Artificial Intelligent Systems

(BMA-EL-IZB-LJ-RE 1. year 2024/2025)

Genetic Algorithms & Evolutionary Machine Learning

Simon Dobrišek

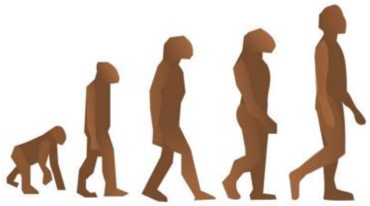
Lecture Topics

- Introduction
- Genetic algorithms
- Chromosome representation
- Examples of the use of genetic algorithm
- Applications of genetic algorithms

Introduction

- In Artificial Intelligence, searching for solutions means searching through the space of possible solutions.
- Solutions or partial solutions are viewed as points in the search space.
- The problems to be solved are normally first formulated as mathematical models expressed in terms of functions.
- Finding solutions then means discovering the parameters that optimize the model or the function components that provide optimal system performance.
- An optimization problem is defined as finding values of the variables that minimize or maximize the objective function while satisfying the constraints.

Introduction



- A genetic algorithm is an AI search heuristic that mimics the process of natural selection.
- Genetic algorithms (GAs) are the main paradigm of evolutionary computing.
- GAs are inspired by Darwin's theory about evolution – the "*survival of the fittest*".
- In nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.
- The basic principles of GAs are selection, reproduction, inheritance, recombination/crossover, and mutation.

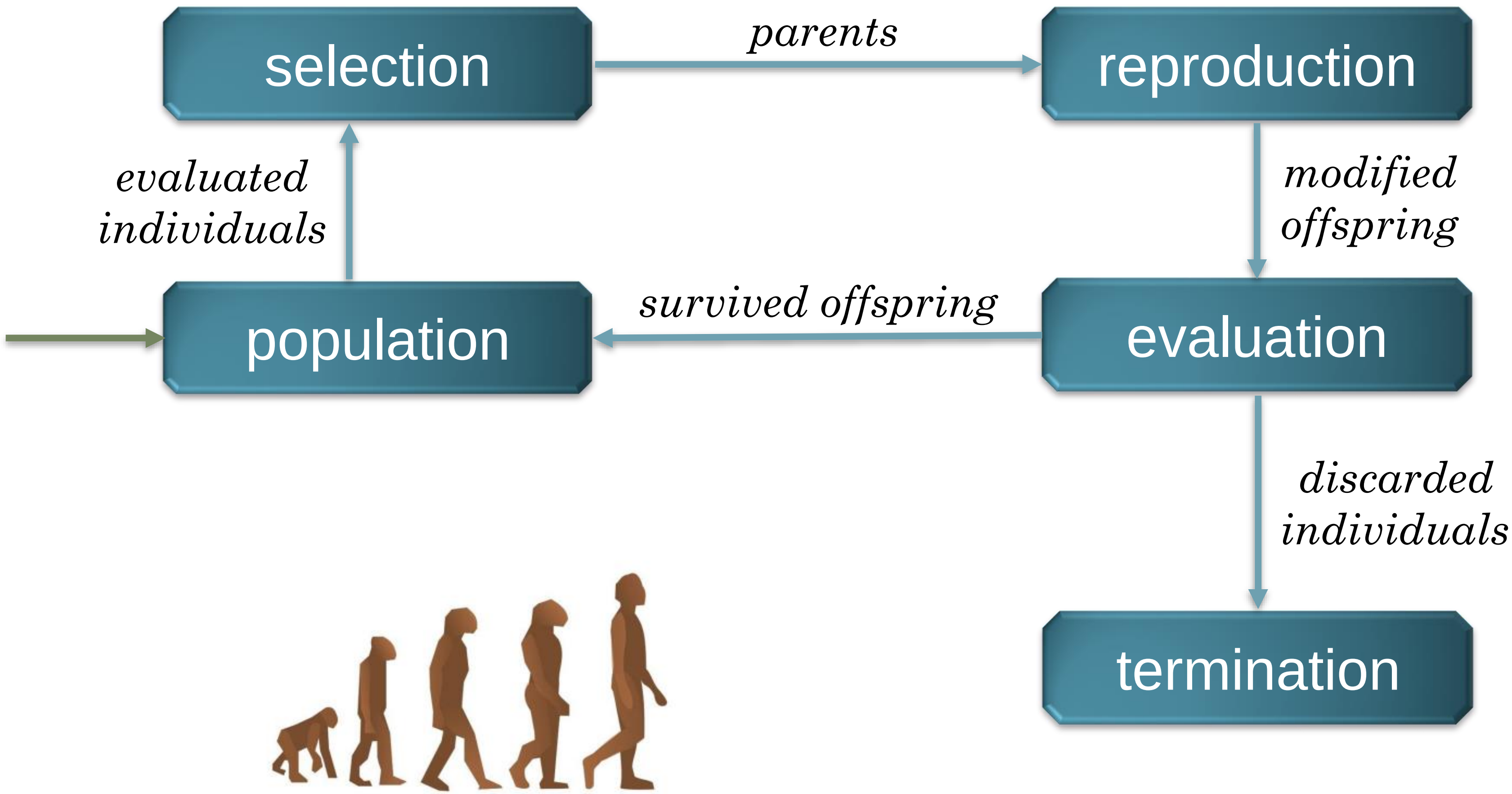
INTRODUCTION

- In a genetic algorithm, a population of candidate solutions (called individuals or phenotypes) to an optimization problem is evolved toward better solutions.
- Each candidate solution has a set of properties (its chromosome or genotype) which can be mutated and altered.
- The evolution usually starts from a population of randomly generated individuals.
- An iterative process then changes the population in each iteration from generation to generation.

Introduction

- In each generation, the fitness of every individual in the population is evaluated.
- The fitness is usually the value of the objective function in the optimization problem being solved.
- The more fit individuals are randomly selected from the current population, and each individual's genome is recombined with other individuals genomes and randomly mutated to form a new generation.
- The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

The Evolutionary Cycle



THE EVOLUTIONARY CYCLE PSEUDO CODE

BEGIN

INITIALISE population with random candidate solution.

EVALUATE each candidate;

REPEAT UNTIL termination condition is satisfied DO

1. SELECT parents;
2. RECOMBINE pairs of parents;
3. MUTATE the resulting offspring;
4. EVALUATE each candidate;
5. SELECT individuals for the next generation;

END.

GENETIC ALGORITHMS

- A typical genetic algorithm requires:
  - a genetic representation of the solution domain,
  - a fitness function to evaluate the solution domain.
- A standard representation of each candidate solution is as an array of bits of fixed size.
- Such genetic representations are convenient, as their parts are easily aligned due to their fixed size.
- A fitness function is defined by an "environment" in which possible solutions "live" and reproduce.

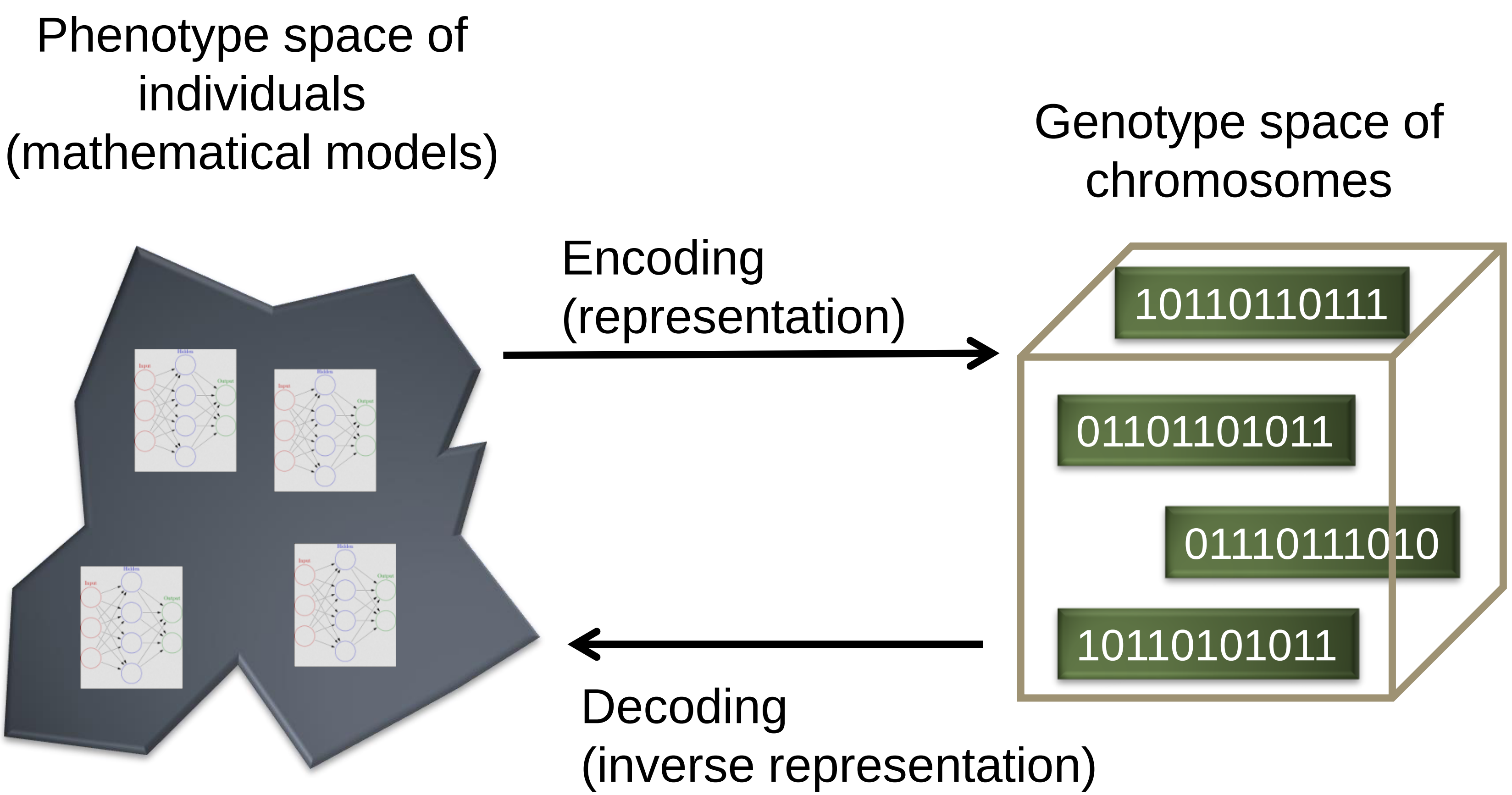
Outline of the Basic Genetic Algorithm

1. [Start] Generate a random population of **n** chromosomes
2. [Fitness] Evaluate the fitness **f(x)** of each chromosome **x** in the population.
3. [New population] Create a new population by repeating following steps until the new population is complete.
   (a) [Selection] Select two parent chromosomes from a population according to their fitness (better the fitness, higher the probability to be chosen)
   (b) [Crossover] With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
   (c) [Mutation] With a mutation probability, mutate new offspring at each locus (position in chromosome).
   (d) [Selection] Place new offspring in the new population
4. [Replace] Use new generated population for a further run of the algorithm
5. [Test] If the end condition is satisfied, stop, and return the best solution in current population
6. [Loop] Go to step 2

Searching for optimal mathematical models and functions

- Individuals in a population are mathematical models or functions represented by the values of their parameters to be optimized (e.g. neural networks, decision trees, etc).
- The genetic representations of the individuals are derived from the parametric variables of the models or functions to be optimized.
- The fitness function are derived from the objective function that is defined for the original optimization problem
- The evolutionary cycle then iteratively reproduces the population of the individuals (models) that achieve the highest values of the fitness (objective) function.

Chromosome Representation



Chromosome Representation

- Many different genetic representations (genotypes) of individual's properties (phenotypes) have been proposed and used.
- Due to its relative simplicity, binary encoding is the most common representation of individual's information that is often used.
- However, binary encoding is not natural for many problems and sometimes corrections must be made after crossover and/or mutation, or more complicated crossover and mutation rules need to be used
- There are many other ways of encoding, e.g., encoding values as integer or real numbers or some other data structures etc.
- The use of a particular encoding method depends on the problem to be solved.

Chromosome Representation

- Binary strings can represent strings of characters, integers or quantized real values.

[00101010|10010101|11101101|00101001|10011110]

- Nowadays it is generally accepted that it is better to encode numerical variables directly as integers

[42535126|09767283|86192876|76243546|67718231]

- Or even directly as floating point values.

[1.311; 3.988; 9.123|4.877; 6.712; 0.981|0.123; 9.712; 7.100]

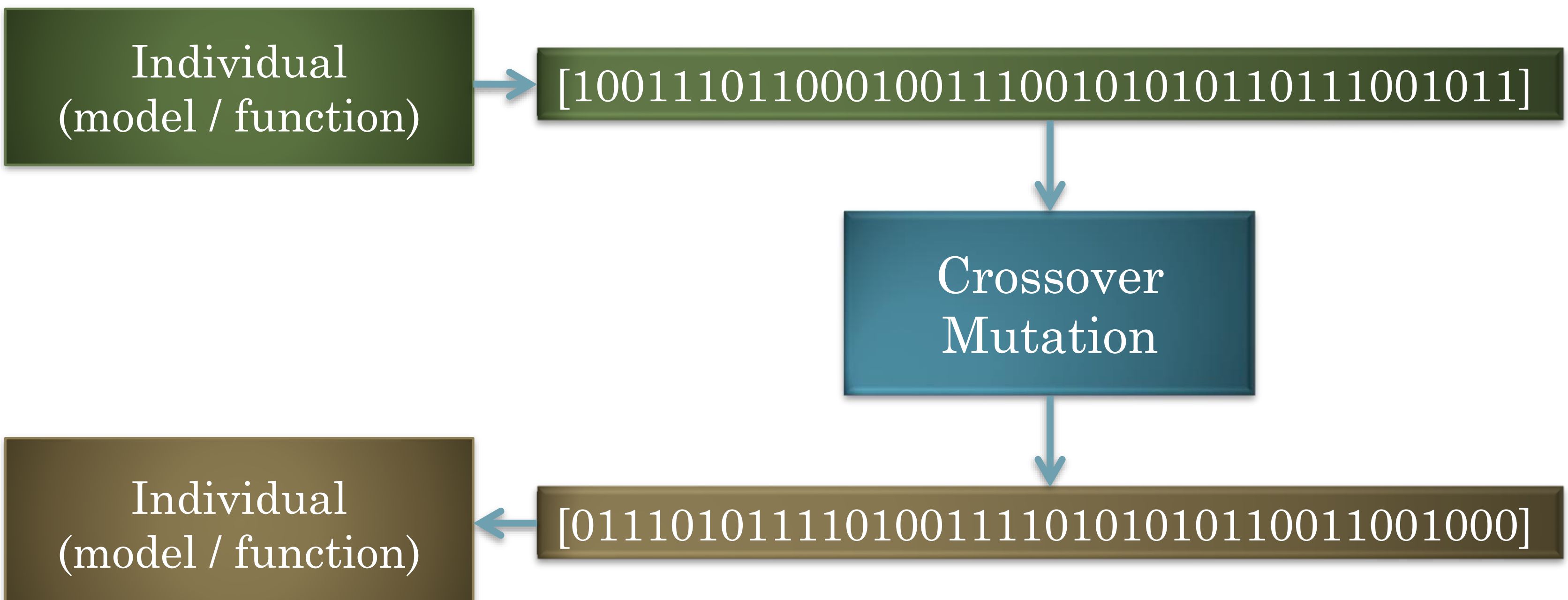
Chromosome Representation

- A critical step for the success of a genetic algorithm is the use of an appropriate encoding of the parameters of the individuals into their genetic representations (chromosomes).
- Any modification of a chromosome should represent valid set of parameter values that are within their range.
- The above should be considered when the crossover and mutation operations are defined as well.
- Gray coding is an example of an encoding that is often used, as it ensures that small changes in the genotype cause small changes in the phenotype and vice-verse (unlike binary coding).

Dec.	Bin.	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

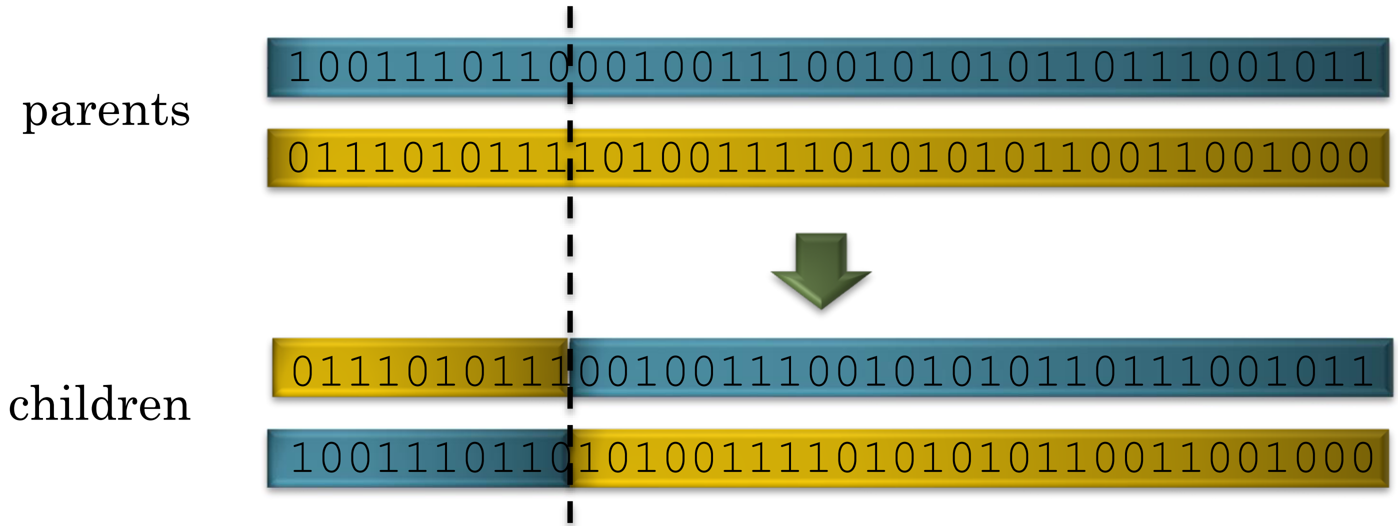
REPRODUCTION

- The crossover and mutation operations are used to modify individuals from generation to generation.
- As mentioned, any modification of a chromosome should represent a valid set of parameter values that are within their range.



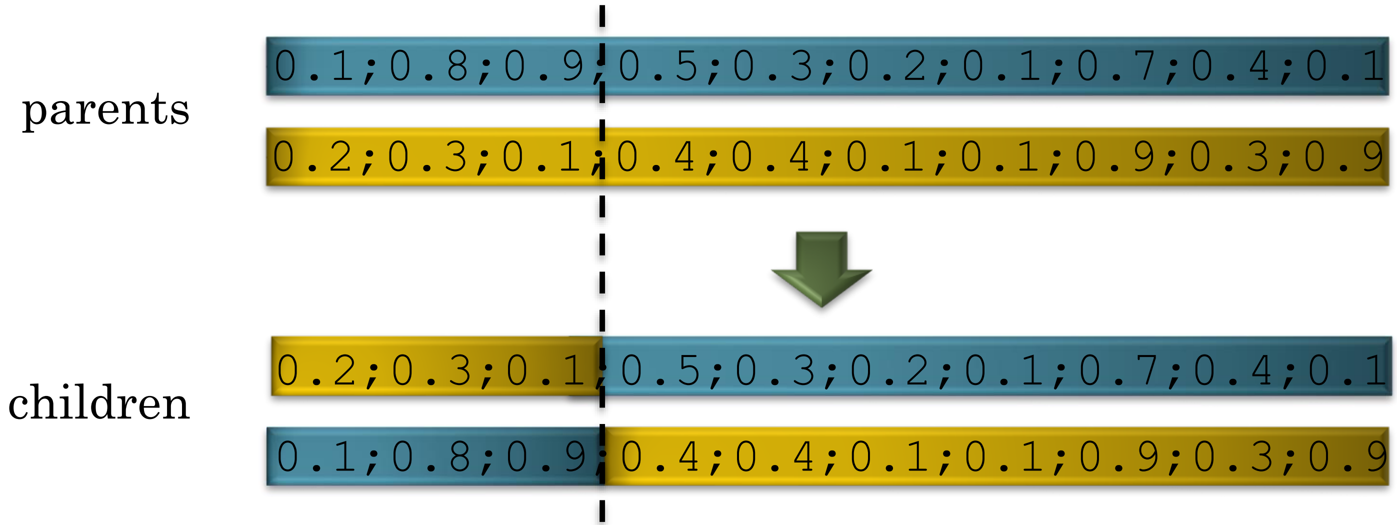
Crossover Operation

- Crossover is a genetic operator used to vary chromosomes from one generation to the next.
- Crossover is a process of taking more than one parent solutions and producing a child solution from them.
- The chromosomes of two randomly chosen individuals (parents) are crossover to create one or more children, where 1-point or *n*-point crossover methods are used.



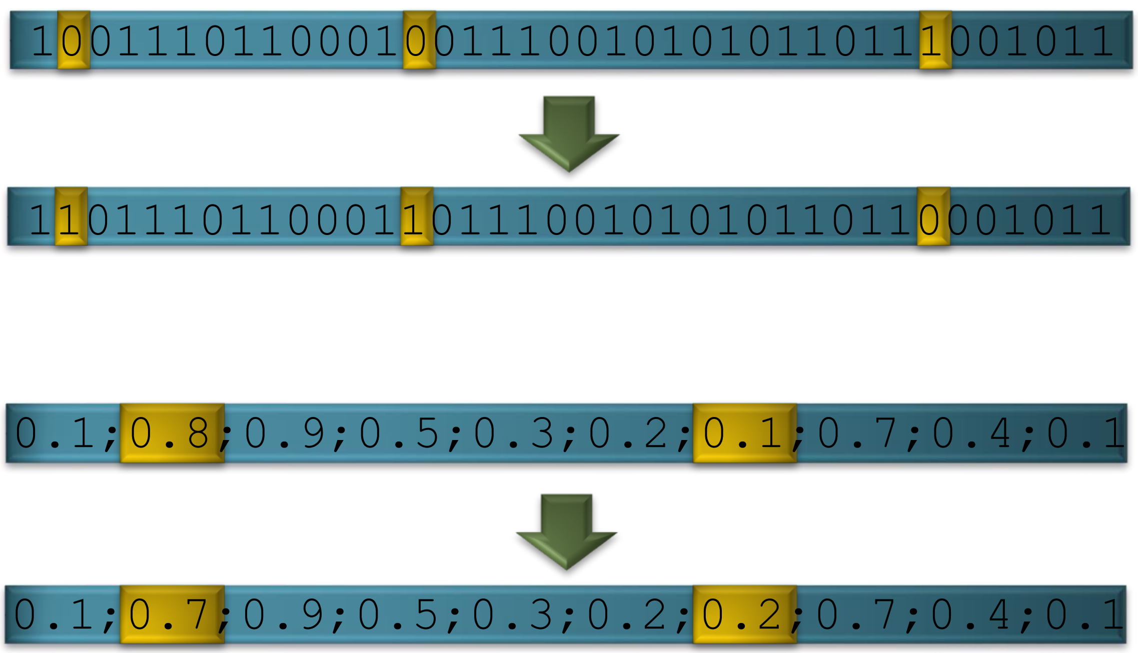
Crossover Operation

- For integer or real values encoded chromosomes different arithmetic crossover operations are used, such as single and simple arithmetic crossover



Mutation Operation

- With a low probability, a few genes are randomly altered in a chromosome.
- Binary, integer or real value mutations are used depending how chromosomes are encoded

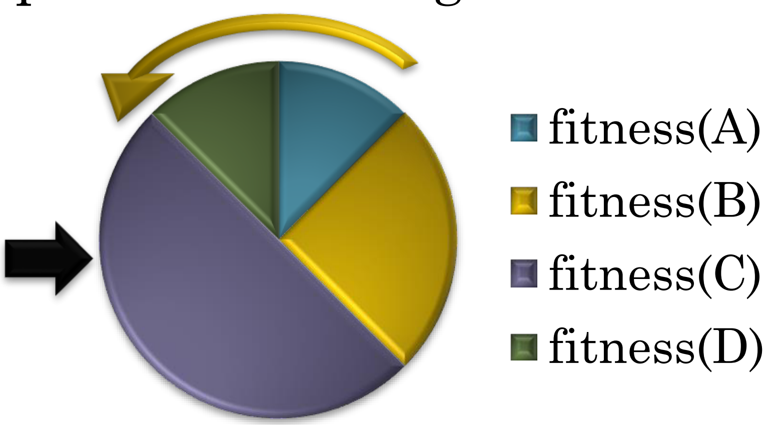


Fitness Function

- In general, a fitness function is a particular type of objective function that is used to summarise how close a given solution is to achieving the optimisation aims.
- The fitness used in GAs is usually the value of the objective function in the original optimization problem being solved.
- Even though the computer should come up with the final solution/design, it is the human designer who has to design the fitness function.
- If the fitness function is designed badly, the algorithm will either converge on an inappropriate solution, or will have difficulty converging at all.

Fitness Function and Selection

- The values of function is normalized to the sum of its values throughout the population.
- The normalized values are then considered as the probabilities of reproduction of each individual represented by their chromosomes.
- A higher value of the normalised fitness functions means a higher probability of reproduction of the individual.
- A random selection is then implemented using the roulette wheel technique.
- Assign to each individual a part of the roulette wheel and spin the wheel n times to select n individuals.



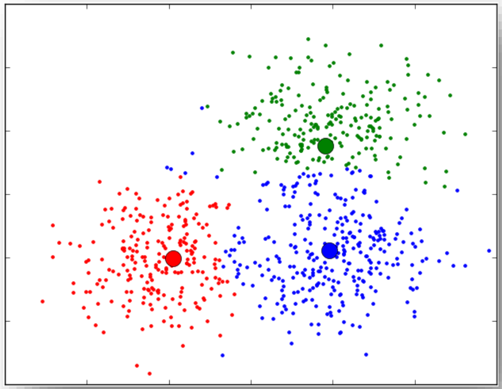
EXAMPLES

- k-means clustering is a classical datamining algorithm that aims at partitioning n observations into k clusters in which each observation belongs to the cluster with the nearest mean observation, serving as a prototype of the cluster.

$$[\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_m] = [w_{11}; w_{12}; \dots; w_{1l} | w_{21}; w_{22}; \dots; w_{2l} | \dots | w_{m1}; w_{m2}; \dots; w_{ml}]$$

$$J([\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_m]) = \sum_{i=1}^{n} \sum_{j=1}^{m} \mu_{ij} d(\boldsymbol{x}_i, \boldsymbol{w}_j)$$

$$\mu_{ij} = \begin{cases} 1 \ , & d(\boldsymbol{x}_i, \boldsymbol{w}_j) = \min\limits_{k=1,\dots,m} d(\boldsymbol{x}_i, \boldsymbol{w}_k) \\ 0 \ , & \text{otherwise} \end{cases}$$



- Chromosomes are encoded as sequences of prototype vectors values.
- Crossover is allowed only on contacts between the vectors in the chromosomes.
- Mutations are small random changes of the vector values.

SEARCHING FOR A MAXIMUM OF A MULTIVARIATE FUNCTION

- Searching for a maximum of a function $f_1(x, y)$ that is constrained with a function $f_2(x, y)$.

- The domain of the both function is the same

$$(x, y) \in [0,10) \times [0,10)$$

- Chromosomes are coded as an eight-digit decimal code

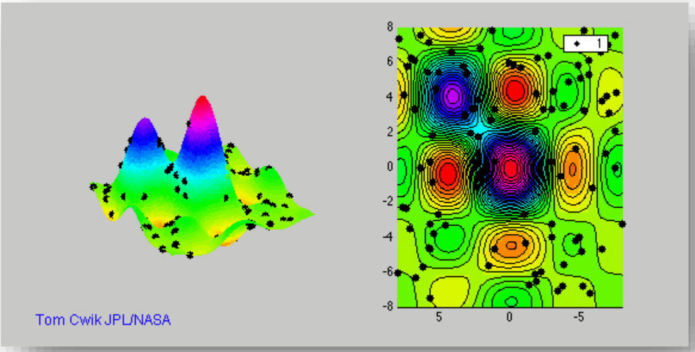
$$(x = 7{,}623\ , y = 3{,}098) \longrightarrow [76233098]$$

- Crossover is carried out with a cut at one random point

[76233098]
[53094145]

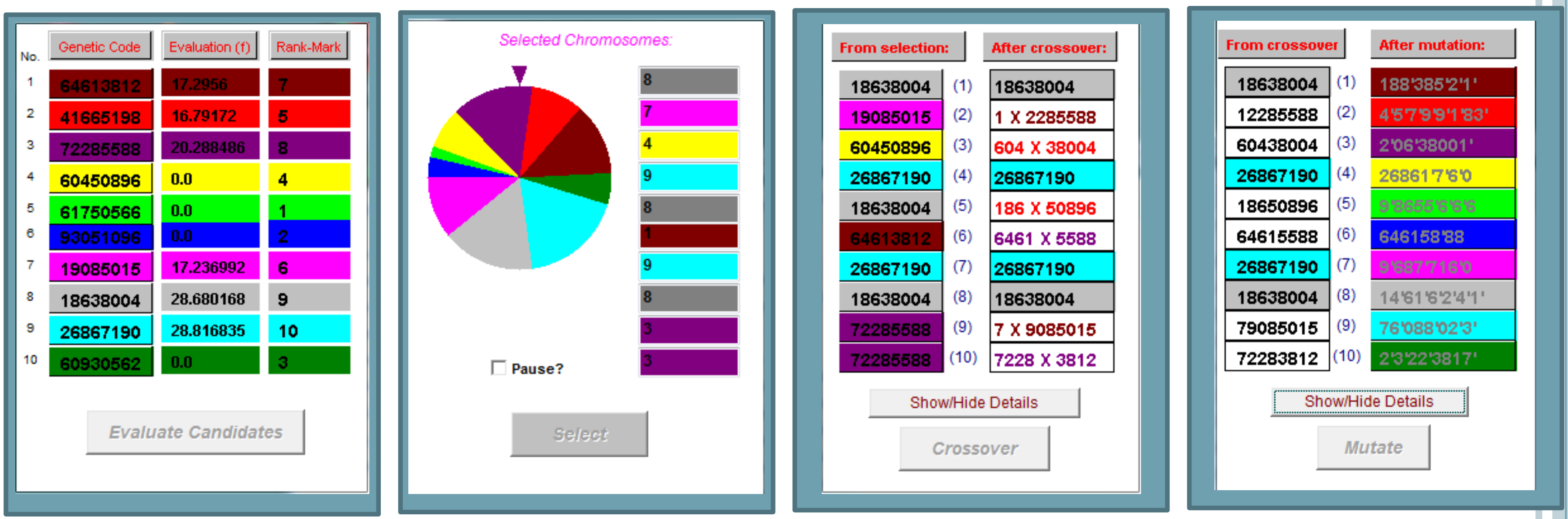
→

[76233145]
[53094098]



https://engineering.purdue.edu/gekcogrp/methodology/GENES/genes_img/2sinc_opt_title.gif

SEARCHING FOR A MAXIMUM OF A MULTIVARIATE FUNCTION



(c) Yun Li & Sylvain Marquios, University of Glasgow

Advantages of Genetic Algorithms

- Genetic algorithms are usually simple and are easy to be implemented.
- They can be parallelized with a little effort.
- Objective functions that are not smooth, are noisy and their derivatives does not exists can still be used as the fitness function for a genetic algorithm.
- Genetic algorithms can also handle with the stochastic nature of objective functions.
- They are flexible to hybridize with other techniques.

Limitations of Genetic Algorithms

- Repeated fitness function evaluation for complex problems is often the most prohibitive and limiting segment of genetic algorithms.
- Where the number of elements which are exposed to mutation is large there is often an exponential increase in search space size.
- The "better" solution is only in comparison to other solutions, and the stop criterion is not clear in every problem.
- In many problems, GAs may have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem.
- GAs cannot effectively solve problems in which the only fitness measure is a single right/wrong measure.
- For specific optimization problems, other optimization algorithms may be more efficient in terms of speed of convergence.

Applications of genetic algorithms

- Automated design
- Bioinformatics - RNA structure prediction.
- Code-breaking
- Control engineering
- Learning robot behavior
- Multimodal Optimization
- Timetabling problems,
- Training artificial neural networks
- Traveling salesman problem and its applications.
- …

Related Algorithms

- Swarm intelligence
  (candidate solutions moves in the search space)
- Ant colony optimization
- Particle swarm optimization
- Intelligent Water Drops
  (modifying the amount of soil on the river's bed)
- Simulated annealing
- Tabu search
- Harmony search
- Memetic algorithm
  (the idea comes from memes, which unlike genes, can adapt themselves.)
- Cultural algorithm
- …

QUESTIONS

- What is the aim of genetic algorithms?
- What are the main operators of generic algorithms?
- How potential solutions of a problem are encoded into chromosomes?
- What are the advantages and limitations of genetic algorithms?
- Give an example of optimization with a genetic algorithm.