Welcome to:

Evolutionary Intelligence

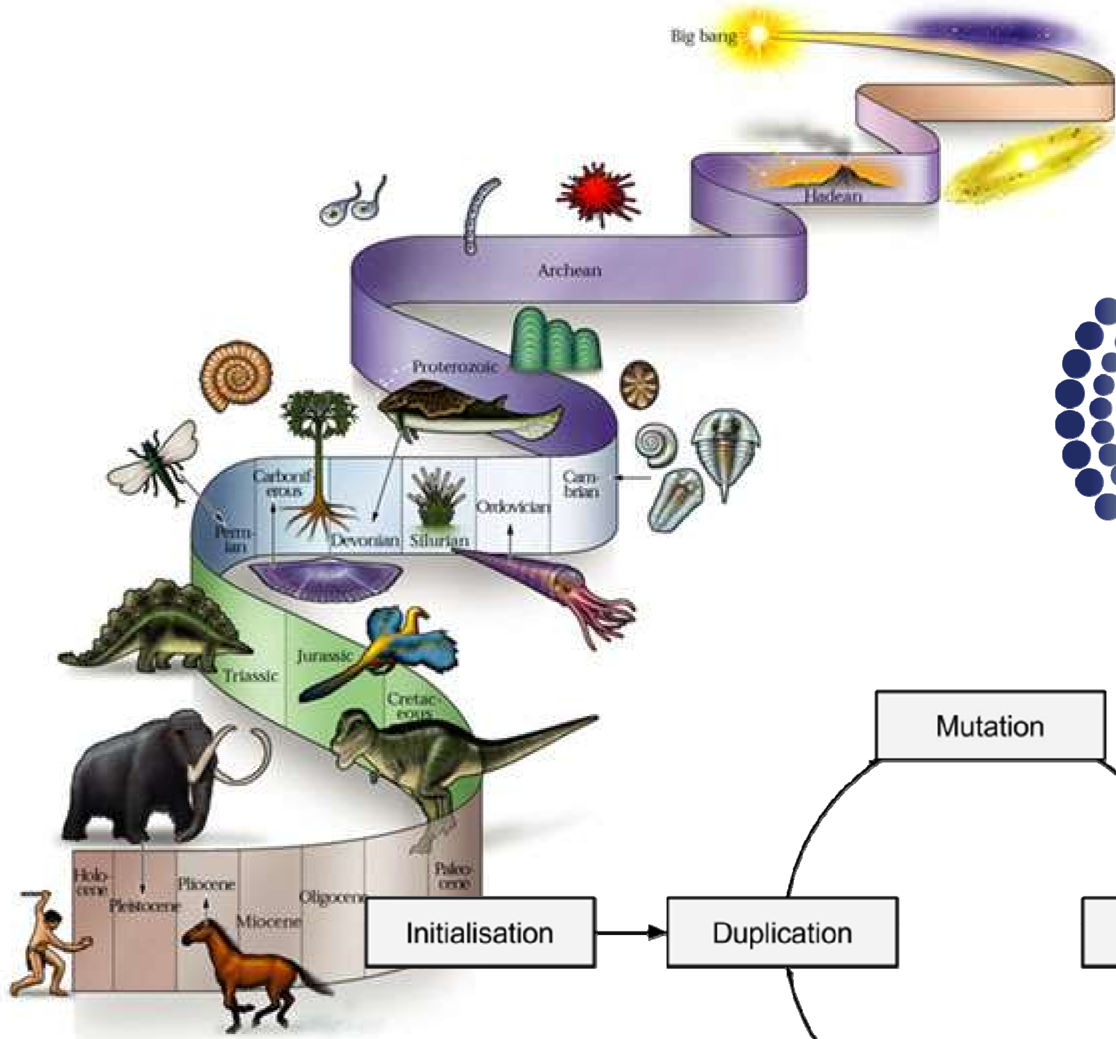
Unit objectives

After completing this unit, you should be able to:

- Understand the importance of genetics
- Learn about the role of Genetics in Artificial Intelligence
- Gain knowledge on the phases in Genetics Algorithms for various applications
- Understand the application of Genetic Algorithm in Travelling Salesman Problem
- Learn about Ant Colony Systems
- Understand the concept of Particle Swarm Optimization

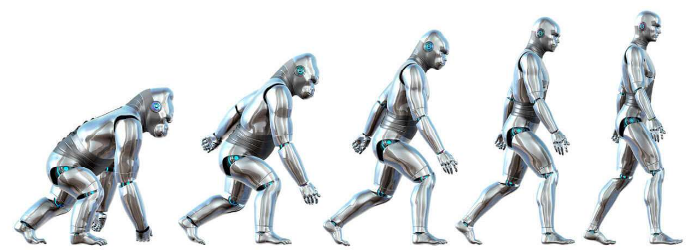
Evolutionary intelligence

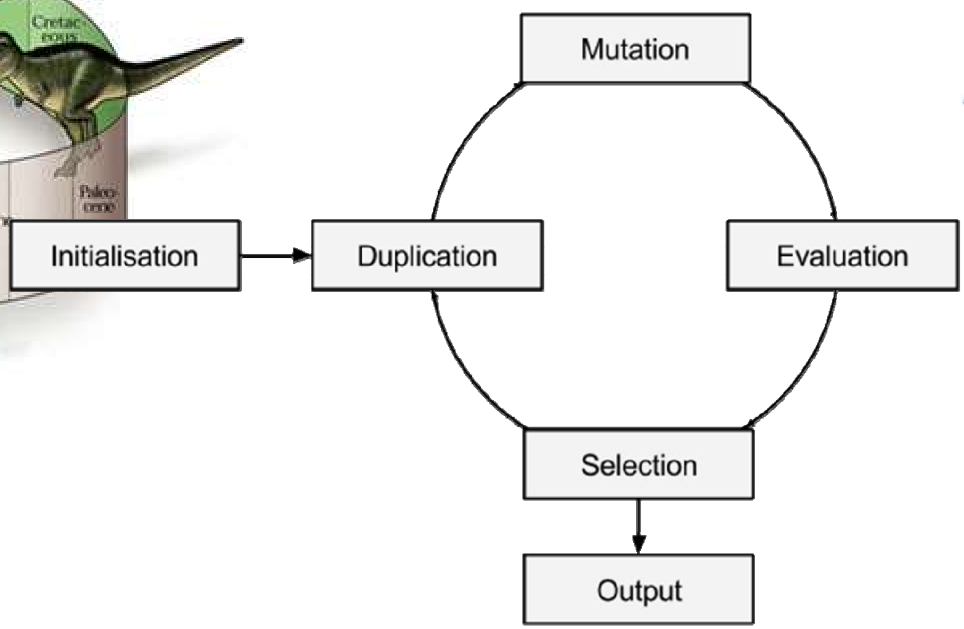
IBM ICE (Innovation Centre for Education)



Evolutionary Intelligence

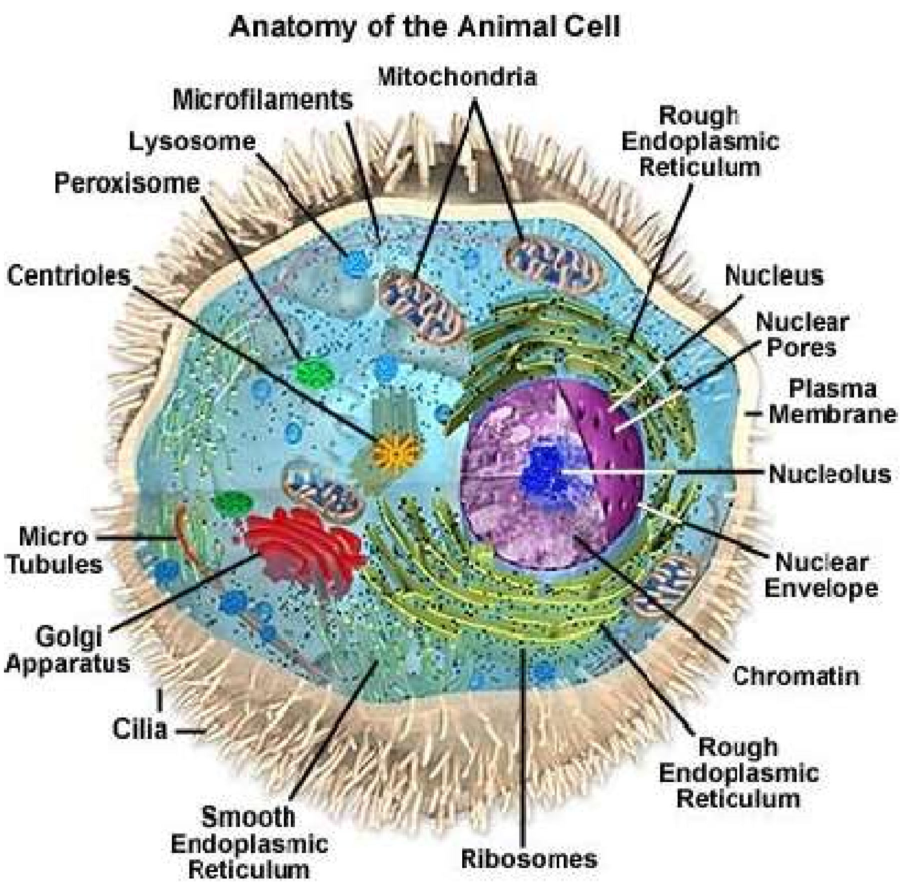
Empowering Conscious Change

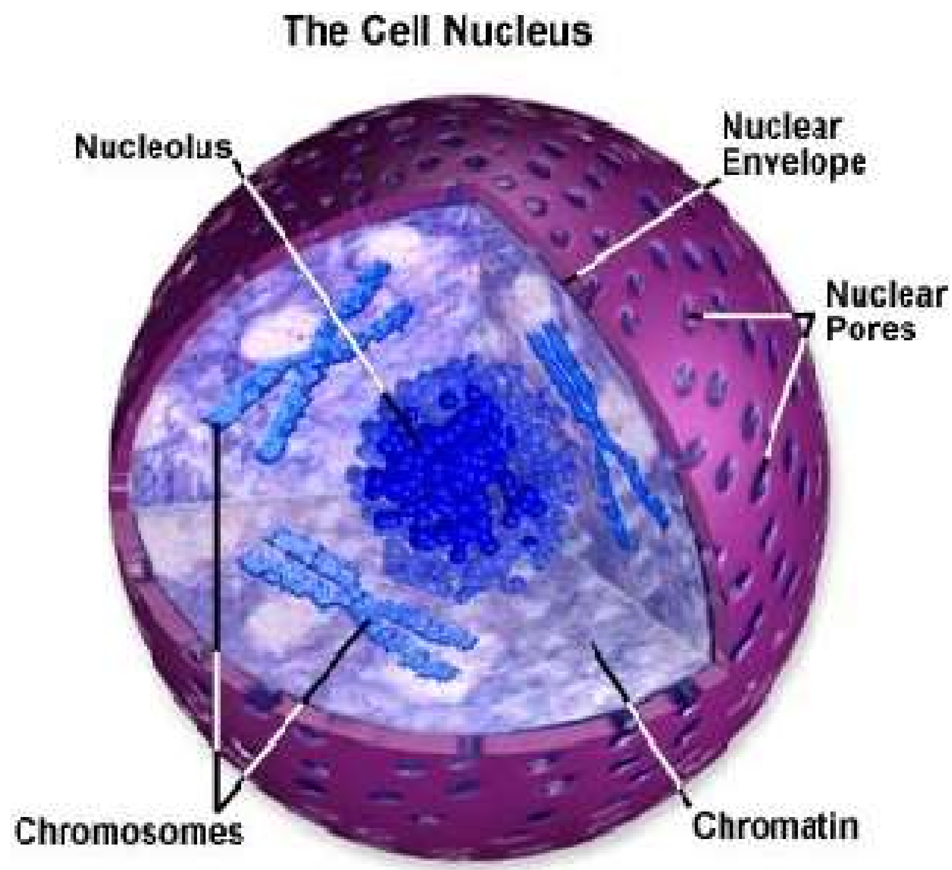




Biological background - The cell

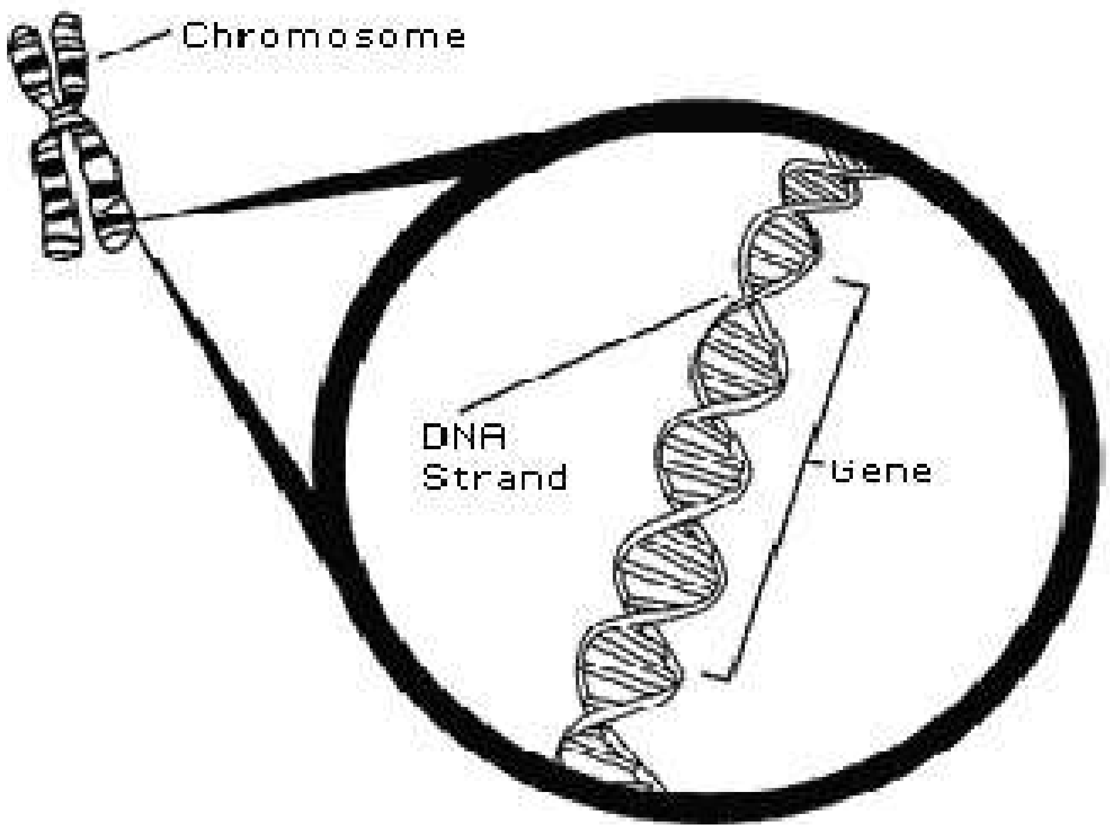
- Every animal cell is a complex of many small factories working together.
- The nucleus in the center of the cell.
- The nucleus contains the genetic information





Chromosome, genes and genomes

- Genetic information is stored in the chromosomes
- Each chromosome is build of DNA
- Chromosomes in humans form 23 number of pairs.
- The chromosome is divided in parts: genes
- Genes code for properties
- Every gene has an unique position on the chromosome: locus

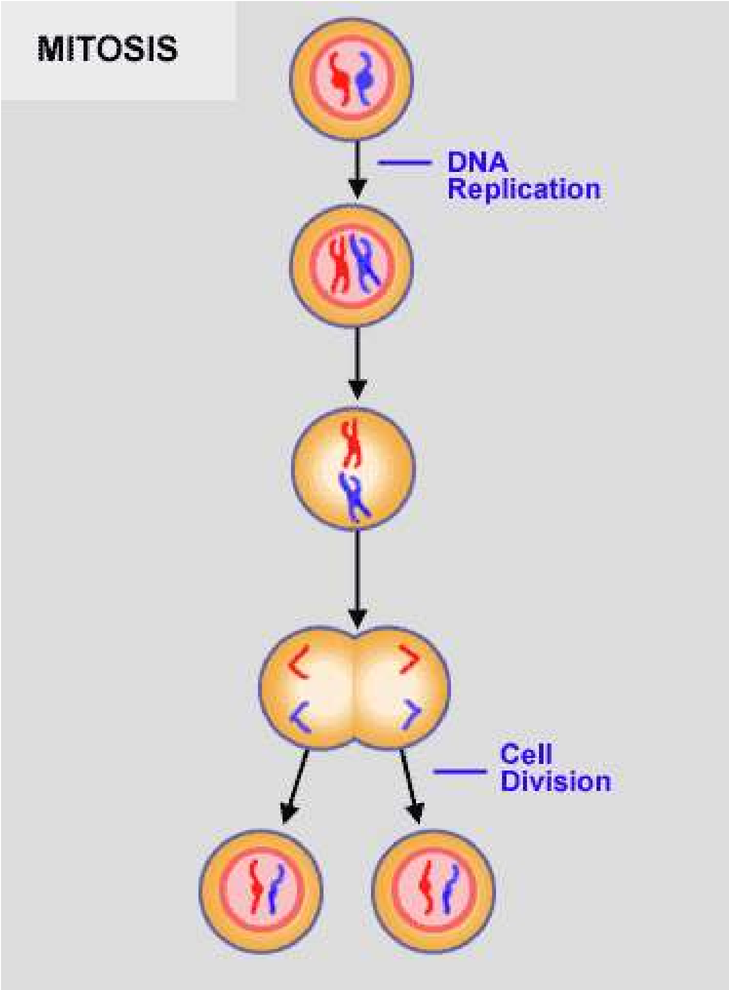


Reproduction

- Reproduction of genetical information
  - Mitosis
  - Meiosis

- Mitosis is copying the same genetic information to new offspring: there is no exchange of information

- Mitosis is the normal way of growing of multicell structures, like organs.



Natural selection

- The origin of species: “Preservation of favorable variations and rejection of unfavorable variations.”

- There are more individuals born than can survive, so there is a continuous struggle for life.

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

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

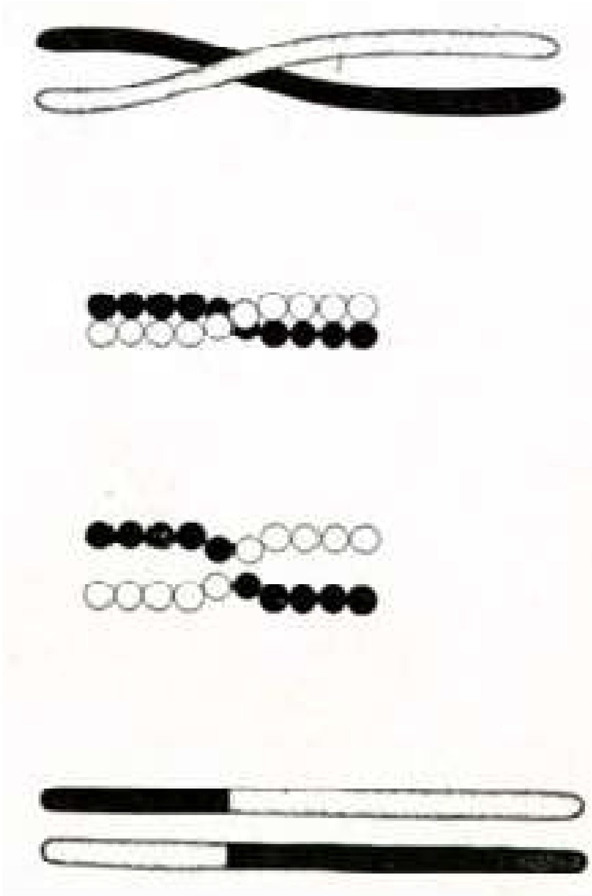
- If small changes in the genotypes of individuals are expressed easily, especially in small populations, we speak of genetic drift

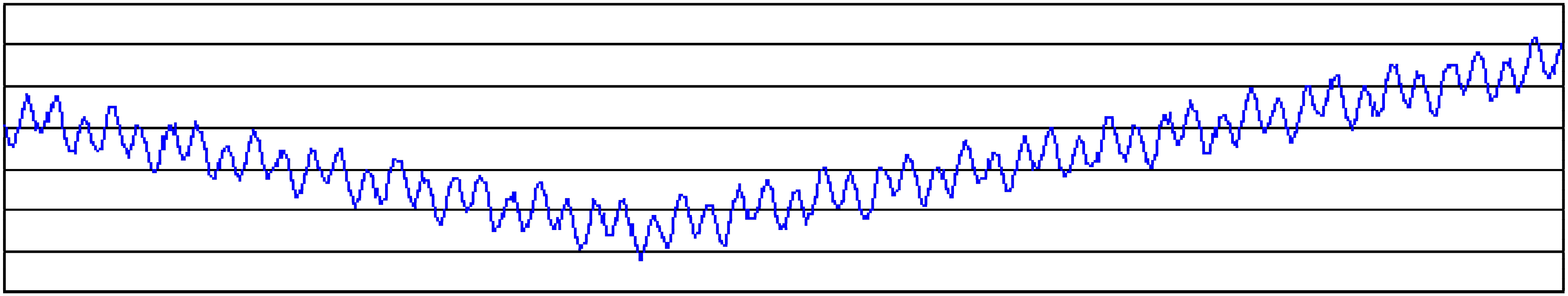
- Mathematical expresses as fitness: success in life

Inspiration - Evolution

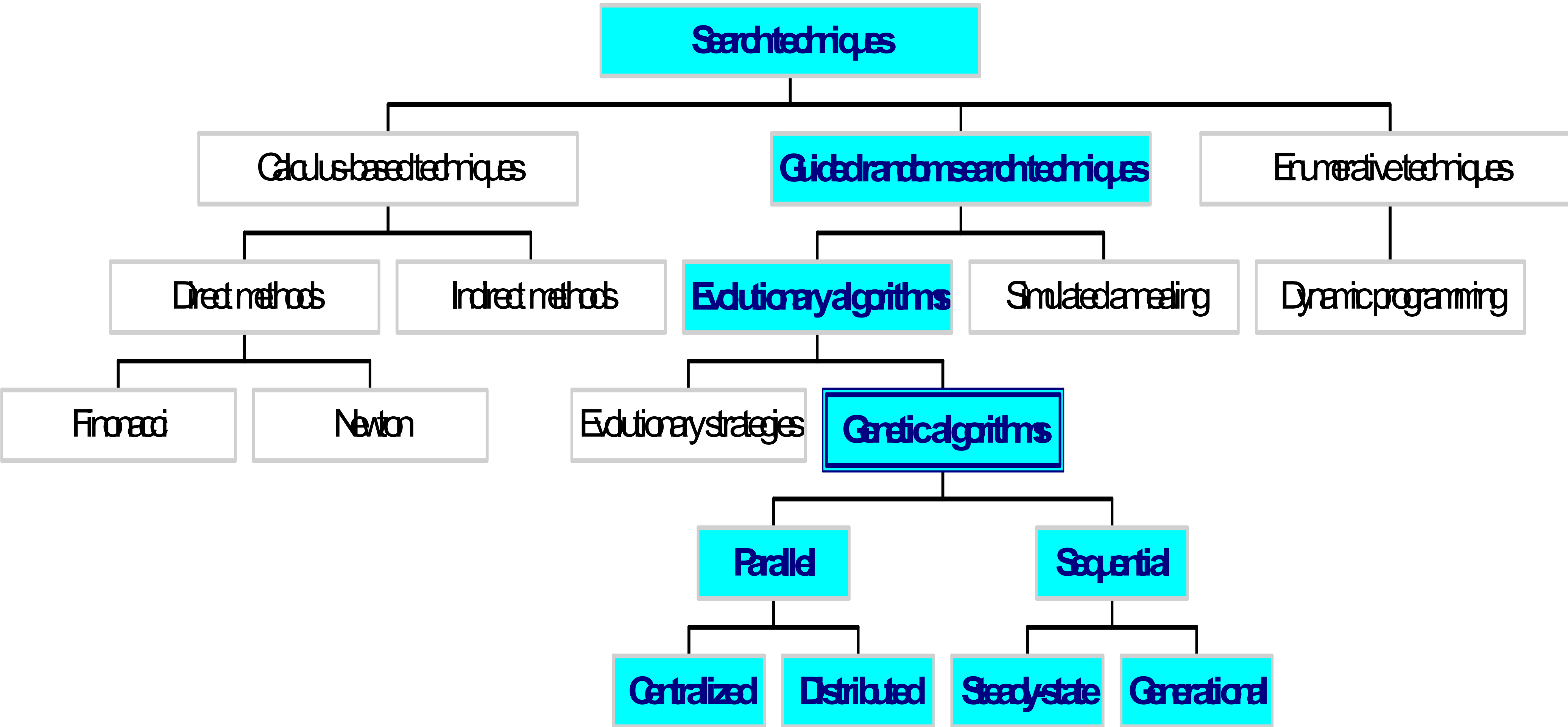
- Natural Selection:
  - “Survival of the Fittest”
  - favourable traits become common and unfavourable traits become uncommon in successive generations

- Sexual Reproduction:
  - Chromosomal crossover and genetic recombination
  - population is genetically variable
  - adaptive evolution is facilitated
  - unfavourable mutations are eliminated





Classes of search techniques



Introduction - Genetic algorithm

- After scientists became disillusioned with classical and neo-classical attempts at modeling intelligence, they looked in other directions.
- Two prominent fields arose, connectionism (neural networking, parallel processing) and evolutionary computing.
- It is the latter that this essay deals with - genetic algorithms and genetic programming.
- As early as 1962, John Holland's work on adaptive systems laid the foundation for later developments in Genetic Algorithms.
  - To understand the adaptive processes of natural systems
  - To design artificial systems software that retains the robustness of natural systems
- By 1975, the publication of the book Adaptation in Natural and Artificial Systems, by Holland and his students and colleagues.
- Early to mid-1980s, genetic algorithms were being applied to a broad range of subjects. In 1992, John Koza has used genetic algorithm to evolve programs to perform certain tasks, called "genetic programming" (GP).

Vocabulary

- Individual - Any possible solution
- Population - Group of all individuals
- Search Space - All possible solutions to the problem
- Chromosome - Blueprint for an individual
- Trait - Possible aspect (features) of an individual
- Allele - Possible settings of trait (black, blond, etc.)
- Locus - The position of a gene on the chromosome
- Genome - Collection of all chromosomes for an individual

Pseudo code - Genetic algorithm


```
{
   initialize population;
   evaluate population;
   while TerminationCriteriaNotSatisfied
   {
      select parents for reproduction;
      perform recombination and mutation;
      evaluate population;
   }
}
```

Quit when you have a satisfactory solution (or you run out of time)

Generate a set of initial solutions.

REPEAT

- Generate new solutions by crossover.
- Mutate the new solutions (optional).
- Evaluate the candidate solutions.
- Retain best candidates and delete the rest.

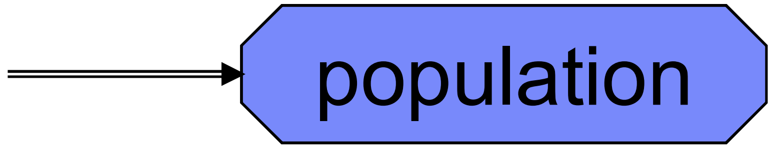
UNTIL stopping criterion met.

Roulette wheel's selection pseudo code

- In roulette wheel selection, individuals are given a probability of being selected that is directly proportionate to their fitness.
- Two individuals are then chosen randomly based on these probabilities and produce offspring.

```
for all members of population
        sum += fitness of this individual
end for
for all members of population
        probability = sum of probabilities + (fitness / sum)
        sum of probabilities += probability
end for
loop until new population is full
        do this twice
                number = Random between 0 and 1
                for all members of population
                        if number > probability but less than next probability then
                        you have been selected
                end for
        end
        create offspring
end loop
```

Population/representation



- Chromosomes could be:
  - Bit strings (0101 ... 1100)
  - 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 ...

Representation example

- We can encode a rule as a binary string, where each bit represents whether a value is accepted.

Make				Tires		Handlebars		Water bottle	
B	C	N	G	K	T	S	C	Y	N

- This rule could be represented as 1100011011:

Make				Tires		Handlebars		Water bottle	
1	1	0	0	0	1	1	0	1	1
B	C	N	G	K	T	S	C	Y	N

*Courtesy: Mark Maloof.

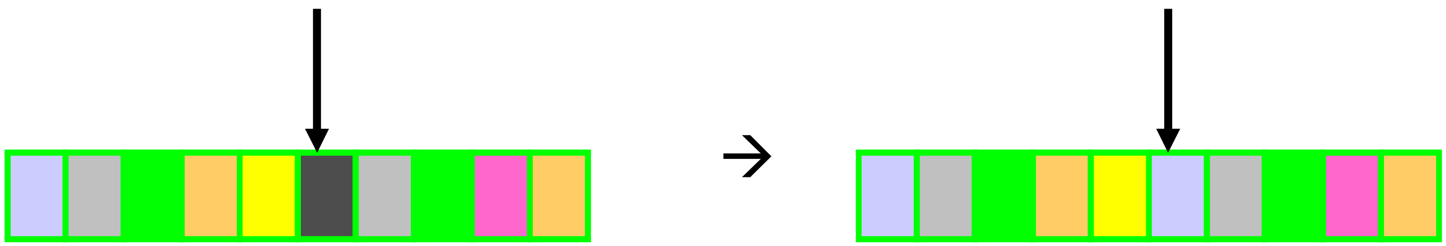
Crossover

- Two parents produce two offspring
- There is a chance that the chromosomes of the two parents are copied unmodified as offspring
- There is a chance that the chromosomes of the two parents are randomly recombined (crossover) to form offspring
- Generating offspring from two selected parents
  - Single point crossover
  - Two point crossover (Multi point crossover)
  - Uniform crossover

Mutation

- Causes movement in the search space (local or global)
- Restores lost information to the population

Generating new offspring from single parent



Before: (1 0 1 1 0 1 1 0)

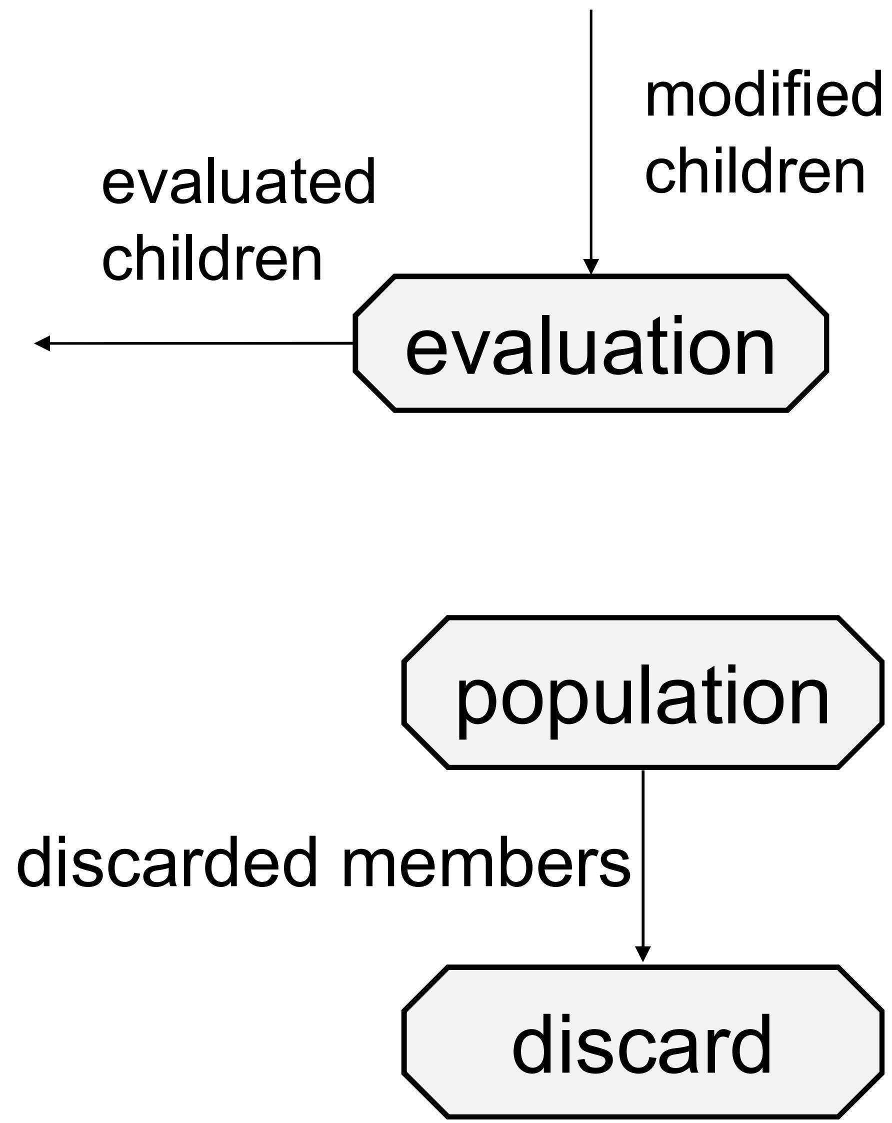
After: (0 1 1 0 0 1 1 0)

Before: (1.38 -69.4 326.44 0.1)

After: (1.38 -67.5 326.44 0.1)

Evaluation and deletion

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



The traveling salesman problem

- Find a tour of a given set of cities so that
  - each city is visited only once
  - the total distance traveled is minimized

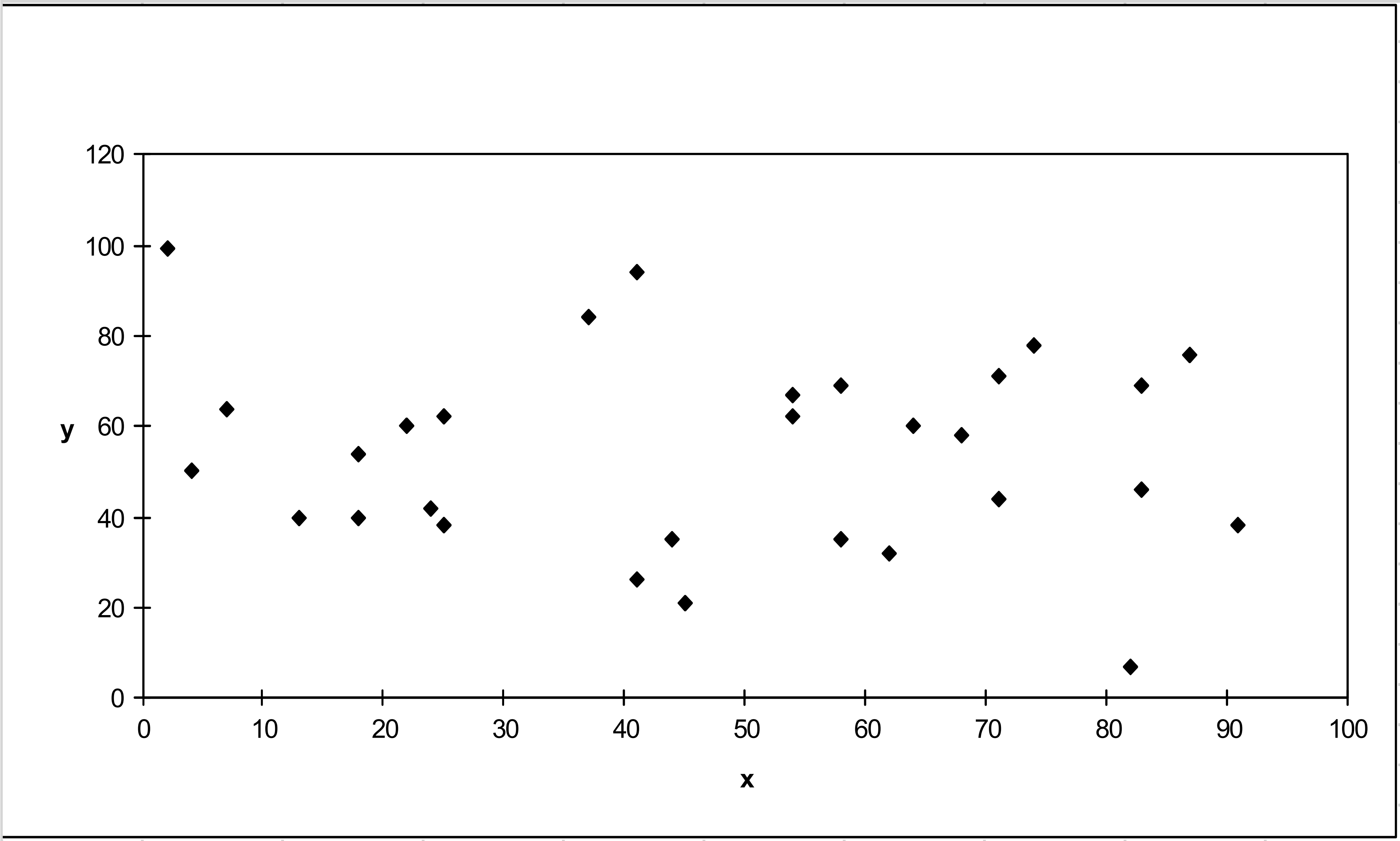
Current tour	Canonic tour	Ordinal representation
$\underline{1}$ 2 5 6 4 3 8 7	$\underline{1}$ 2 3 4 5 6 7 8	1
1 $\underline{2}$ 5 6 4 3 8 7	$\underline{2}$ 3 4 5 6 7 8	1 1
1 2 $\underline{5}$ 6 4 3 8 7	3 4 $\underline{5}$ 6 7 8	1 1 3
1 2 5 $\underline{6}$ 4 3 8 7	3 4 $\underline{6}$ 7 8	1 1 3 3
1 2 5 6 $\underline{4}$ 3 8 7	3 $\underline{4}$ 7 8	1 1 3 3 2
1 2 5 6 4 $\underline{3}$ 8 7	$\underline{3}$ 7 8	1 1 3 3 2 1
1 2 5 6 4 3 $\underline{8}$ 7	7 $\underline{8}$	1 1 3 3 2 1 2
1 2 5 6 4 3 8 $\underline{7}$	$\underline{7}$	1 1 3 3 2 1 2 1

Figure The ordinal representation.

Representation, cross over and mutation

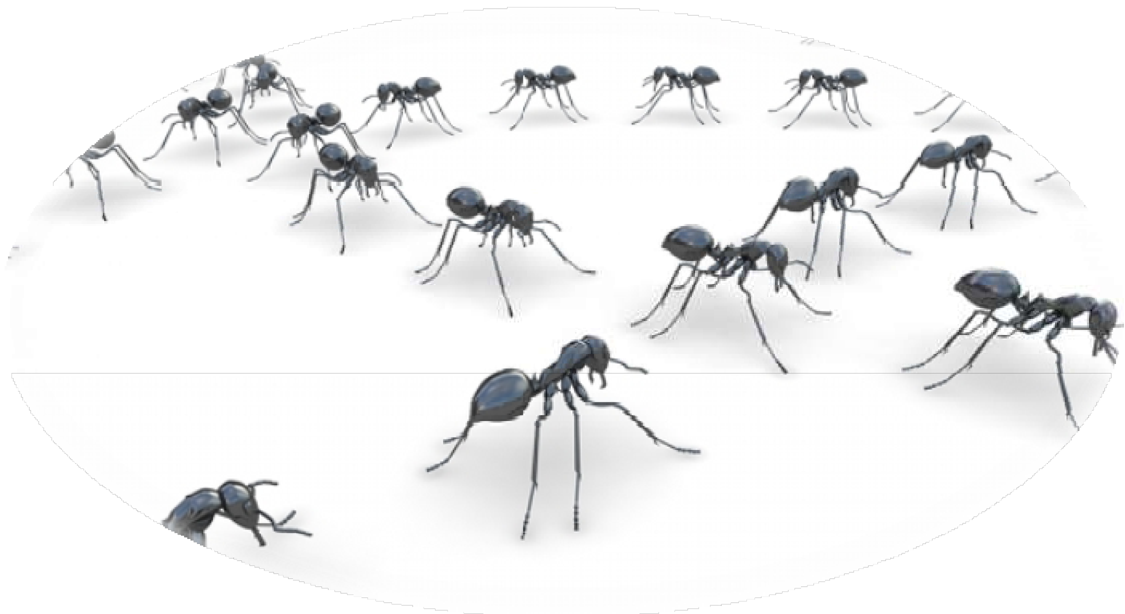
- Representation is an ordered list of city numbers known as an order-based GA.
- Crossover combines inversion and recombination
- Mutation involves reordering of the list

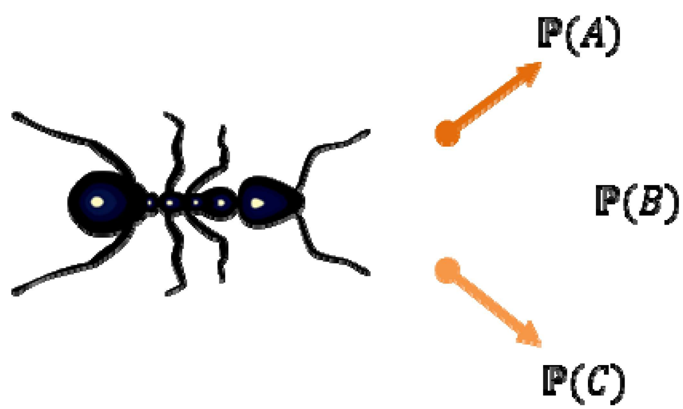
TSP Example - 30 cities



Ant colony and artificial ants for TSP

- Pheromone trails
  - It is well known that the primary means for ants to form and maintain the line is a pheromone trail. Ants deposit a certain amount of pheromone while walking, and each ant probabilistically prefers to follow a direction rich in pheromone.

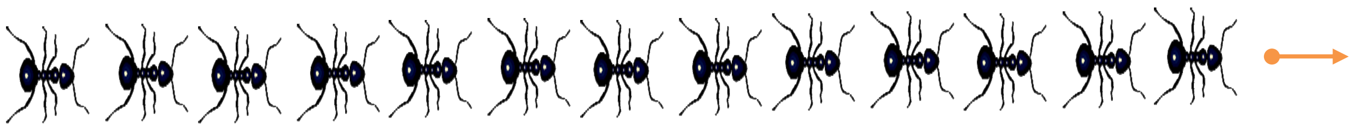




$$\boxed{\mathbb{P}(C) < \mathbb{P}(B) < \mathbb{P}(A)}$$

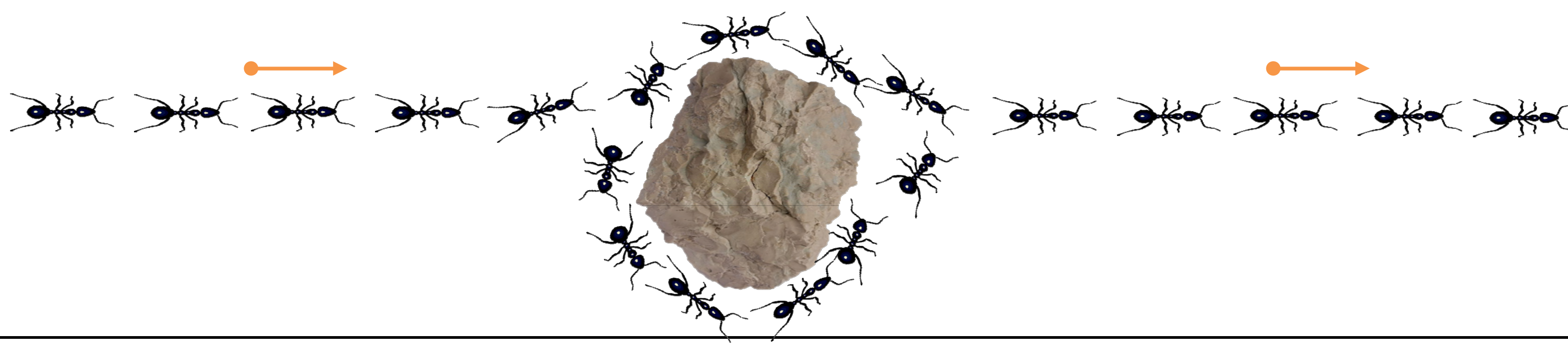
Pheromone trails

- **Shortest path around an obstacle:** This elementary behavior of real ants can be used to explain how they can find the shortest path that reconnects a broken line after the sudden appearance of an unexpected obstacle has interrupted the initial path.
- Let us consider the following scenario:
  - Ants are moving on a straight line that connects a food source to their nest.



  - An obstacle appears on the path.

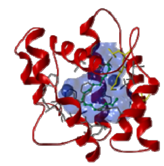
  - Those ants which are just in front of the obstacle cannot continue to follow the pheromone trail and therefore they have to choose between turning right or left. In this situation we can expect half the ants to choose to turn right and the other half to turn left.

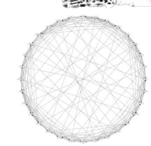


Ant colony optimization algorithms

- Ant Colony Optimization (ACO) studies artificial systems that take inspiration from the behavior of real ant colonies and which are used to solve discrete optimization problems.
- Some Applications:
  - Set partition problem: Deciding whether a given multiset of positive integers can be partitioned into two subsets A and B such that the sum of the numbers in A equals the sum of the numbers in B.
  - Job – Shop problems: Given a number of jobs have to be done and every job consists of using a number of machines for a certain amount of time. Find the best planning to do all the jobs on all the different machines in the shortest period of time.
  - Multiple knapsack problem: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.
  - Protein folding: The process by which a protein structure assumes its functional shape or conformation.
  - Data Mining: The computational process of discovering patterns in large data sets.
  - Travelling salesman problem: Given a list of cities and the distances between each pair of cities, determine the shortest possible route that visits each city exactly once and returns to the origin city.





Particle swarm optimization – Introduction

- Concept first introduced by Kennedy and Eberhart (1995)
- Original idea was to develop a realistic visual simulation of bird flock behaviour
- Simulation was then modified to include a point in the environment that attracted the virtual bird agents
- Potential for optimisation applications then became apparent
- A flock of birds (or school of fish, etc.) searching for food and their objective is to efficiently find the best source of food
- Nature-based theory underlying PSO: The advantage of sharing information within a group outweighs the disadvantage of having to share the reward

Image: http://www.nerjarob.com/nature/wp-content/uploads/Flock-of-pigeons.jpg

Kennedy and Eberhart's (1995) refined algorithm

- Some number of agent particles are initialised with individual positions and velocities (often just done randomly)

new velocity = old velocity + 2 * rand1 * (*pbest* location - current location)
+ 2 * rand2 * (*gbest* location - current location)

where rand1 and rand2 are random numbers between 0 and 1.
(Multiplying by the constant 2 causes particles to "overshoot" their target about half of the time, resulting in further exploration.)

A (partial) example in two dimensions

	0	1	2	3	4	5	6	7	8	9
0	3	2	1							
1	3	1			1					
2	2	1		2	2		1			
3	1			3	4	3	2			
4			2	4	5	3	1		1	1
5			1	3	2	3	2	1	2	1
6		1	3	1	1			1	3	2
7	1	3	2	1				1	2	1
8	1	4	2	1					1	1
9	1	2	1							

<u>pbests</u>

Blue: 0

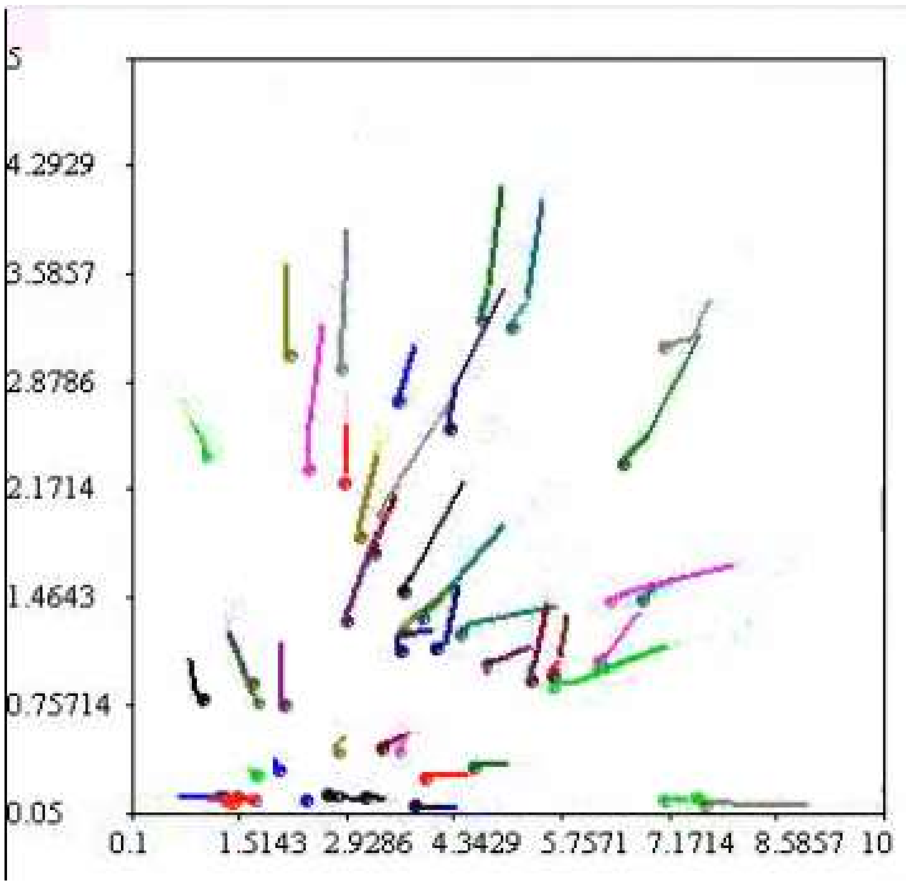
Red: 0

gbest: 0

(dots indicate agents, yellow star indicates the global optimum)

Algorithm termination

Kennedy, J.; Eberhart, R.; , "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE International Conference on* , vol.4, no., pp.1942-1948 vol.4, Nov/Dec 1995.
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=488968&isnumber=10434



Financial applications

- Forecasting of financial distress
- Credit scoring
- Multistage stochastic financial optimization
- Modeling of time series, prediction and forecasting

An automatic stock trading system using Particle Swarm Optimization

- A trading strategy based on a learning method to combine a set of technical trading signals.
- The learning employs a modified Particle Swarm Optimization to optimize the weights of signals.
- The set of weighted signals is then used to determine trading decisions, i.e. to buy, to sell or to hold.
- Results on some financial data is explored for understanding.

PSO based methodology

- The required inputs are daily closing prices, open prices, volumes, high prices and low prices of the stock in consideration. The steps of the algorithm are described below.
  - A chosen technical trading signals are calculated from the inputs for each historical data record.
  - The vectors of trading signals and closing prices are fed into the optimization process.
  - The weighted signals are summed up to determine a trading decision for each day.
  - The return of the investment, as a percentage of final portfolio value per initial portfolio value, is then calculated for performance comparison.

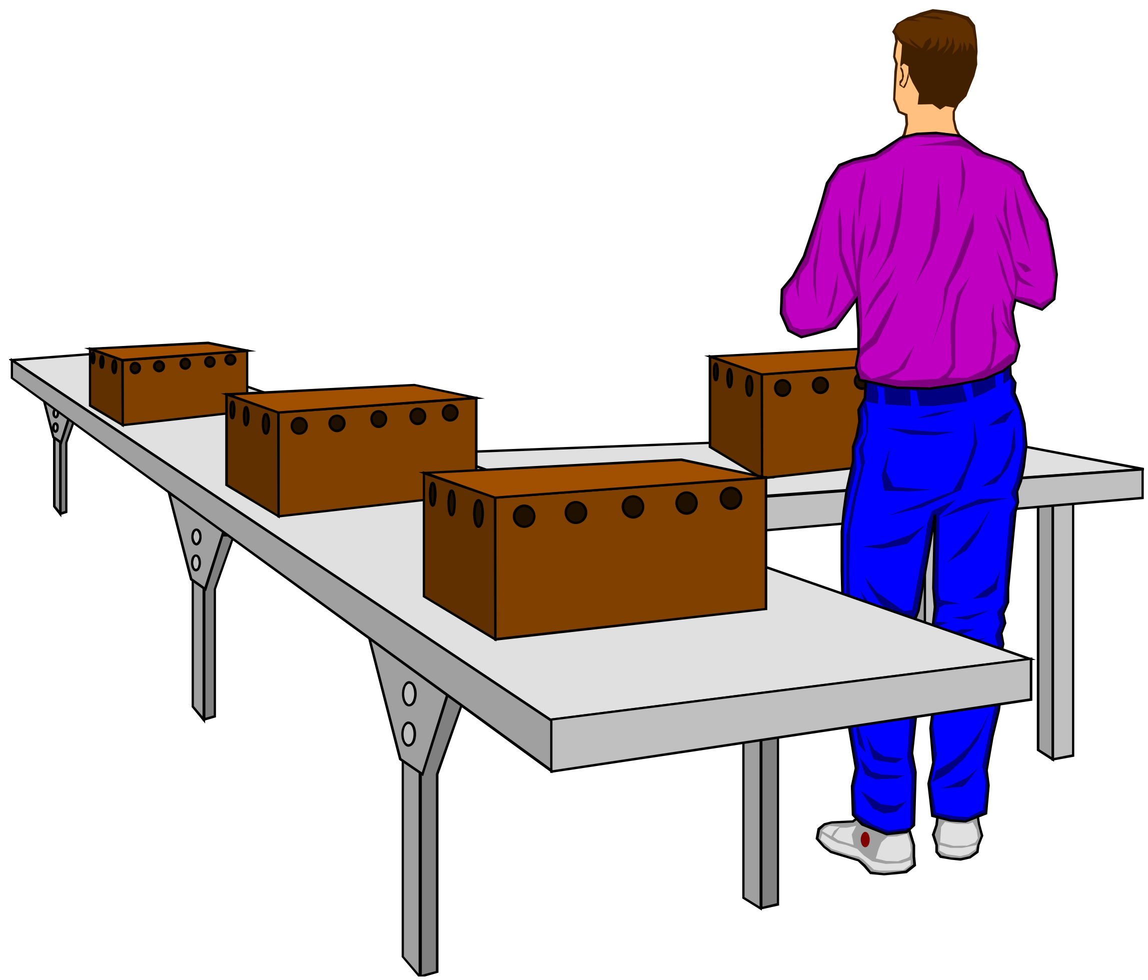
Trading decision

- The $Decision_d$ from equation is used to determine whether to buy, to sell or to hold the particular stock on that day.
- A report returns on investments for stocks in NYSE and SET can be used to know the accuracy of the method.
- A sample result is given here.

Considering the GA *t*echnology

"Almost eight years ago ... people at Microsoft wrote a program [that] uses some genetic things for finding short code sequences. Windows 2.0 and 3.2, NT, and almost all Microsoft applications products have shipped with pieces of code created by that system."

- Nathan Myhrvold, Microsoft Advanced Technology Group, Wired, September 1995



Some GA application types

Domain	Application Types
Control	gas pipeline, pole balancing, missile evasion, pursuit
Design	semiconductor layout, aircraft design, keyboard configuration, communication networks
Scheduling	manufacturing, facility scheduling, resource allocation
Robotics	trajectory planning
Machine Learning	designing neural networks, improving classification algorithms, classifier systems
Signal Processing	filter design
Game Playing	[redacted]
Combinatorial Optimization	set covering, travelling salesman, routing, bin packing, graph colouring and partitioning