The Rest”.
- “Recombination” is different from “Reproduction”.

Recombination

- Identify the good solutions in a population.
- Make multiple copies of the good solutions.
- Eliminate bad solutions from the population so that multiple copies of good solutions can be placed in the population.

Roulette Wheel Selection

- Each current string in the population has a slot assigned to it which is in proportion to it’s fitness.

- We spin the weighted *roulette wheel* thus defined *n* times (where *n* is the total number of solutions).

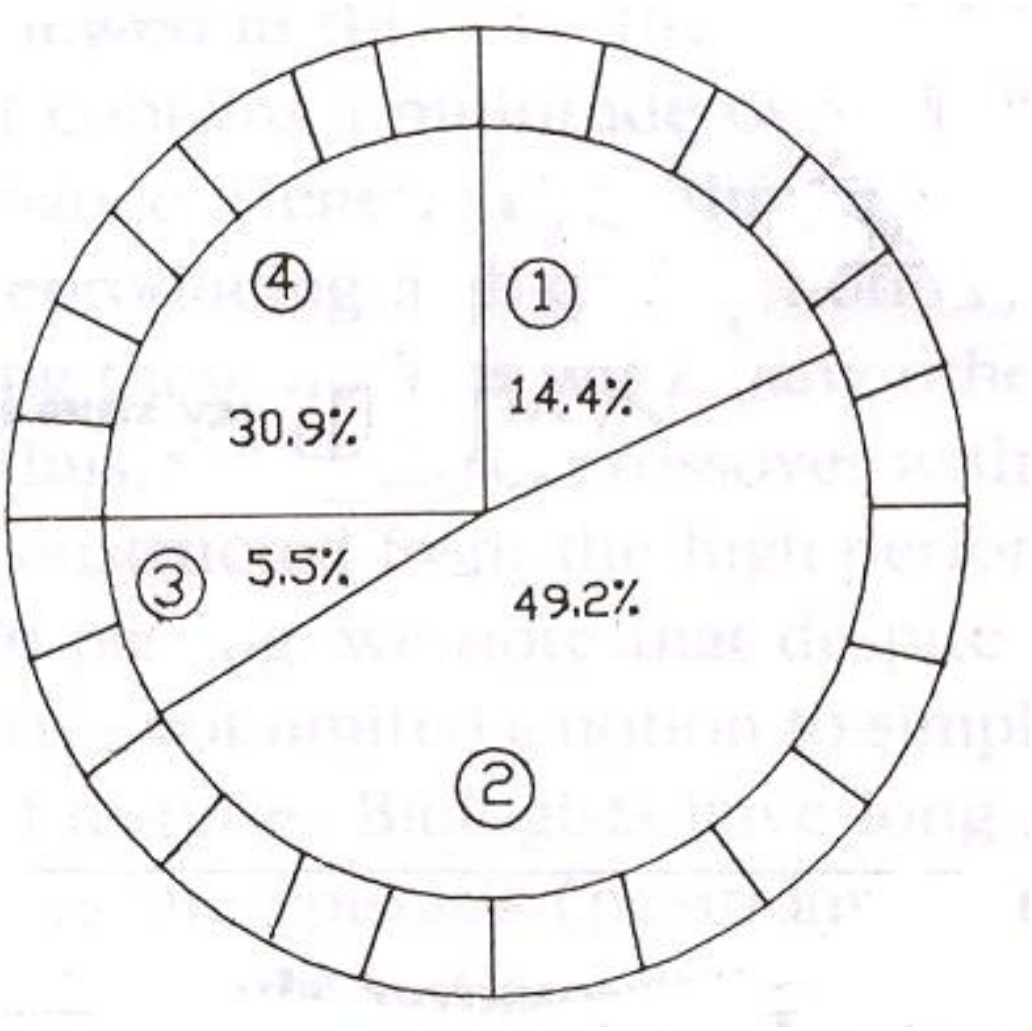
- Each time Roulette Wheel stops, the string corresponding to that slot is created.

Strings that are fitter are assigned a larger slot and hence have a better chance of appearing in the new population.

Example Of Roulette Wheel Selection

No.	String	Fitness	% Of Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

Roulette Wheel For Example



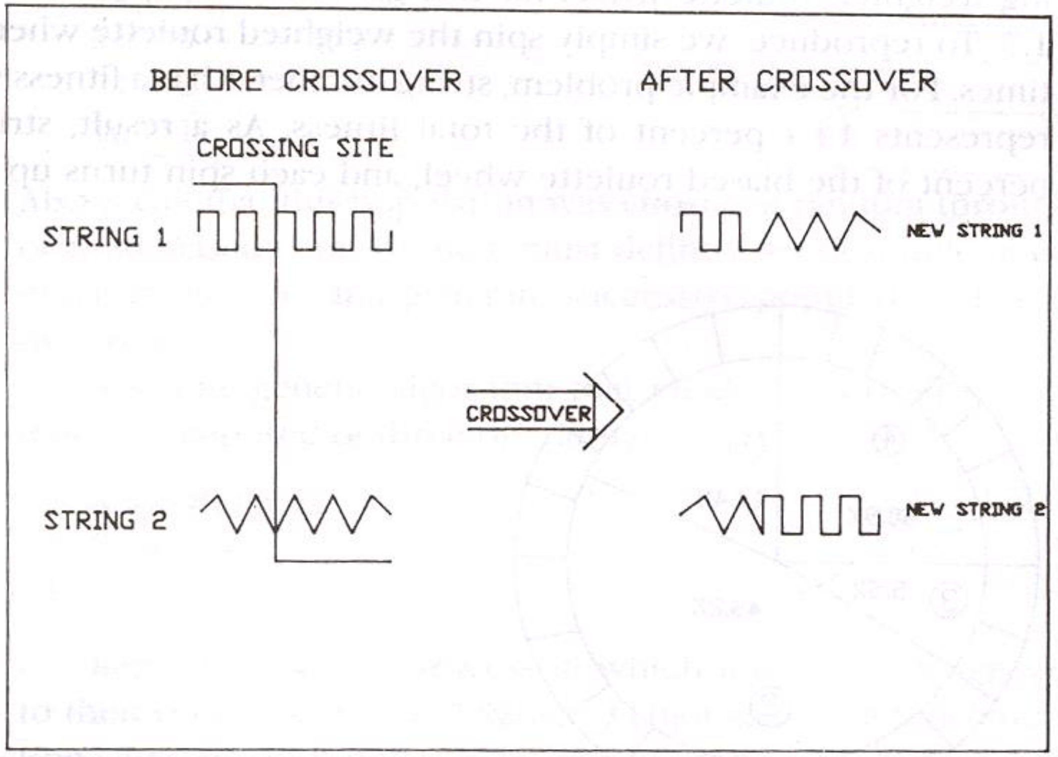
Crossover

It is the process in which two chromosomes (strings) combine their genetic material (bits) to produce a new offspring which possesses both their characteristics.

- *Two strings are picked from the mating pool at random to cross over.*
- *The method chosen depends on the Encoding Method.*

Crossover Methods

- **Single Point Crossover-** A random point is chosen on the individual chromosomes (strings) and the genetic material is exchanged at this point.



Crossover Methods (contd.)

- Single Point Crossover

Chromosome1	11011 | 00100110110
Chromosome 2	11011 | 11000011110
Offspring 1	11011 | 11000011110
Offspring 2	11011 | 00100110110

Crossover Methods (contd.)

- **Two-Point Crossover-** Two random points are chosen on the individual chromosomes (strings) and the genetic material is exchanged at these points.

Chromosome1	**11011 | 00100 | 110110**
Chromosome 2	**10101 | 11000 | 011110**
Offspring 1	**10101 | 00100 | 011110**
Offspring 2	**11011 | 11000 | 110110**

NOTE: These chromosomes are different from the last example.

Crossover Methods (contd.)

- **Uniform Crossover-** Each gene (bit) is selected randomly from one of the corresponding genes of the parent chromosomes.

Chromosome1	11011 | 00100 | 110110
Chromosome 2	10101 | 11000 | 011110
Offspring	10111 | 00000 | 110110

NOTE: Uniform Crossover yields ONLY 1 offspring.

Crossover (contd.)

- Crossover between 2 good solutions **MAY NOT ALWAYS** yield a better or as good a solution.
- Since parents are good, probability of the child being good is high.
- If offspring is not good (poor solution), it will be removed in the next iteration during “Selection”.

Elitism

Elitism is a method which copies the best chromosome to the new offspring population before crossover and mutation.

- When creating a new population by crossover or mutation the best chromosome might be lost.
- Forces GAs to retain some number of the best individuals at each generation.
- Has been found that elitism significantly improves performance.

Mutation

It is the process by which a string is deliberately changed so as to maintain diversity in the population set.

We saw in the giraffes' example, that mutations could be beneficial.

Mutation Probability- determines how often the parts of a chromosome will be mutated.

Example Of Mutation

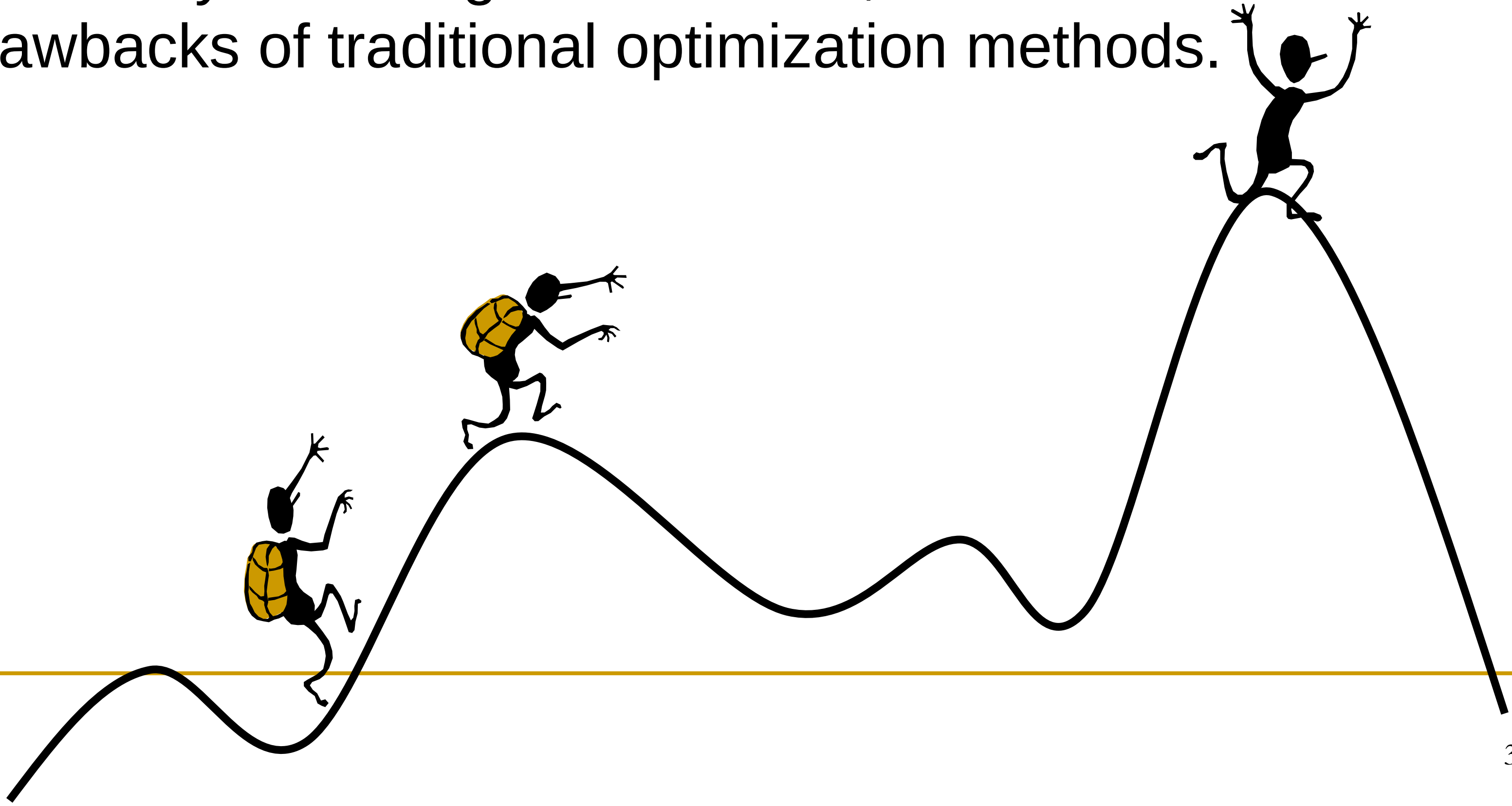
- For chromosomes using Binary Encoding, randomly selected bits are inverted.

Offspring	**11011 00100 110110**
Mutated Offspring	**11010 00100 100110**

NOTE: The number of bits to be inverted depends on the Mutation Probability.

Advantages Of GAs

- **Global Search Methods:** GAs search for the function optimum starting from a *population of points* of the function domain, not a single one. This characteristic suggests that GAs are global search methods. They can, in fact, climb many peaks in parallel, reducing the probability of finding local minima, which is one of the drawbacks of traditional optimization methods.



Advantages of GAs (contd.)

- **Blind Search Methods:** GAs only use the information about the *objective function*. They do not require knowledge of the first derivative or any other auxiliary information, allowing a number of problems to be solved without the need to formulate restrictive assumptions. For this reason, GAs are often called blind search methods.

Advantages of GAs (contd.)

- **GAs use probabilistic transition rules** during iterations, unlike the traditional methods that use fixed transition rules.

  This makes them more robust and applicable to a large range of problems.

Advantages of GAs (contd.)

- **GAs can be easily used in *parallel machines-*** Since in real-world design optimization problems, most computational time is spent in evaluating a solution, with multiple processors all solutions in a population can be evaluated in a distributed manner. This reduces the overall computational time substantially.

Simple Example (Goldberg98)

- **Simple problem: max x^2 over {0,1,…,31}**
- GA approach:
  - Representation: binary code, e.g. 01101 $\leftrightarrow$ 13
  - Population size: 4
  - 1-point xover, bitwise mutation
  - Roulette wheel selection
  - Random initialization
- We show one generational cycle done by hand

Selection

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

Crossover

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

Mutation

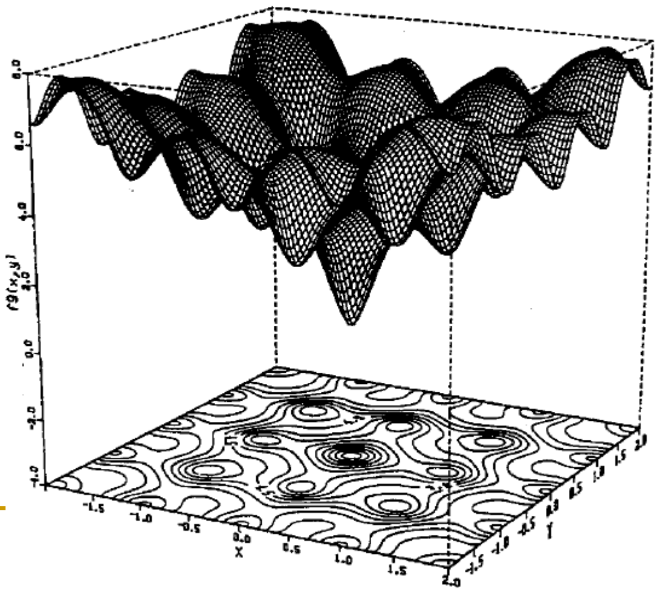
String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	$\boxed{1}$ 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 $\boxed{1}$ 0 0	18	324
Sum				2354
Average				588.5
Max				729

Hard Problems

- Solution of $f(x)=x^2$ is easy to find and not realistic for GA
- Many problems occur as real valued problems, e.g. continuous parameter optimization $f : \Re^n \rightarrow \Re$
- Illustration: Ackley's function (often used in EC)

$$f(\overline{x}) = -c_1 \cdot exp\left(-c_2 \cdot \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n} \cdot \sum_{i=1}^{n} cos(c_3 \cdot x_i)\right) + c_1 + 1$$

$$c_1 = 20,\ c_2 = 0.2,\ c_3 = 2\pi$$



Genetic Algorithms To Solve The Traveling Salesman Problem (TSP)

The Problem

The **Traveling Salesman Problem** is defined as:

'We are given a set of cities and a symmetric distance matrix that indicates the cost of travel from each city to every other city.

*The goal is to find **the shortest circular tour,** visiting every city exactly once, so as to **minimize the total travel cost**, which includes the cost of traveling from the last city back to the first city'.*

Encoding

- I represent every city with an integer .
- Consider 6 Indian cities –

  Mumbai, Nagpur , Calcutta, Delhi , Bangalore and Chennai and assign a number to each.

Mumbai	$\longrightarrow$	1
Nagpur	$\longrightarrow$	2
Calcutta	$\longrightarrow$	3
Delhi	$\longrightarrow$	4
Bangalore	$\longrightarrow$	5
Chennai	$\longrightarrow$	6

Encoding (contd.)

- Thus a path would be represented as a sequence of integers from 1 to 6.
- The path [1 2 3 4 5 6] represents a path from Mumbai to Nagpur, Nagpur to Calcutta, Calcutta to Delhi, Delhi to Bangalore, Bangalore to Chennai, and finally from Chennai to Mumbai.
- This is an example of **Permutation Encoding** as the position of the elements determines the fitness of the solution.

Fitness Function

- The fitness function will be the total cost of the tour represented by each chromosome.
- This can be calculated as the sum of the distances traversed in each travel segment.

The Lesser The Sum, The Fitter The Solution Represented By That Chromosome.

Distance/Cost Matrix For TSP

	1	2	3	4	5	6
1	0	863	1987	1407	998	1369
2	863	0	1124	1012	1049	1083
3	1987	1124	0	1461	1881	1676
4	1407	1012	1461	0	2061	2095
5	998	1049	1881	2061	0	331
6	1369	1083	1676	2095	331	0

Cost matrix for six city example.

Distances in Kilometers

Fitness Function (contd.)

- So, for a chromosome [4 1 3 2 5 6], the total cost of travel or fitness will be calculated as shown below

- Fitness = *1407* + *1987* + *1124* + *1049* + *331*+ *2095*

  = *7993 kms.*

- Since our objective is to Minimize the distance, the lesser the total distance, the fitter the solution.

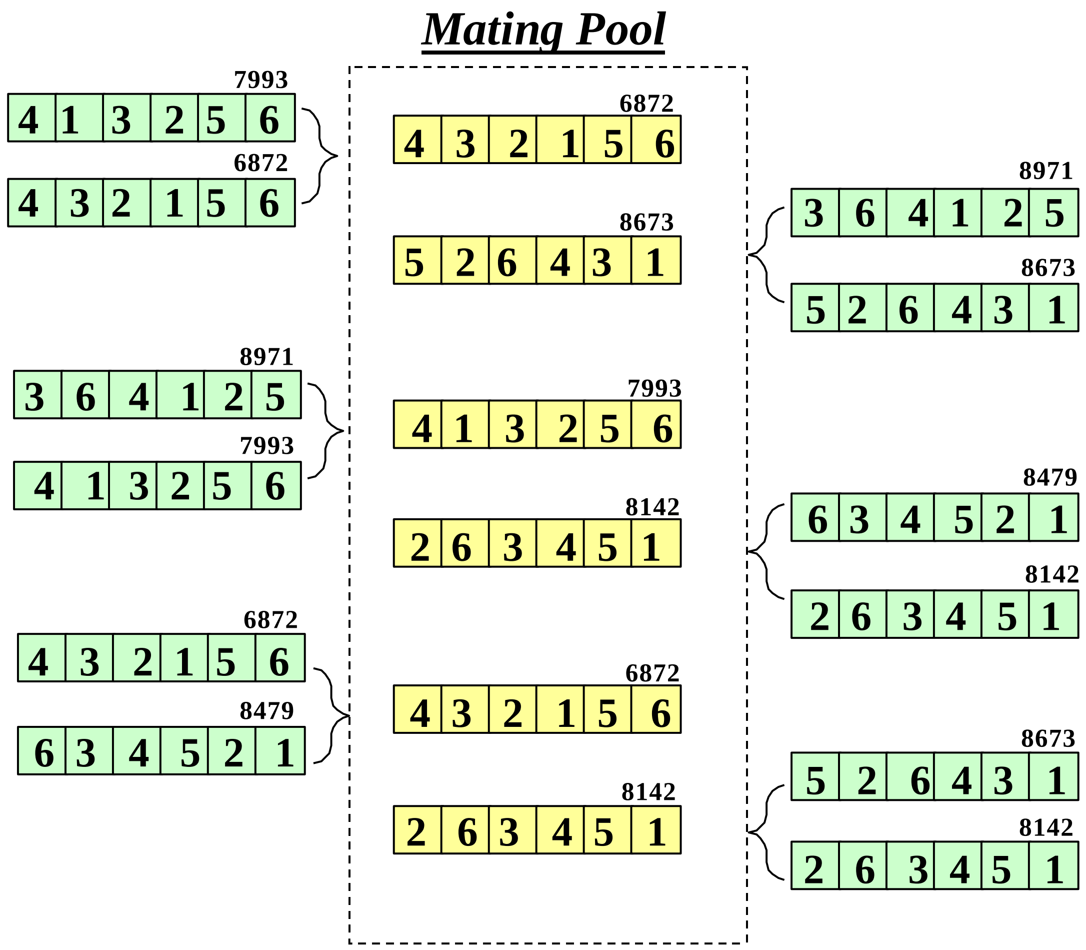
Selection Operator

May use ***Tournament Selection***.

As the name suggests *tournaments* are played between two solutions and the better solution is chosen and placed in the *mating pool.*

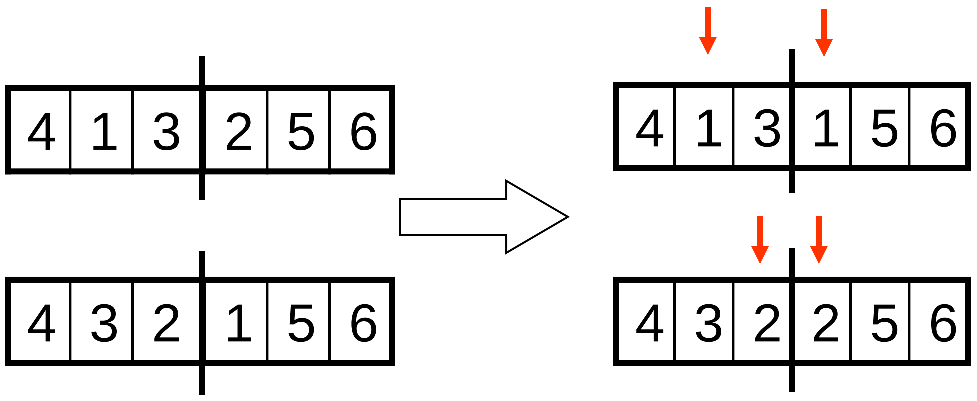
Two other solutions are picked again and another slot in the *mating pool* is filled up with the better solution.

Tournament Selection (contd.)



Why we cannot use single-point crossover:

- Single point crossover method randomly selects a crossover point in the string and swaps the substrings.
- This may produce some invalid offsprings as shown below.



Crossover Operator

- Used the *Enhanced Edge Recombination* operator (T.Starkweather, et al, *'A Comparison of Genetic Sequencing Operators,* International Conference of GAs, 1991).
- This operator is different from other genetic sequencing operators in that it emphasizes *adjacency information* instead of the order or position of items in the sequence.
- The algorithm for the Edge-Recombination Operator involves constructing an Edge Table first.

Edge Table

The *Edge Table* is an *adjacency table* that lists links *into* and *out of* a city found in the two parent sequences.

If an item is already in the edge table and we are trying to insert it again, that element of a sequence must be a *common edge* and is represented by inverting it's sign.

Finding The Edge Table

Parent 1	4	1	3	2	5	6
Parent 2	4	3	2	1	5	6

1	4	3	2	5
2	-3	5	1	
3	1	-2	4	
4	-6	1	3	
5	1	2	-6	
6	-5	-4		

Enhanced Edge Recombination Algorithm

1. Choose the initial city from one of the two parent tours. (It can be chosen randomly as according to criteria outlined in *step 4*). This is the *current city.*

2. Remove all occurrences of the *current city* from the left hand side of the edge table.(These can be found by referring to the edge-list for the *current city*).

3. If the *current city* has entries in it's edge-list, go to *step 4* otherwise go to *step 5*.

4. Determine which of the cities in the edge-list of the *current city* has the fewest entries in it's own edge-list. The city with fewest entries becomes the *current city*. In case a negative integer is present, it is given preference. Ties are broken randomly. Go to *step 2.*

5. If there are no remaining *unvisited* cities, then *stop.* Otherwise, randomly choose an *unvisited* city and go to *step 2.*

Example Of Enhanced Edge Recombination Operator

Step 1

1	4	3	2	5
2	-3	5	1	
3	1	-2	4	
4	-6	1	3	
5	1	2	-6	
6	-5	-4		

4					

Step 2

1	3	2	5
2	-3	5	1
3	1	-2	
4	-6	1	3
5	1	2	-6
6	-5		

4	6				

Example Of Enhanced Edge Recombination Operator (contd.)

Step 3

1	3	2	5
2	-3	5	1
3	1	-2	
4	1	3	
5	1	2	
6	-5		

4	6	5			

Step 4

1	3	2
2	-3	1
3	1	-2
4	1	3
5	1	2
6		

4	6	5	1		

Example Of Enhanced Edge Recombination Operator (contd.)

Step 5

1	3	2
2	-3	
3	-2	
4	3	
5	2	
6		

				↓	
4	6	5	1	3	

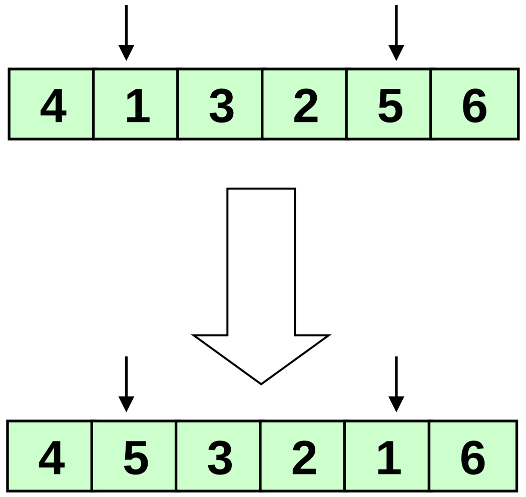
Step 6

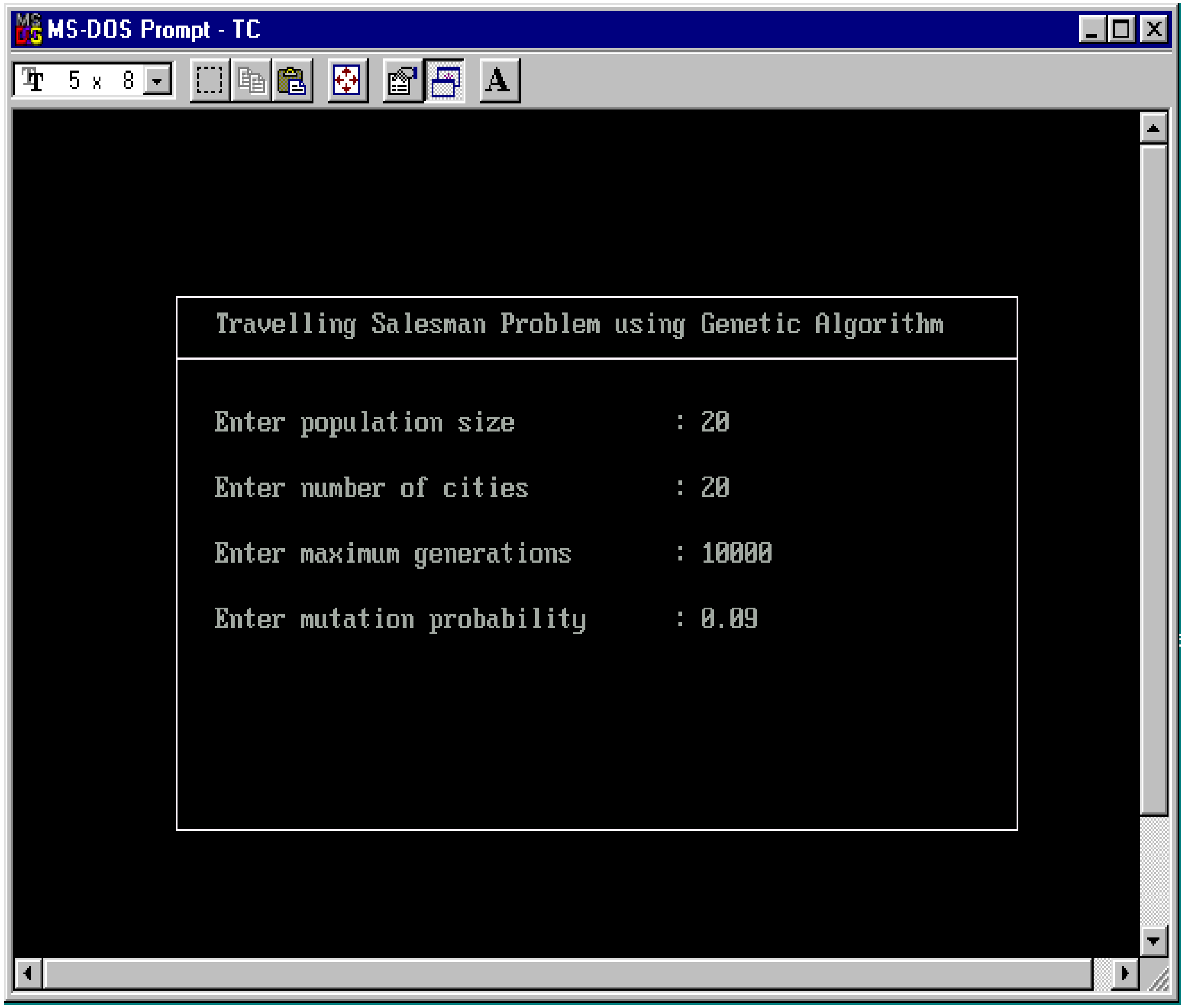
1	2
2	
3	-2
4	
5	2
6	

					↓
4	6	5	1	3	2

Mutation Operator

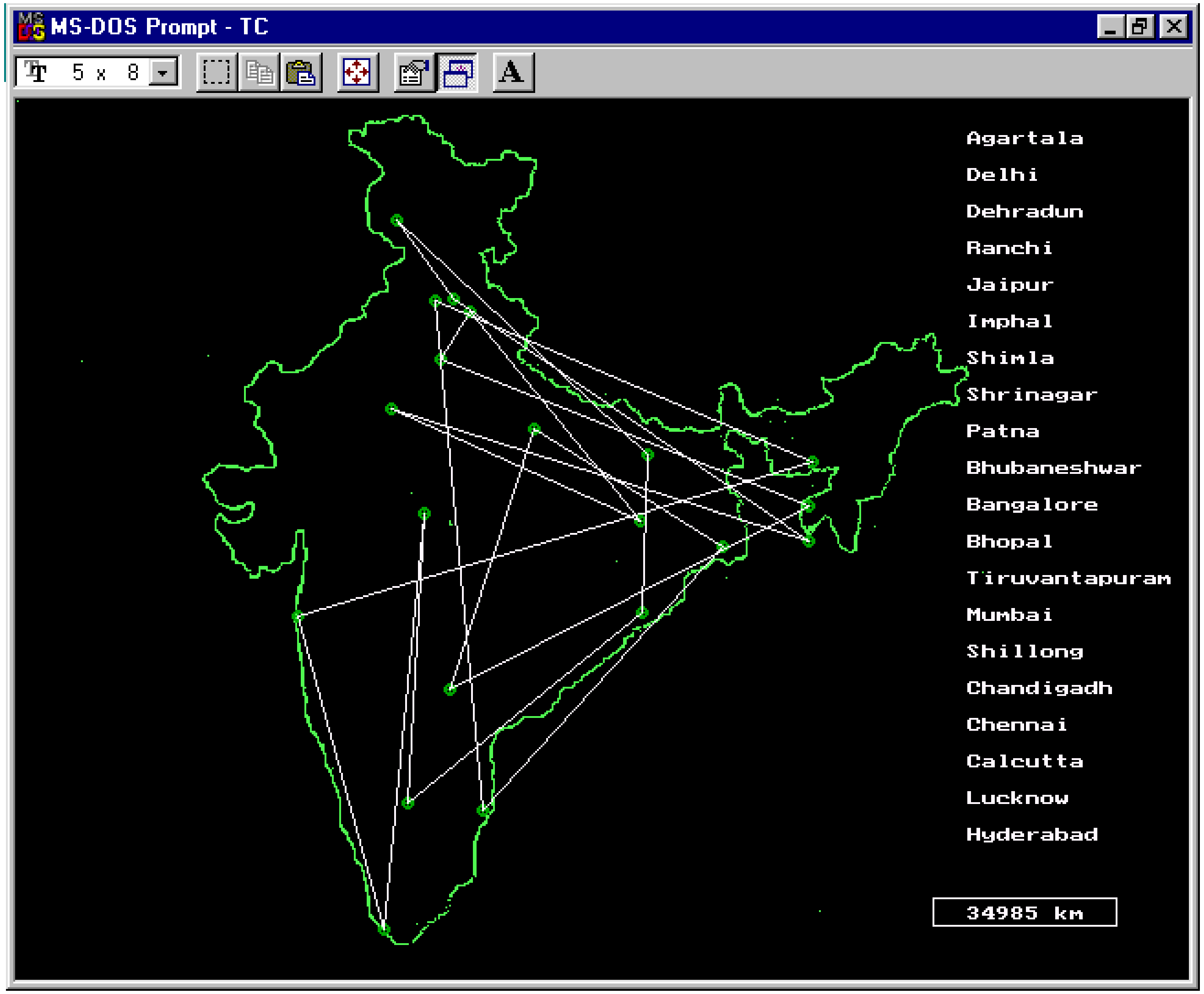
- The mutation operator induces a change in the solution, so as to maintain diversity in the population and prevent *Premature Convergence*.
- In our project, we mutate the string by randomly selecting any two cities and interchanging their positions in the solution, thus giving rise to a new tour.





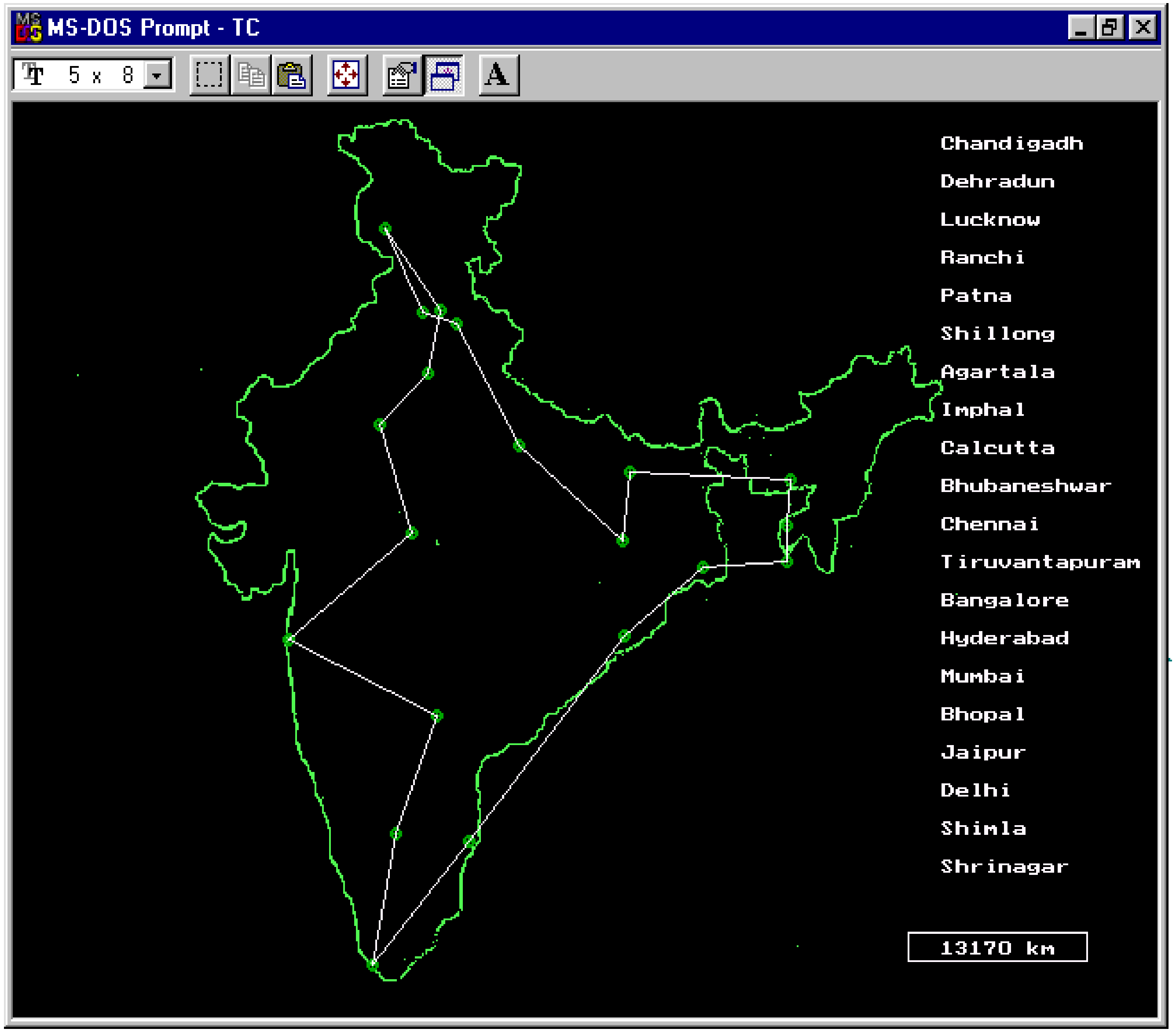
```
Travelling Salesman Problem using Genetic Algorithm

Enter population size            : 20

Enter number of cities           : 20

Enter maximum generations        : 10000

Enter mutation probability       : 0.09
```

Input To Program



```
Agartala
Delhi
Dehradun
Ranchi
Jaipur
Imphal
Shimla
Shrinagar
Patna
Bhubaneshwar
Bangalore
Bhopal
Tiruvantapuram
Mumbai
Shillong
Chandigadh
Chennai
Calcutta
Lucknow
Hyderabad

34985 km
```

Initial Output For 20 cities : Distance=34985 km
Initial Population



```
Chandigadh
Dehradun
Lucknow
Ranchi
Patna
Shillong
Agartala
Imphal
Calcutta
Bhubaneshwar
Chennai
Tiruvantapuram
Bangalore
Hyderabad
Mumbai
Bhopal
Jaipur
Delhi
Shimla
Shrinagar

13170 km
```

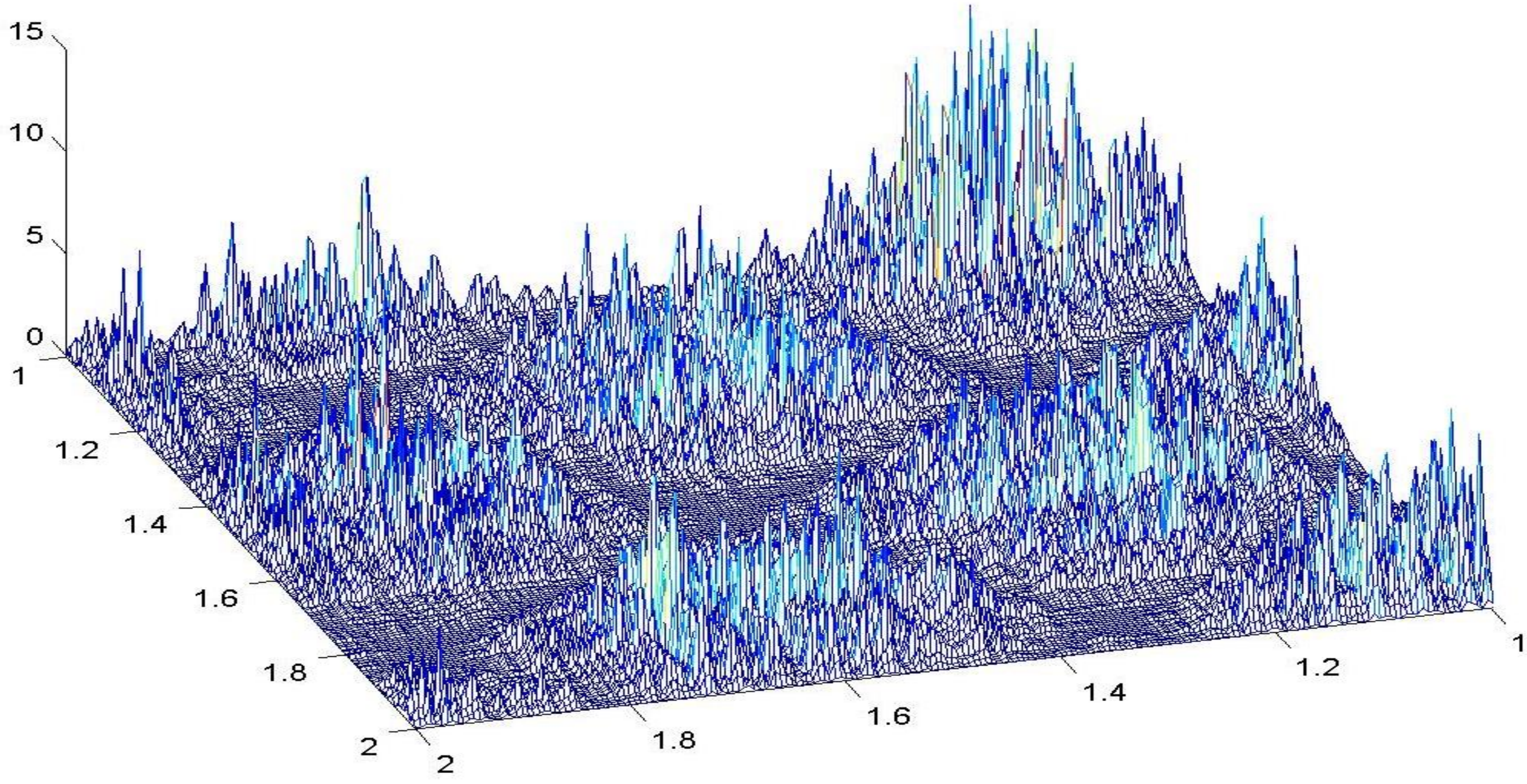
Final Output For 20 cities : Distance=13170 km
Generation 4786

Other Typical applications:

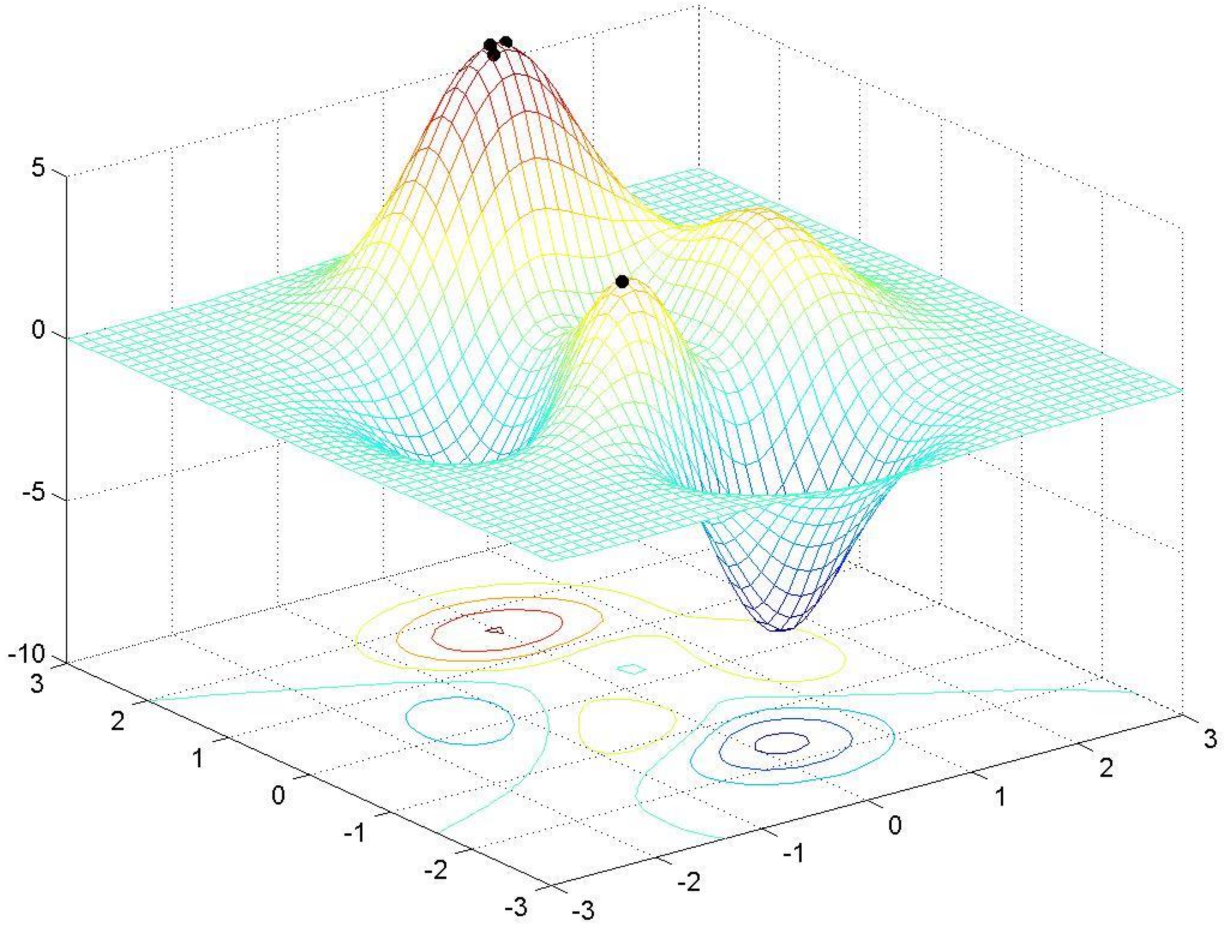
When to use it?

- machine learning
- timetabling, scheduling
- data mining
- robot trajectory
- etc.
- optimisation problems
- designing neural networks
- strategy planning
- evolving programs

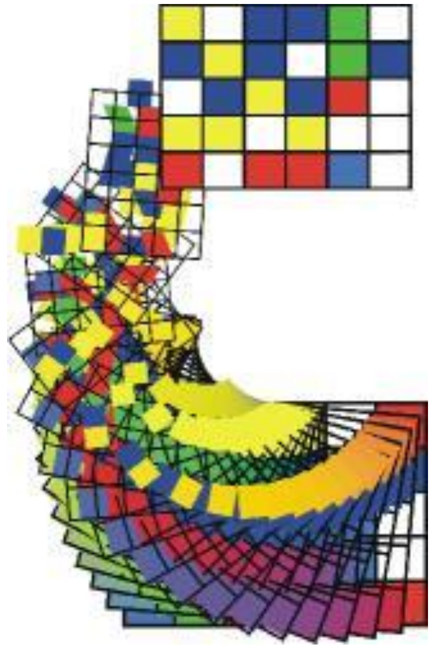
Extremes of multimodal functions



Extremes of multimodal functions



Timetabling



- Each chromosome represents a timetable
- A random population of feasible
- timetables is created
- Each timetable is evaluated according to chosen criteria
- Timetables are selected for reproduction
- Crossover and mutation operators are used to produce offspring
- The population increasingly consists of “good” timetables

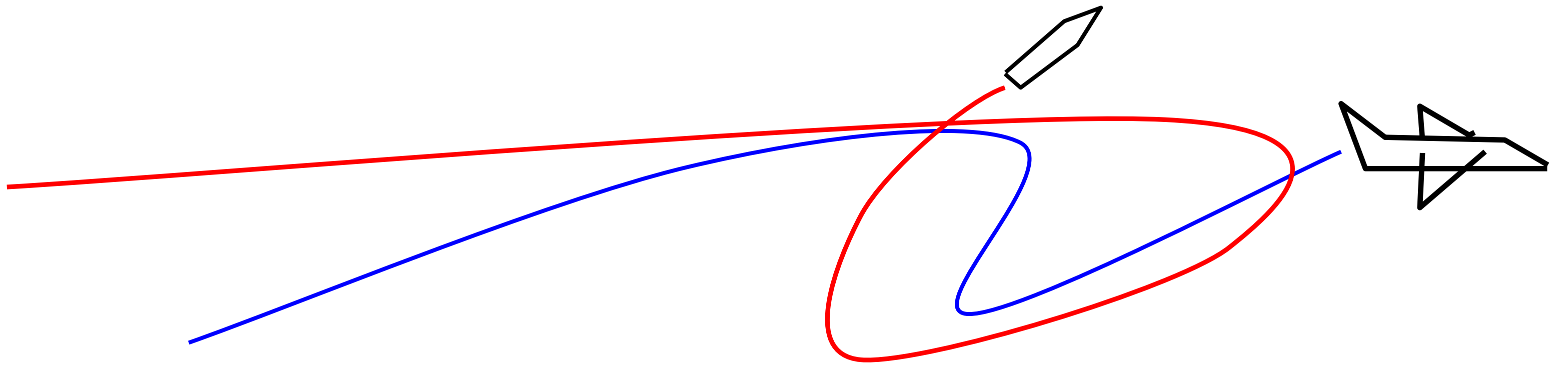
Evolution of Strategies - Game Playing

- Each chromosome represents different strategies for playing the game
- Initial population is generated randomly
- During the selection process each strategy is required to play a certain number of games against other strategies
- The strategies with the most wins are selected for reproduction
- The population increasingly consists of “good” strategies

Rule Discovery

- SAMUEL system discovers rules by which a slower but more manoeuvrable aircraft can evade a faster but less agile missile until the missile runs out of fuel
- The aircraft can sense the range, speed, heading and bearing of the missile
- Rules are of a fairly uniform sort, such as to turn left by 90 degrees if the missile parameters are lie within certain intervals

Rule Discovery

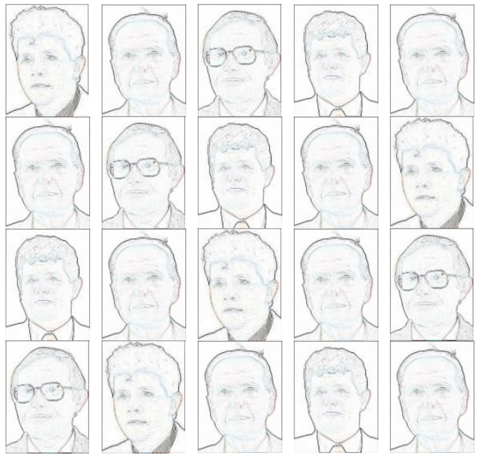


- Chromosomes are ordered sets of possible rules (if situation x occurs do action encoded within x-th gene)
- Fitness evaluation is by simulation (it is examined for how long the aircraft can evade the missile using particular set of rules)
- It takes hours to evaluate initial population !

Criminal suspect recognition

- It is easy to identify a criminal from photo but hard to describe his or her features
- It is difficult to generate even from computer library of visual features
- Idea to use genetic algorithms to help witnesses in the identification of criminal suspects

Criminal suspect recognition



Faceprints

- randomly generates 20 faces on a computer screen
- witness evaluates each face on a 10 point scale
- GA generates additional faces from 5 building blocks: eyes, mouth, nose, hair and chin (7-bit string each)
- Chromosome is a 35-bit binary string (34 billion faces)
- Witness rates successive generations with 10 point scale
- Convergence often occurs after 20 generations

Boeing 777

- The engines were designed by classical techniques with the emphasis on efficient fuel consumption
- Genetic algorithms were used to fine-tuning of some parameters of the already designed engines
- Due to their utilization at this stage the consumption of fuel has been reduced by 2,5%
- (operating cost savings cca 2 mil USD at one plane per year)

Summary

- *Genetic Algorithms* (GAs) implement optimization strategies based on simulation of the natural law of evolution of a species by *natural selection*

- The basic GA Operators are:
  - Encoding
  - Recombination
  - Crossover
  - Mutation

- GAs have been applied to a variety of function optimization problems, and have been shown to be highly effective in searching a large, poorly defined search space even in the presence of difficulties such as high-dimensionality, multi-modality, discontinuity and noise.

Questions ?