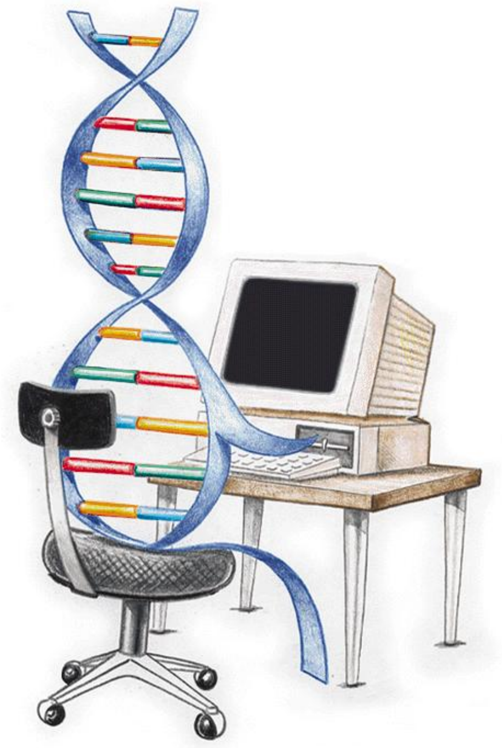
The Genetic Algorithm



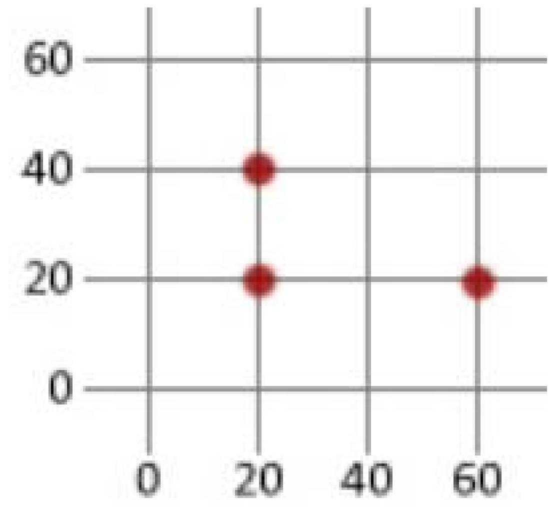
The Genetic Algorithm

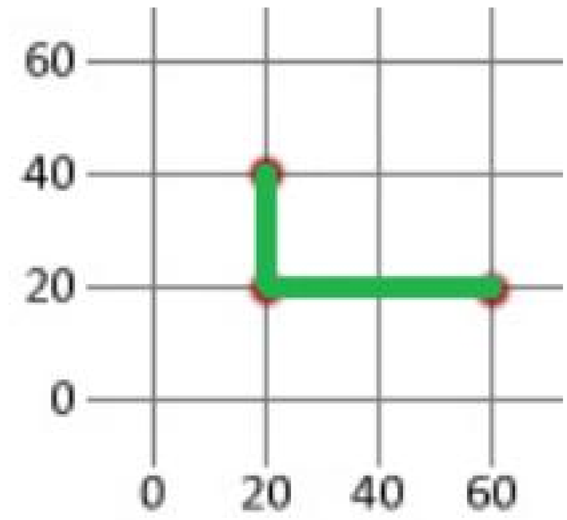
Computer programs that "evolve" in ways that resembles natural selection can solve complex problems even their creators do not fully understand.

John Holland

NEED FOR GENETIC ALGORITHM

- Travelling salesman problem
- Compute the shortest route for three cities in the map
- 3! = 3x2x1 different routes

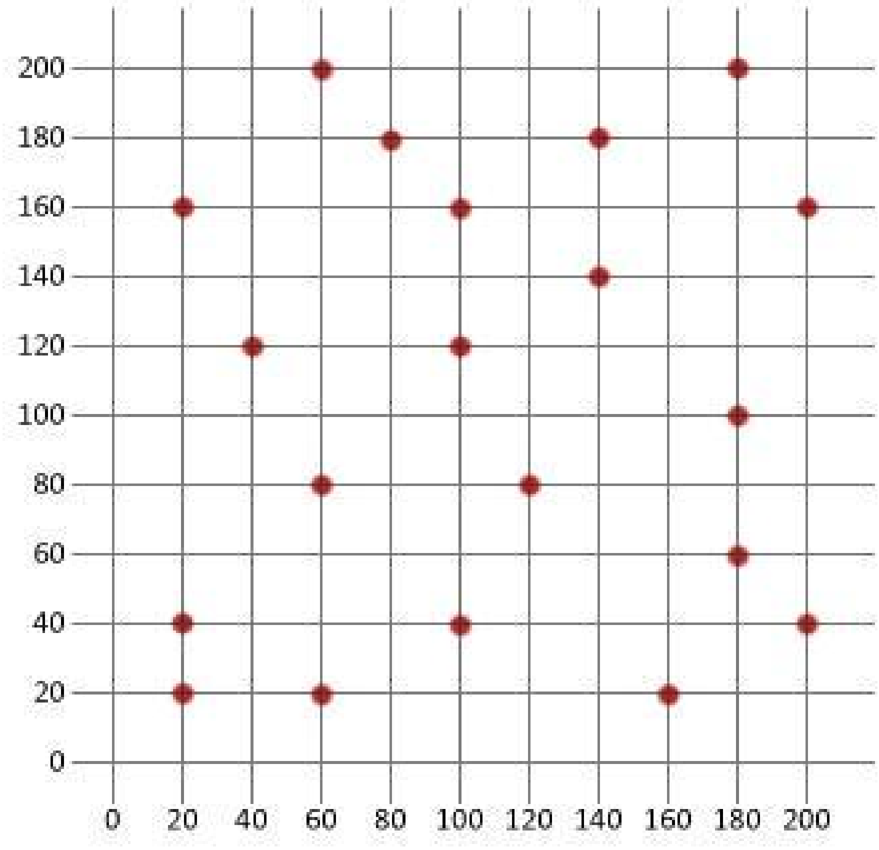




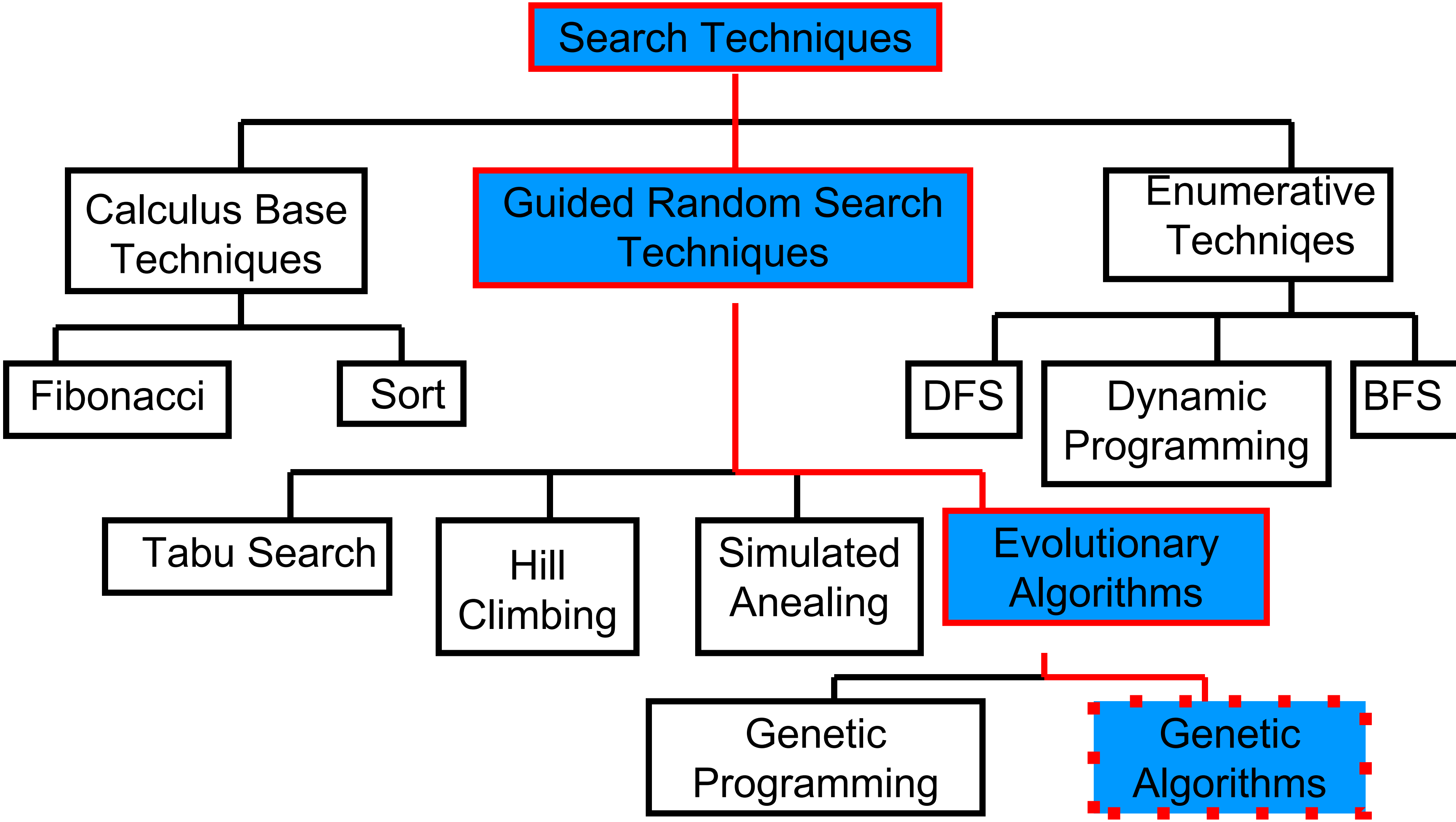
NEED FOR GENETIC ALGORITHM (contd..)

- 20 cities

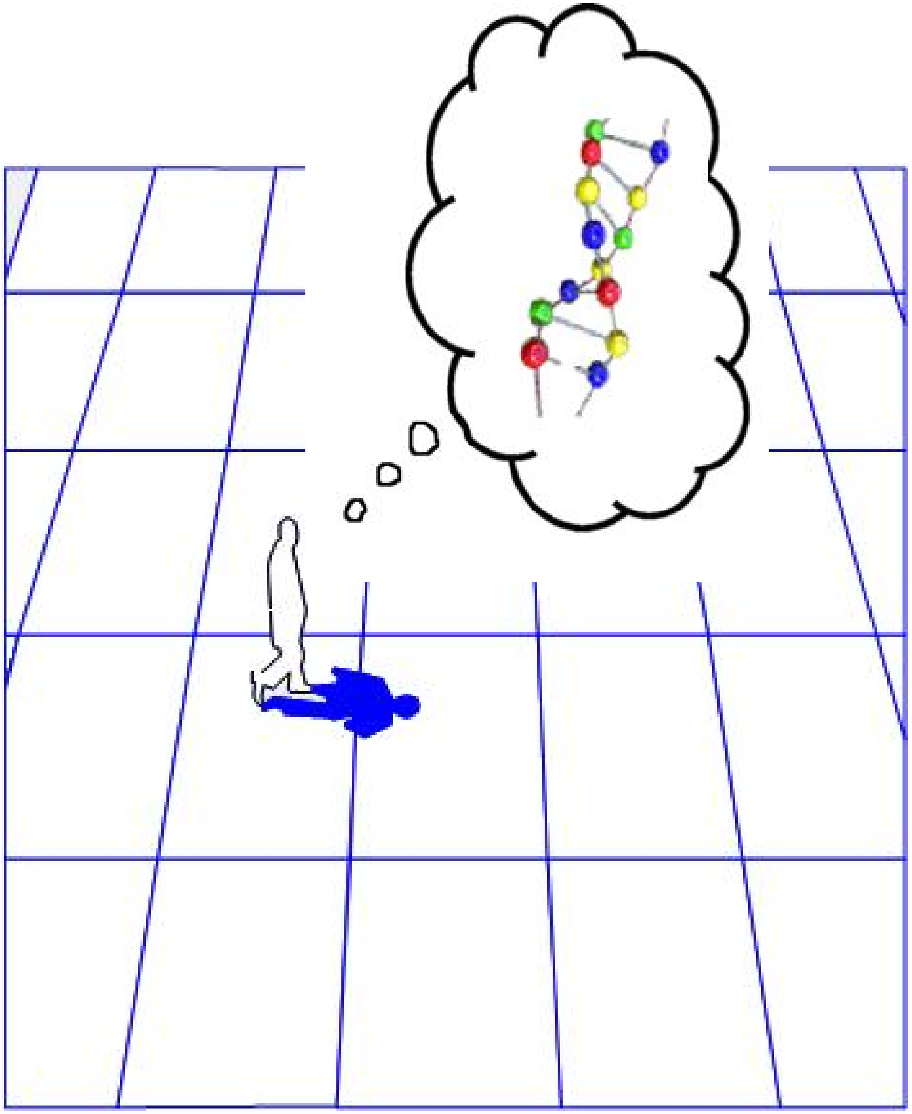
- 20! = 2432902008176640000
  different routes



Classes of Search Techniques



Genetic Algorithm



"Genetic Algorithms are good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, solutions you might not otherwise find in a lifetime."

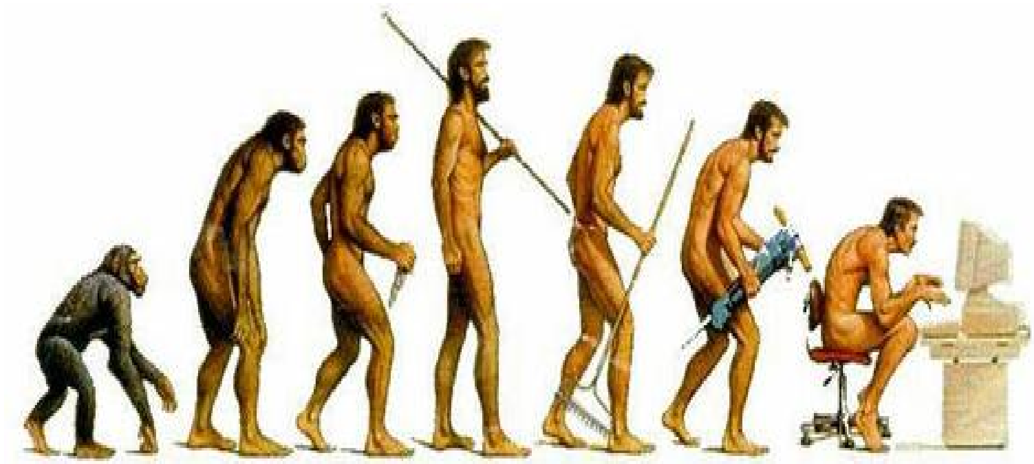
- Salvatore Mangano
Computer Design, May 1995

The Genetic Algorithm

- *Search* and *Optimization* techniques based on Darwin’s Principle of *Natural Selection and* the mechanics of genetics.

- Developed by John Holland, University of Michigan (1970’s)
  - To understand the adaptive processes of natural systems
  - To design artificial systems software that retains the robustness of natural systems

GENETIC ALGORITHM



Charles Darwin	John Holland
1859	1975
Origin of the Species	**Genetic Algorithms**
Survival of the Fittest	Artificial Survival of the Fittest

Biological Background – Natural Selection

- The origin of species:
"**Preservation** of favorable variations and **rejection** of unfavorable variations."

- More individuals born than can survive, so continuous struggle for life.

- Individuals with an advantage have a greater chance for survive: **survival of the fittest**.

- Important aspects in natural selection are
  - Adaptation to the environment
  - Isolation of populations in different groups which cannot mutually mate

Evolution Through Natural Selection

Initial Population of Living Beings

Struggle For Existence-Survival Of the Fittest

Surviving Individuals Reproduce, Propagate Favorable Characteristics

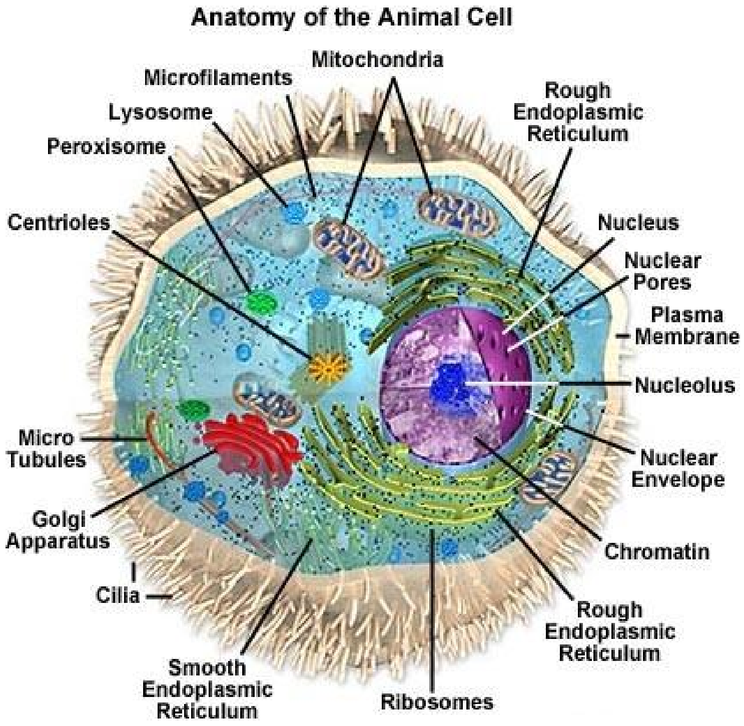
Millions Of Years

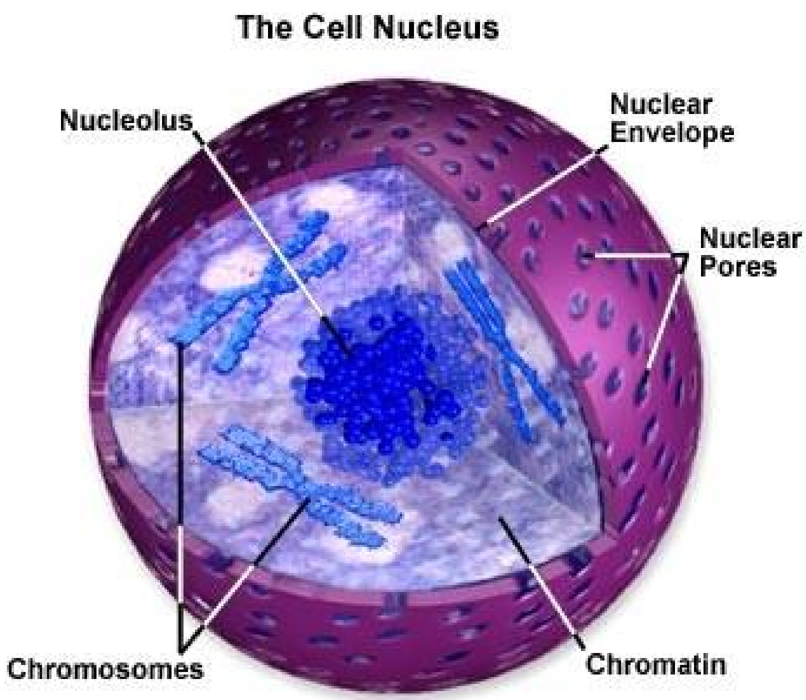
Evolved Species

(Favorable Characteristic Now A Trait Of Species)

Biological Background "The cell"

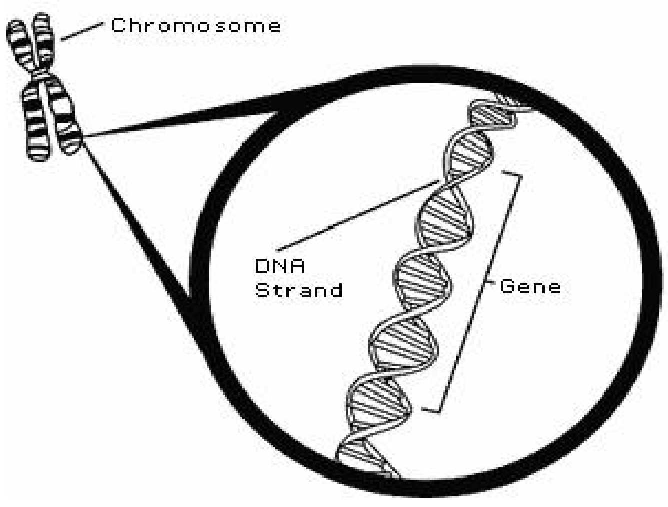
- Every animal cell is a complex of many small "factories" working together.
- The nucleus in the center of the cell.
- The nucleus contains the genetic information





Biological Background "Chromosomes"

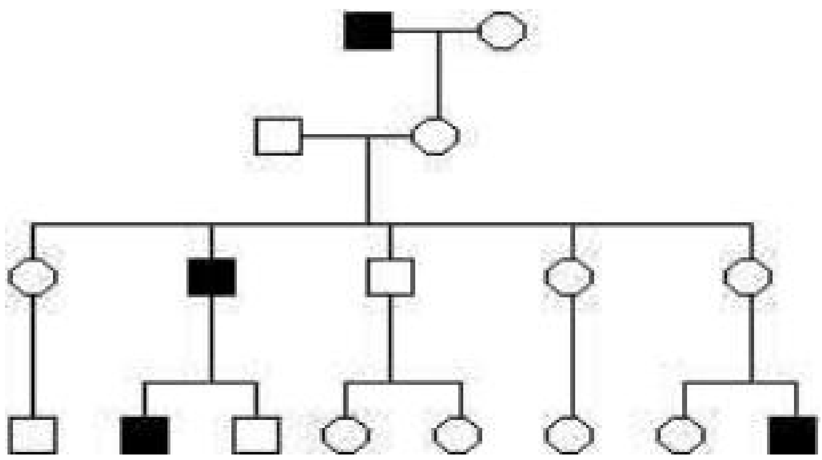
- Genetic information is stored in the chromosomes.
- Each chromosome contains a set of ***genes*** - blocks of DNA
- Each gene determines some aspect of the organism (like eye colour)
- Every gene has an unique position on the chromosome: *locus*



Biological Background "Genetics"

- A gene or genetic locus may have a number of alternative forms: Allele
- Alleles can be either dominant or recessive
- Complete set of genetic material(all chromosomes) is called genome.
- Particular set of genes in a genome is called genotype.
- A genotype develops

  to a phenotype.



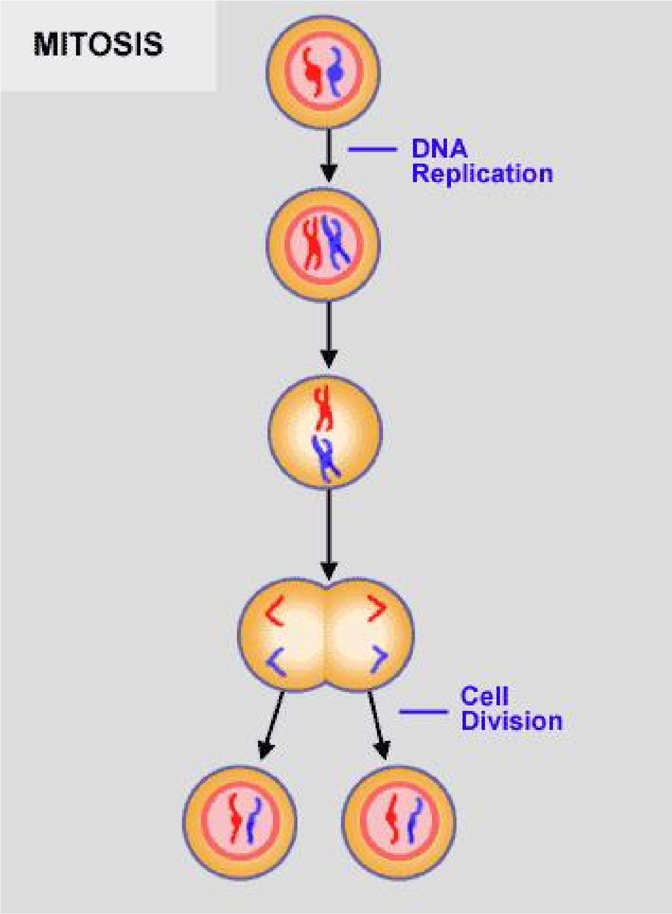
Biological Background “Reproduction ”

- Reproduction of genetical information

  -Mitosis   - Meiosis

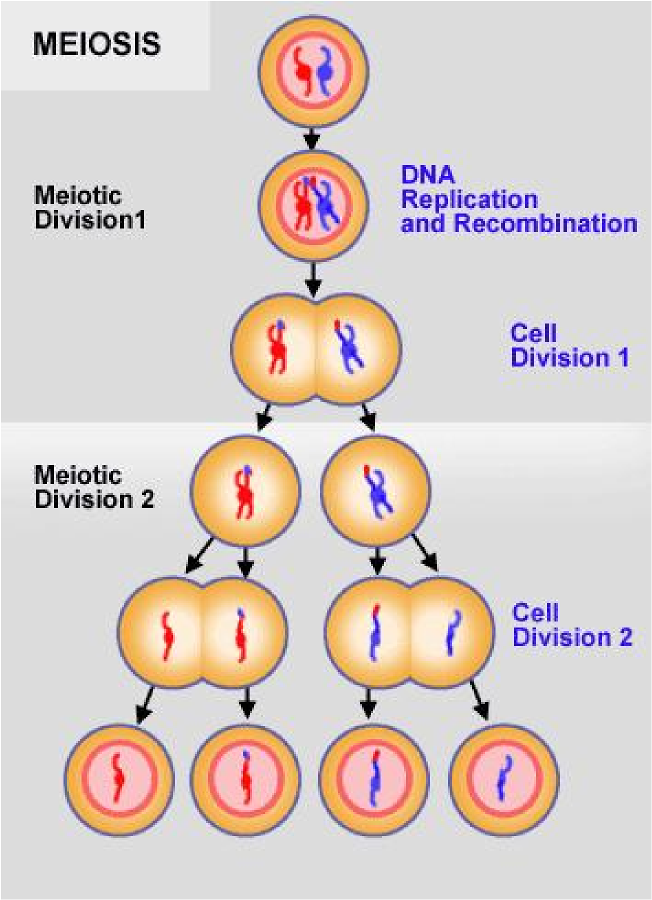
- Mitosis is copying the same genetic information to new offspring:

  -there is no exchange of information



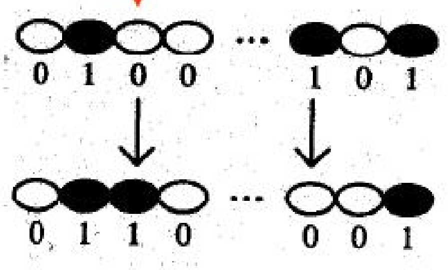
Biological Background Reproduction

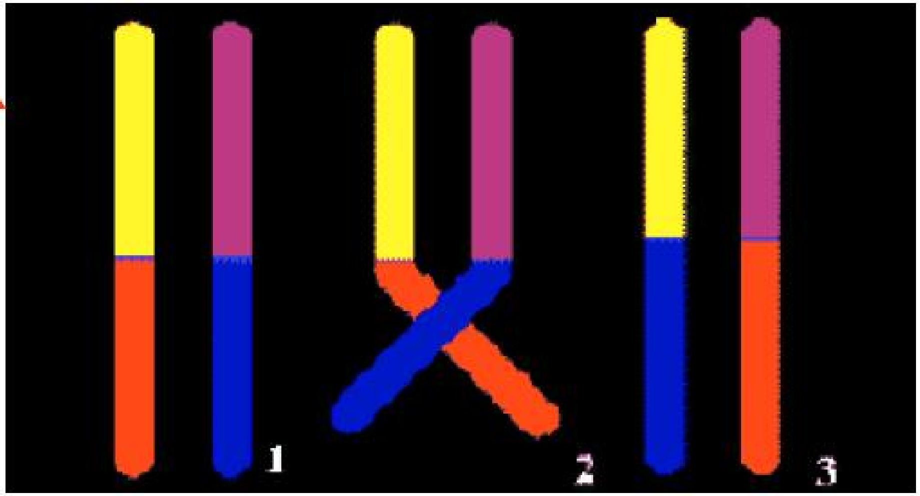
- Meiosis is the basis of sexual reproduction
- Occurs in 2 stages
- Genetic information
  is shared between the parents
  in order to create new offspring
  (Cross-over)



Biological Background "Reproduction "

- During reproduction "errors" occur
- Due to these "errors" genetic variation exists
- Most important "errors" are:
  - Recombination (cross-over)
  - Mutation





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
Selection, Mutation	Stochastic Operators

The Genetic Algorithm

- In evolutionary systems, **populations** evolve by selective pressures, mating between individuals, and alterations such as mutations.

- A genetic algorithm simulates Darwinian theory of evolution using operators such as: **reproduction, mutation** and **crossover.**

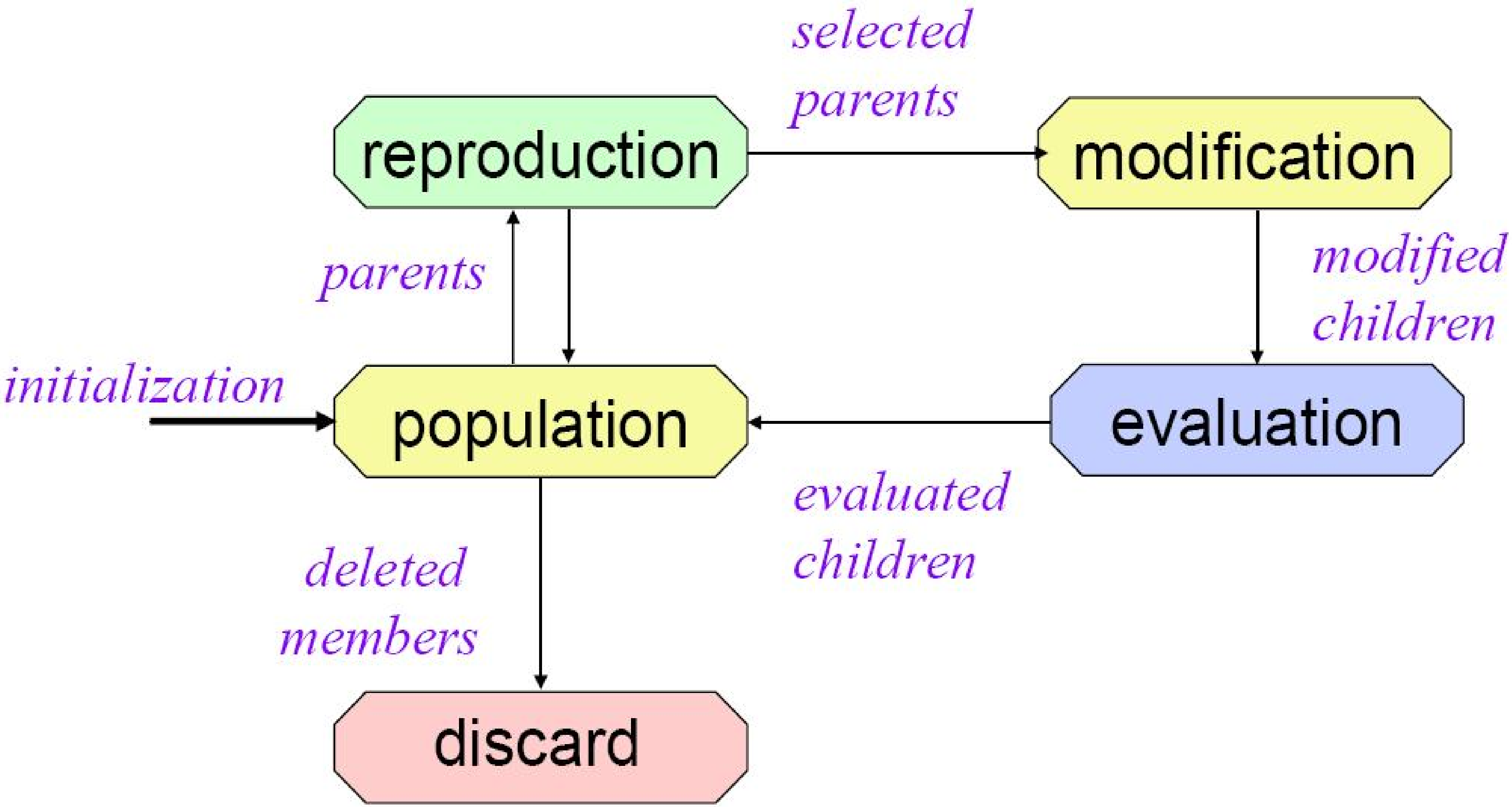
The Genetic Algorithm

- Determine the initial **population** of creatures
- Determine the **fitness** of the population
- **Reproduce** the population using the fittest parents of the last generation
- Determine the **crossover** point, this can also be random
- Determine if **mutation** occurs and if so on which creature(s)
- •Repeat from step 2 with the new population until condition (X) is true

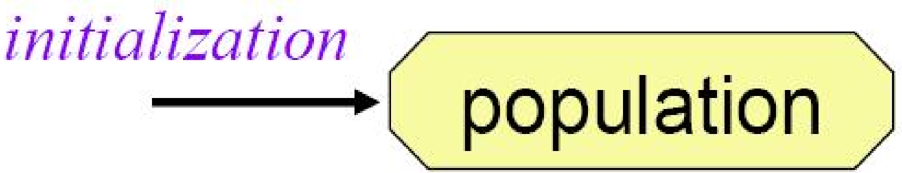
The Genetic Algorithm

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

      While(Fitness Value != Optimal Value)
      {
             Selection;//Natural Selection,
                       //Survival Of Fittest

             Crossover;//Reproduction,
                       //Propagate favorable characters


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

Cycle of Reproduction



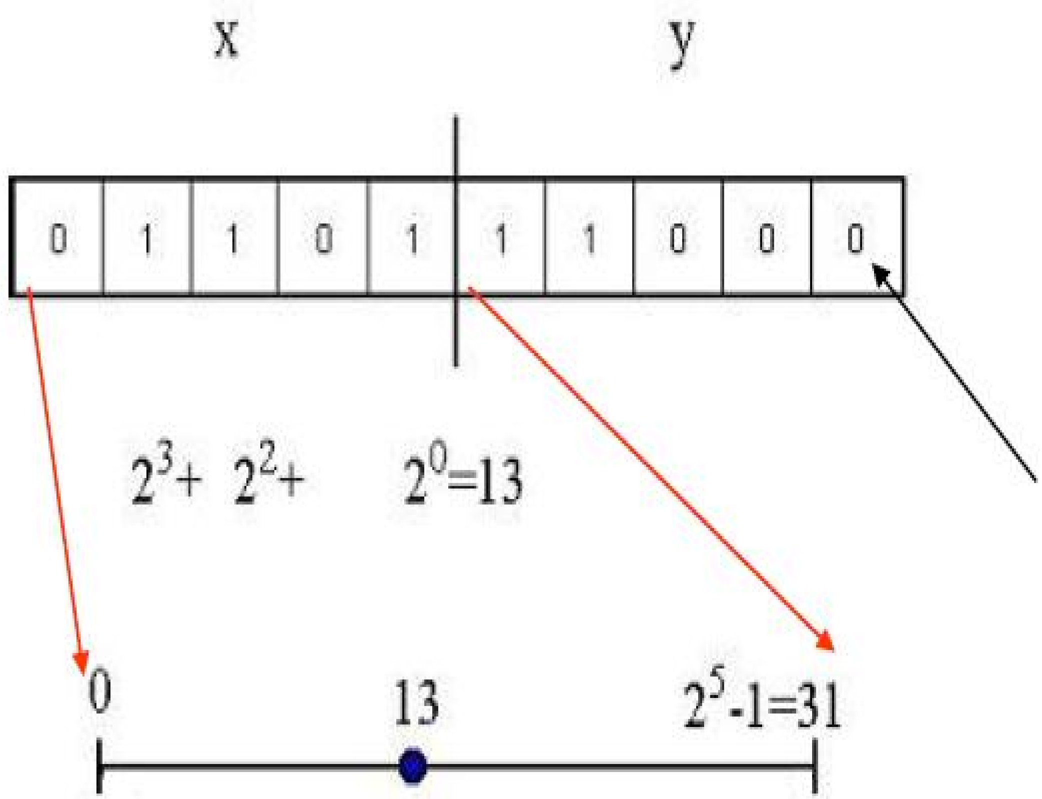
Cycle of Reproduction, Population



Elements of population (*chromosomes*) could be:

- Bit strings (**01100010110100111100**)
- Real numbers (**43.2, -33.1, ... 0.0, 89.2**)
- Permutations of element (**E11, E3, E7, ... E1, E15**)
- Lists of rules (**R1, R2, R3, ... R22, R23**)
- Program elements (**genetic, programming**)
- ... any data structure ...

Bit string chromosome



parameters

chromosome (bit string)

a gen = 1 bit

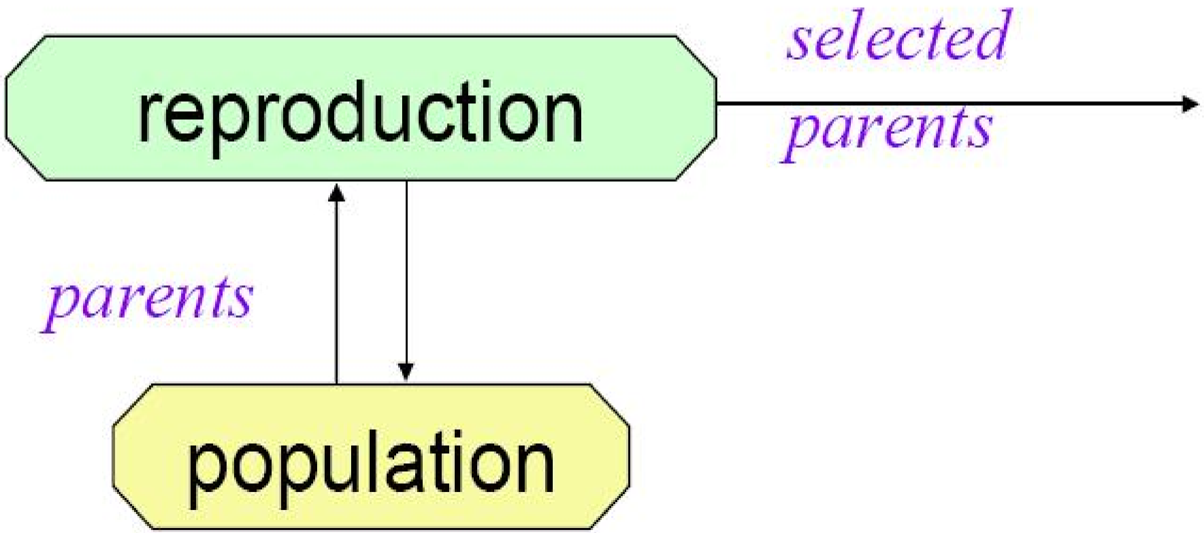
range of integer number

Permutation chromosome

- **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.

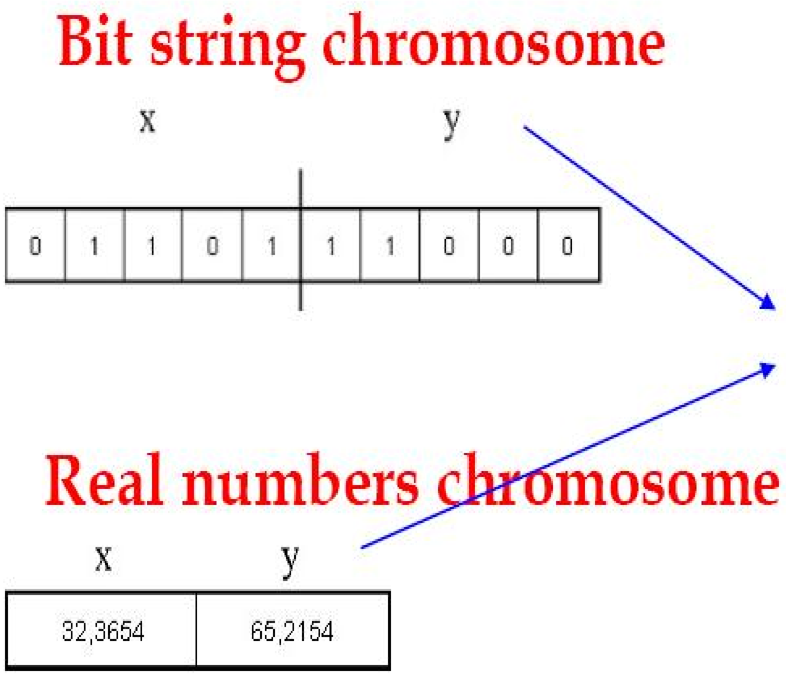
Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Cycle of Reproduction; Reproduction



Parents are selected at random with selection chances biased in relation to chromosome evaluations (*fitness function*).

Reproduction; Fitness function



Fitness(x,y) *is any calculated value*

Fitness Function

- The fitness of the parents(string) is always defined *with respect to other members of the current population.*
-
- *Fitness is defined by:* f_i / f_A , where f_i is the evaluation associated with string *i* and f_A is the average evaluation of all the strings in the population.
- After calculating f_i / f_A for all the strings in the current population, selection is carried out.
- The probability that strings in the current population are copied (i.e. duplicated) and placed in the intermediate generation is in proportion to their fitness.

Fitness Function

- *For each string i where f_i / f_A is greater than 1.0, the integer portion of this number indicates* how many copies of that string are directly placed in the intermediate population.
- For example, a string with f_i / f_A = 1.36 places 1 copy in the intermediate population, and then receives a 0.36 chance of placing a second copy.
- A string with a fitness of f_i / f_A = 0.54 has a 0.54 chance of placing one string in the intermediate population.

Parent Selection

When selecting chromosomes to mate with in a GA simulation, you want to select the best fitting chromosomes to mate in the hopes that their offspring will produce a better solution.

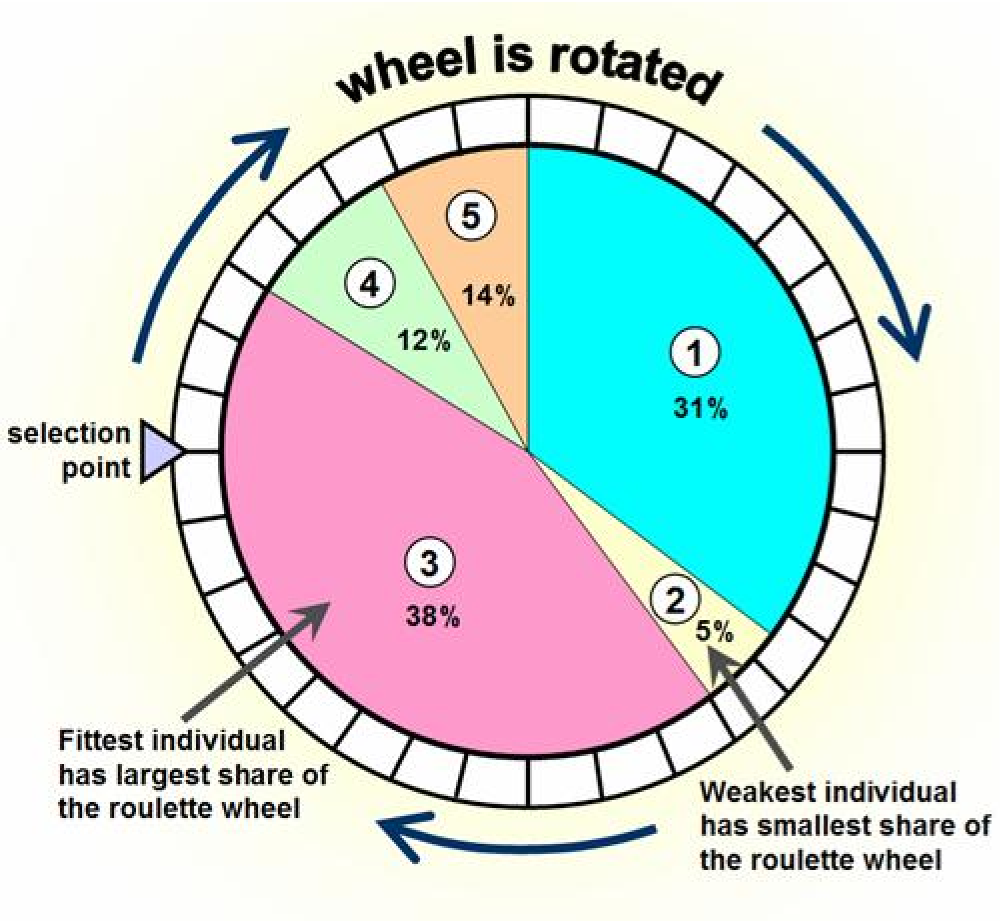
Methods of Selection

- *Roulette-wheel selection*.
- *Elitist selection*.
- *Fitness-proportionate selection*.
- *Scaling selection*.
- *Rank selection*.
- *Generational selection*.
- *Hierarchical selection*.

Roulette wheel selection

No.	String	Fitness	% Of Total
1	01101	169	14
2	11000	576	38
3	01000	64	12
4	10011	361	31
5	00110	35	5
Total		1170	100.0

Roulette wheel selection



Another methods of selection

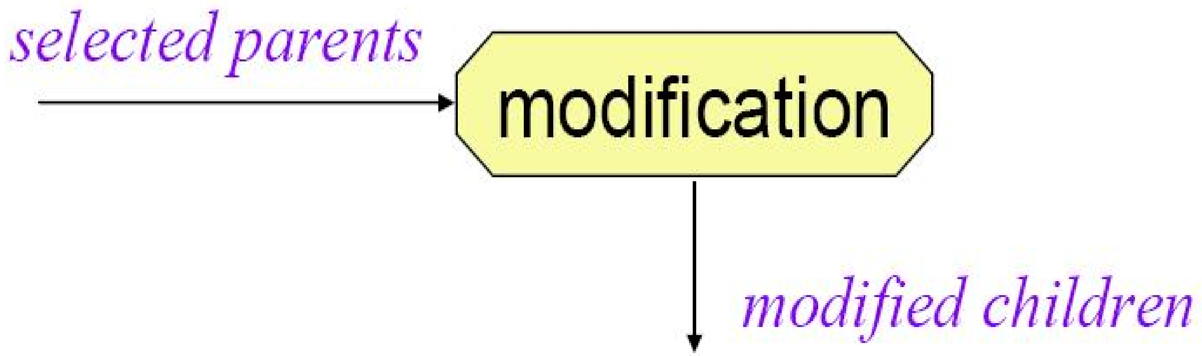
- *Elitist selection*:

  The most fit members of each generation are guaranteed to be selected.

- *Rank selection*:

  Each individual in the population is assigned a numerical rank based on fitness, and selection is based on this ranking.

Cycle of Reproduction; Modification



Modifications are stochastically triggered

- Operator types are:
  - Crossover (recombination) (probability of crossover)
  - Mutation (probability of mutation)

Methods of Reproduction

Crossover

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

- *Two strings are picked from the mating pool at random to cross over.*

One-point crossover

- Randomly one position in the chromosomes is chosen
- Child 1 is head of chromosome of parent 1 with tail of chromosome of parent 2
- Child 2 is head of 2 with tail of 1

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

Two-point crossover

- Randomly two positions in the chromosomes are chosen
- Avoids that genes at the head and genes at the tail of a chromosome are always split when recombined

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

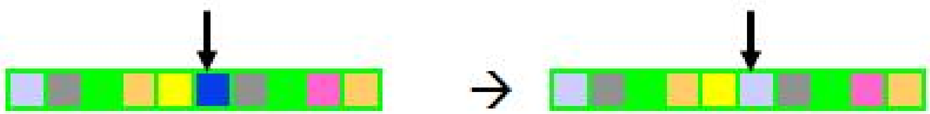
Uniform crossover

- A random mask is generated
- The mask determines which bits are copied from one parent and which from the other parent
- Bit density in mask determines how much material is taken from the other parent (takeover parameter)

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

Mutation

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



random selected positions

Bit string chromosome

Before: (1 0 1 $\boxed{1}$ 0 1 1 0)

After: (0 1 1 $\boxed{0}$ 0 1 1 0)

Real numbers chromosom

Before: (1.38 $\boxed{-69.4}$ 326.44 0.1)

After: (1.38 $\boxed{-67.5}$ 326.44 0.1)

Crossover Probability

A crossover probability of 1.0 indicates that all the selected chromosomes are used in reproduction

better results are achieved by a crossover probability of between 0.65 and 0.85

It implies that the probability of a selected chromosome surviving to the next generation unchanged ranges from 0.35 to 0.15.

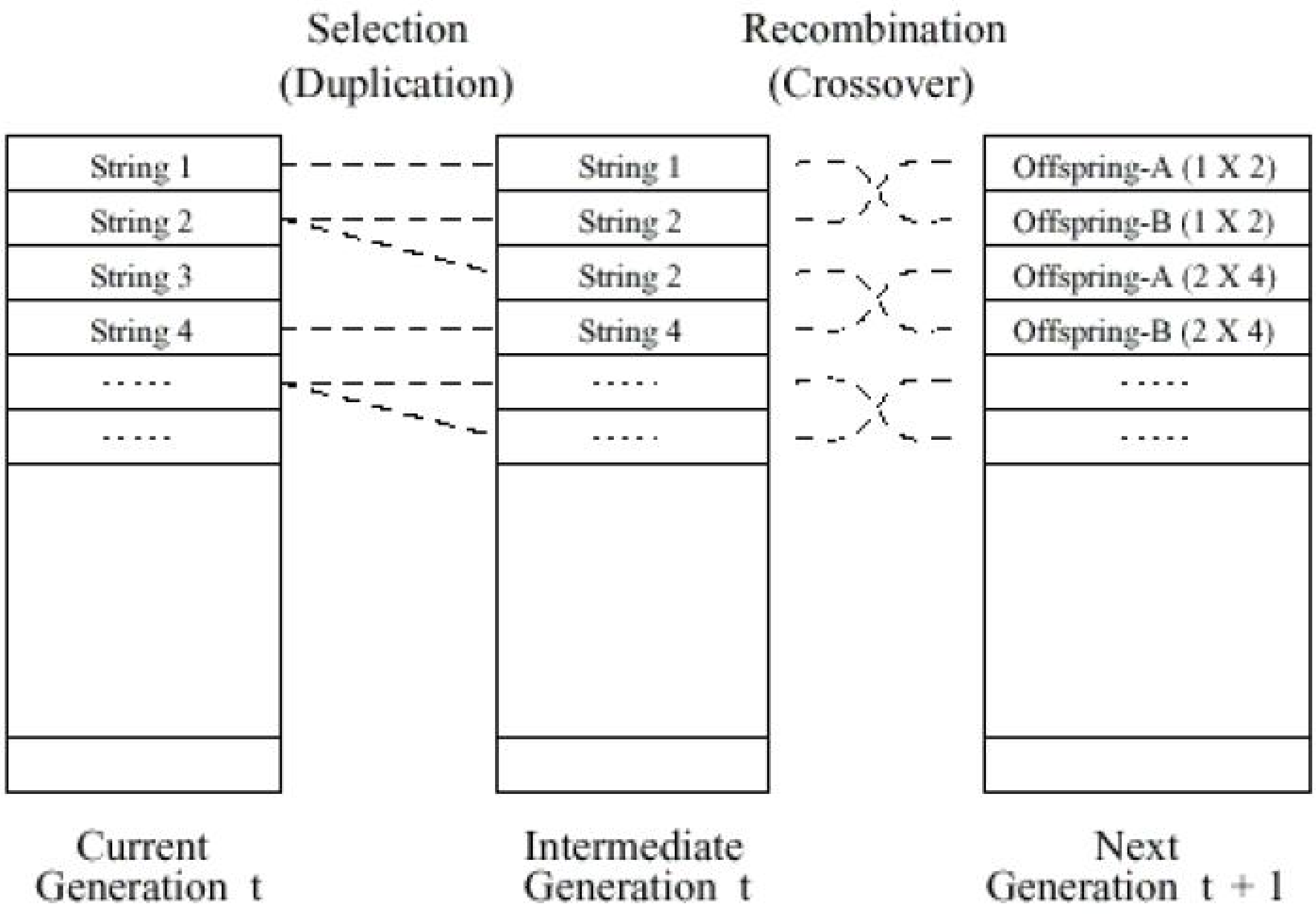
Mutation Probability

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

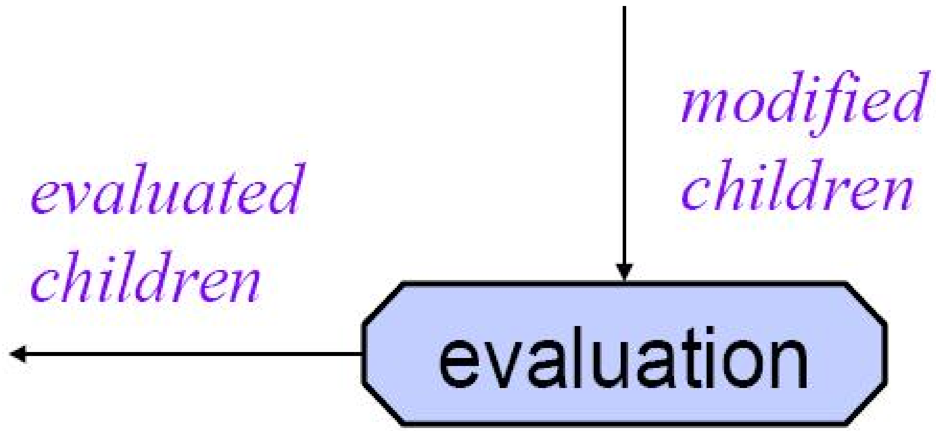
better results are achieved by a mutation probability of between 0 and 1%

Each bit in each chromosome is checked for possible mutation by generating a random number between zero and one and if this number is less than or equal to the given mutation probability e.g. 0.001 then the bit value is changed.

Standard GA

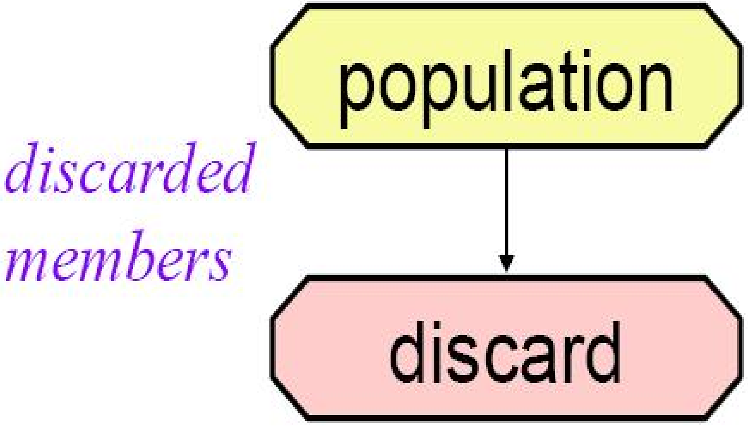

Selection
(Duplication)
Recombination
(Crossover)
String 1
String 2
String 3
String 4
.....
.....
String 1
String 2
String 2
String 4
.....
.....
Offspring-A (1 X 2)
Offspring-B (1 X 2)
Offspring-A (2 X 4)
Offspring-B (2 X 4)
.....
.....
Current
Generation t
Intermediate
Generation t
Next
Generation t + 1

Cycle of Reproduction; Evaluation



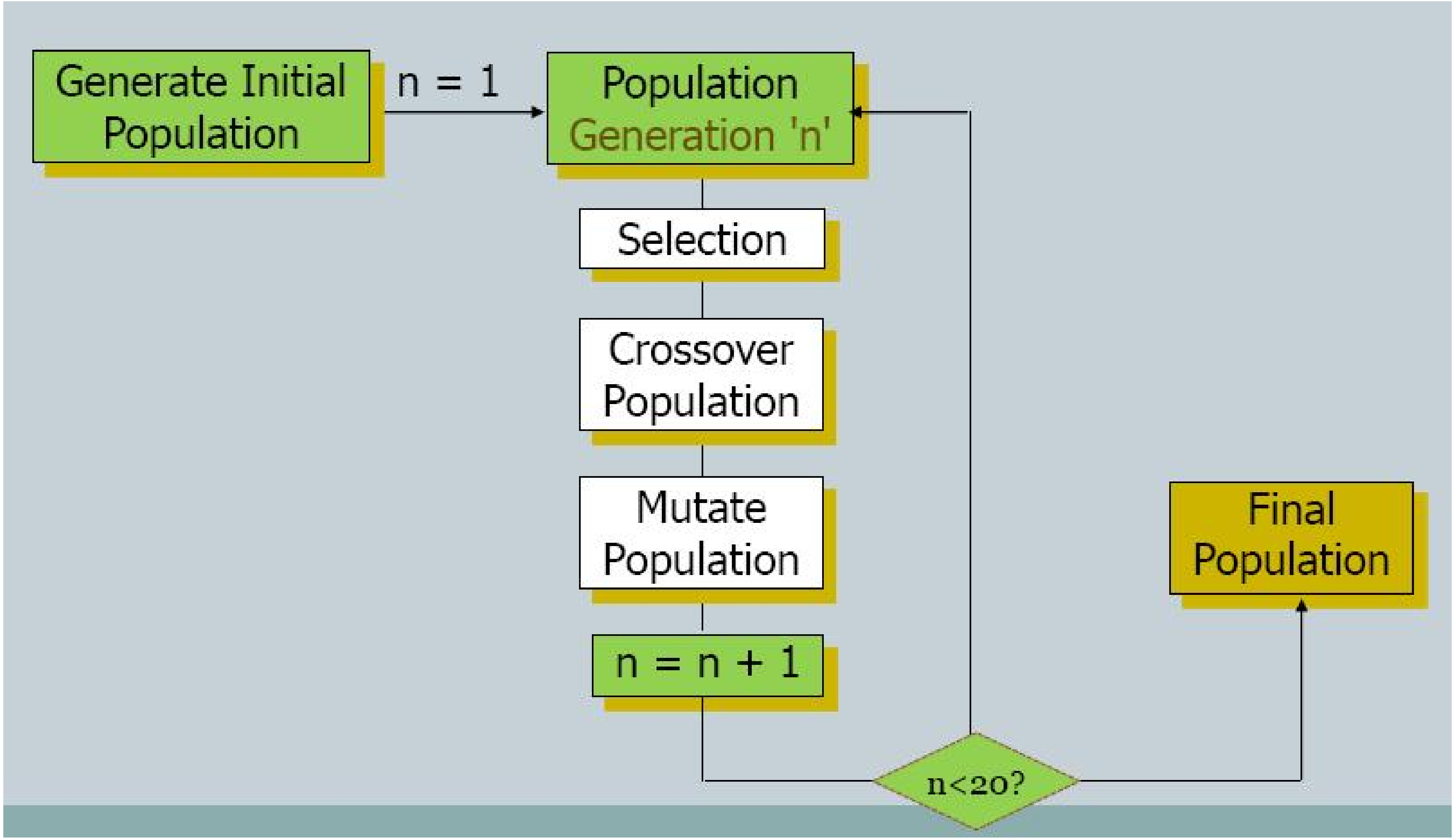
- The evaluator decodes a chromosome and assigns it a fitness measure
- The evaluator is the only link between a classical GA and the problem it is solving

Cycle of Reproduction; Deletion



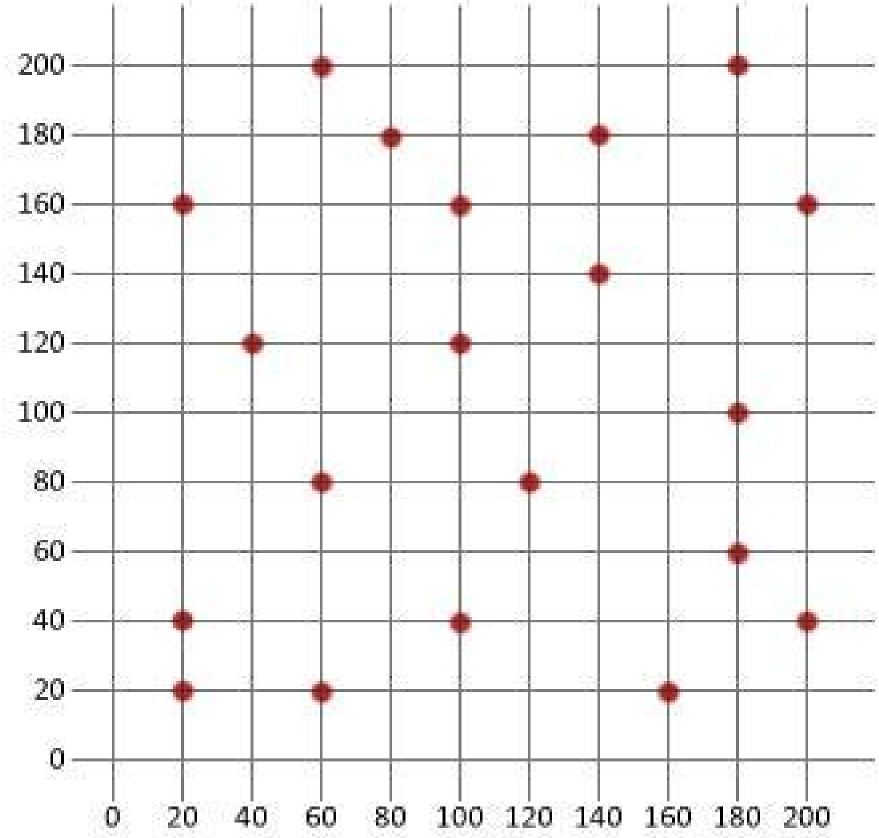
- Generational GA:
  entire populations (fully) replaced with each iteration
- Steady-state GA:
  a few members replaced each generation

Genetic Algorithms Flowchart



Travelling Salesman Problem

- Representation is an ordered list of city numbers known as an ordered based GA



Citylist 1	1	2	3	4	5	6	7	8	9
Citylist 2	9	8	7	6	5	4	3	2	1

Fitness Function and Selection

- For TSP, the fitness function of chromosome is computed at the total distance of the represented solution
- The total distance equation does not have to be the best fitness function.
- As TSP algorithm tends sometime to create quite long distance connections, root mean square (RMS) value could be used too.

- Selection is made based on the **Roulette Wheel.**

Crossover of TSP

- A valid solution would need to represent a route where every city is included at least once and only once.
- To produce valid offspring for our next generation is ordered crossover. In this algorithm we select a subset from our first parent, then add that subset to our child. Finally we add the objects which are not yet in our child to our child in the second parent's order.

Parent 1	1	2	3	4	5	6	7	8	9
Parent 2	9	8	7	6	5	4	3	2	1
Child	9	5	4	3	2	6	7	8	1

Mutation of TSP

- The mutation we use to prevent us evolving invalid solutions is swap mutation. In swap mutation if we have a set of objects we select two objects at random then simply swap their positions. Because we are only swapping objects around we don't risk causing duplicate objects within our solution.

- Before

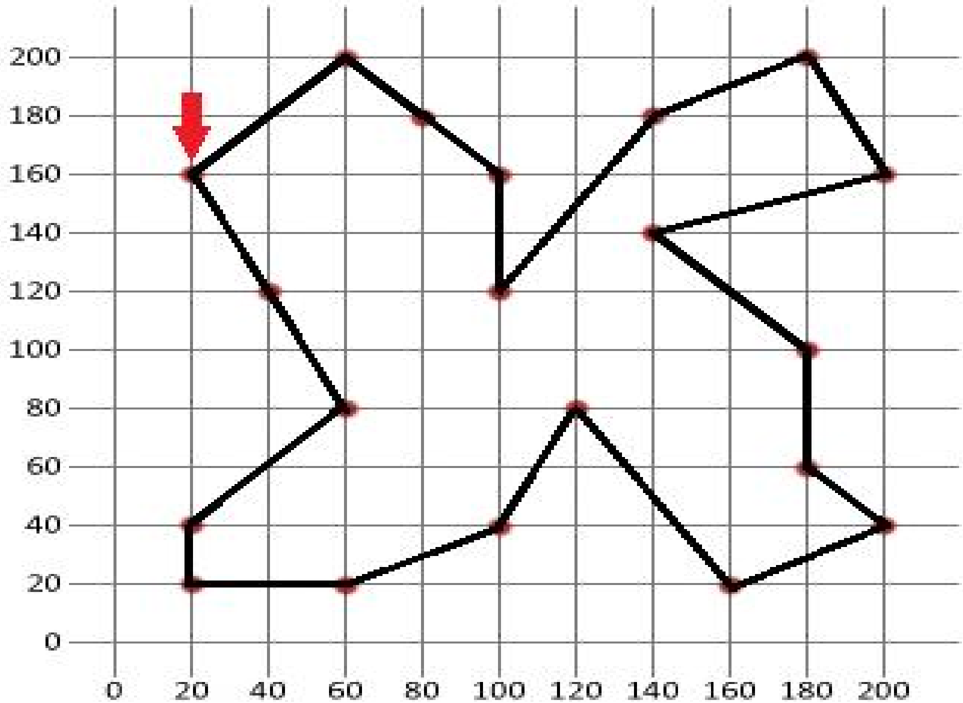
1	2	3	4	5	6	7	8	9

- After

1	2	8	4	5	6	7	3	9

Output

```
Initial distance: 1935.444391860985

Finished

Final distance: 915.0755622990547

Solution:
|20, 160|40, 120|60, 80|20, 40|20, 20|60, 20|100, 40|120, 80|160, 20|200, 40|180
, 60|180, 100|140, 140|200, 160|180, 200|140, 180|100, 120|100, 160|80, 180|60,
200|

Time Taken
0:00:03.631208
```



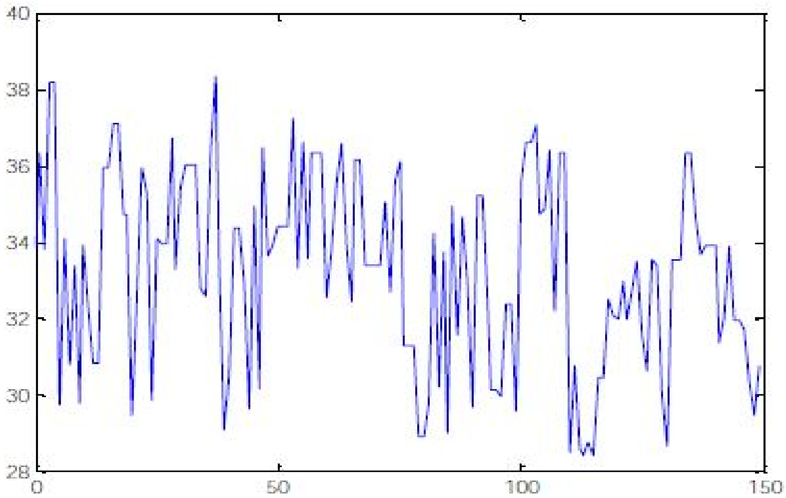
Example of Genetic Algorithm

Genetic Algorithm for finding maximum values of

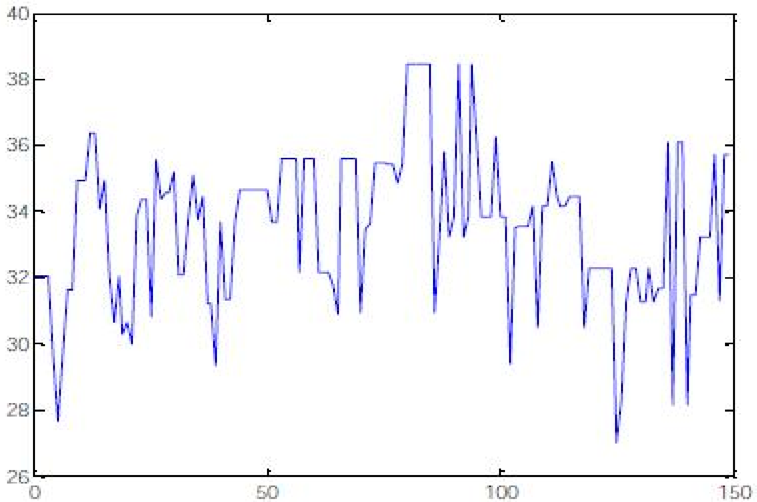
$f(x1, x2) = 21.5 + x1\sin(4\Pi x1) + x2\sin(20\Pi x2)$

- The chromosome of the population is [$x1$ $x2$]. The initial values of $x1$ and $x2$ are extracted randomly from intervals $-3.0 <= x1 >= 12.1$, $4.1 <= x2 >= 5.8$.
- All the chromosomes are tested for fitness
- Mutation and crossover operations, we must define Pm and Pc. For example, when POP (population size) =100 (must be an even number), Pm=0.1, Pc=0.5.
- Population consists of 100 chromosomes which is formed by two genes ($x1$ and $x2$).

Example cont..



Case#1,**Pm=0.4, Pc=0.5.**

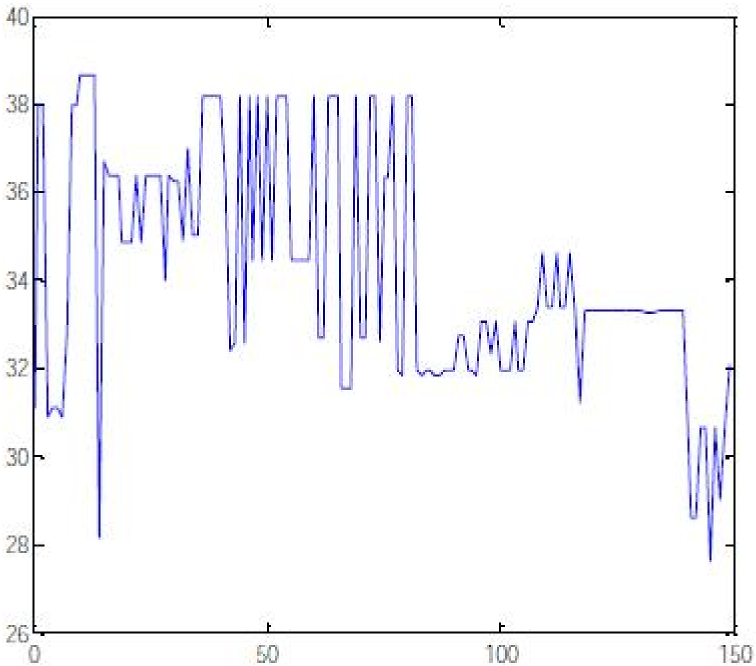


Case#2, **Pm=0.1, Pc=0.65.**

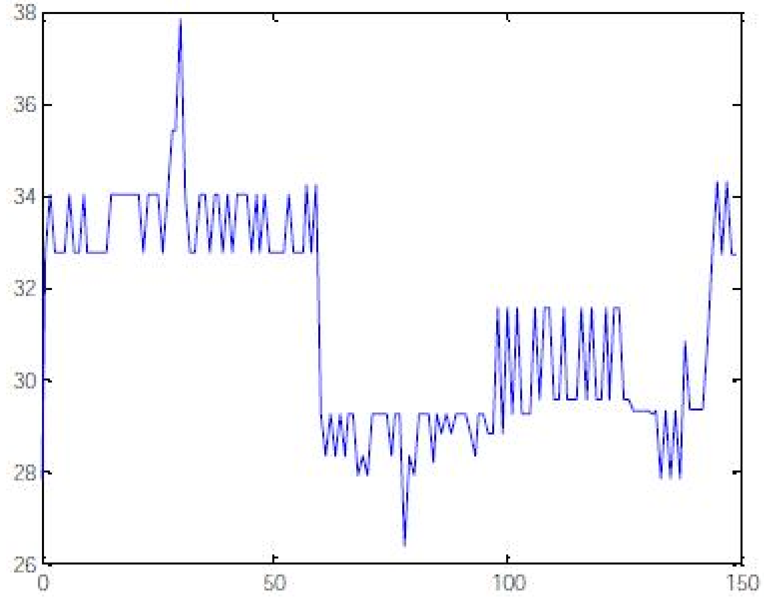
The maximum value of *f* is 38.3329 at (11.1218, 5.7244).

The maximum value of *f* is 38.4845 at (12.0956, 5.7264).

Example cont..



Case#3,**Pm=0.05, Pc=0.8.**



Case#4,**Pm=0.01, Pc=0.99**

The maximum value of *f* is 38.6861 at (11.6168, 5.7222).

The maximum value of *f* is 37.8387 at (11.6220, 4.7249).

Discussion

- If Pm is larger, more new (fresh) values joint in from outside. The fitness picture will fluctuate (up and down) dramatically. But more chances to get optimal values.
- If Pc is larger, the exchange occurs in the population more often. Also more chances to get the optimal values.
- But larger Pm and Pc will increase the computer time dramatically.
- Thus a compromise must be reached between optimal result and computer time.

Comparison

- The genetic algorithm differs from a classical, derivative-based, optimization algorithm in two main ways, as summarized in the following table.

Classical Algorithm	Genetic Algorithm
Generates a single point at each iteration. The sequence of points approaches an optimal solution	Generates a population of points at each iteration. The best point in the population approaches an optimal solution.
Selects the next point in the sequence by a deterministic computation	Selects the next population by computation which uses random number generators

Applications of Genetic Algorithm

- Time series forecasting.
- Robot trajectory planning
- Image Processing
- Design–semiconductor layout, aircraft design, keyboard configuration, communication networks
- Scheduling–manufacturing, facility scheduling, resource allocation
- Machine Learning–Designing neural networks, both architecture and weights, improving classification algorithms, classifier systems
- Signal Processing–filter design
- Combinatorial Optimization–set covering, traveling salesman (TSP), Sequence scheduling, routing, bin packing, graph coloring and partitioning

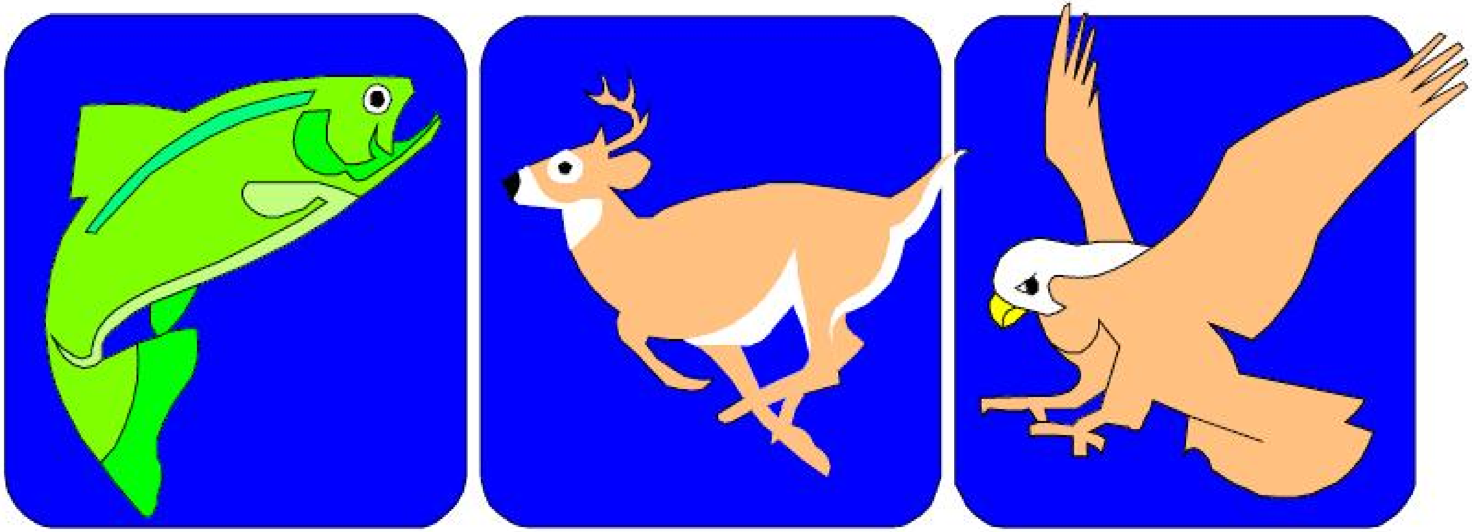
Advantages of Genetic Algorithm

- Parallelism
- Solution space is wider.
- Easy to discover global optimum
- Only uses function evaluations.
- Handles noisy functions well.
- Handles large, poorly understood search spaces easily
- They require no knowledge or gradient information about the response surface
- Discontinuities present on the response surface have little effect on overall optimization performance
- They are resistant to becoming trapped in local optima

Limitations of Genetic Algorithm

- The problem of identifying fitness function
- The problem of choosing the various parameters like the size of the population, mutation rate, cross over rate, the selection method and its strength.
- Cannot easily incorporate problem specific information
- Not good at identifying local optima
- No effective terminator.
- Not effective for smooth unimodal functions
- Needs to be coupled with a local search technique.
- Require large number of response (fitness) function evaluations

THANK YOU



"The Gene is by far the most sophisticated program around."

- Bill Gates, *Business Week*, June 27, 1994