ISLab

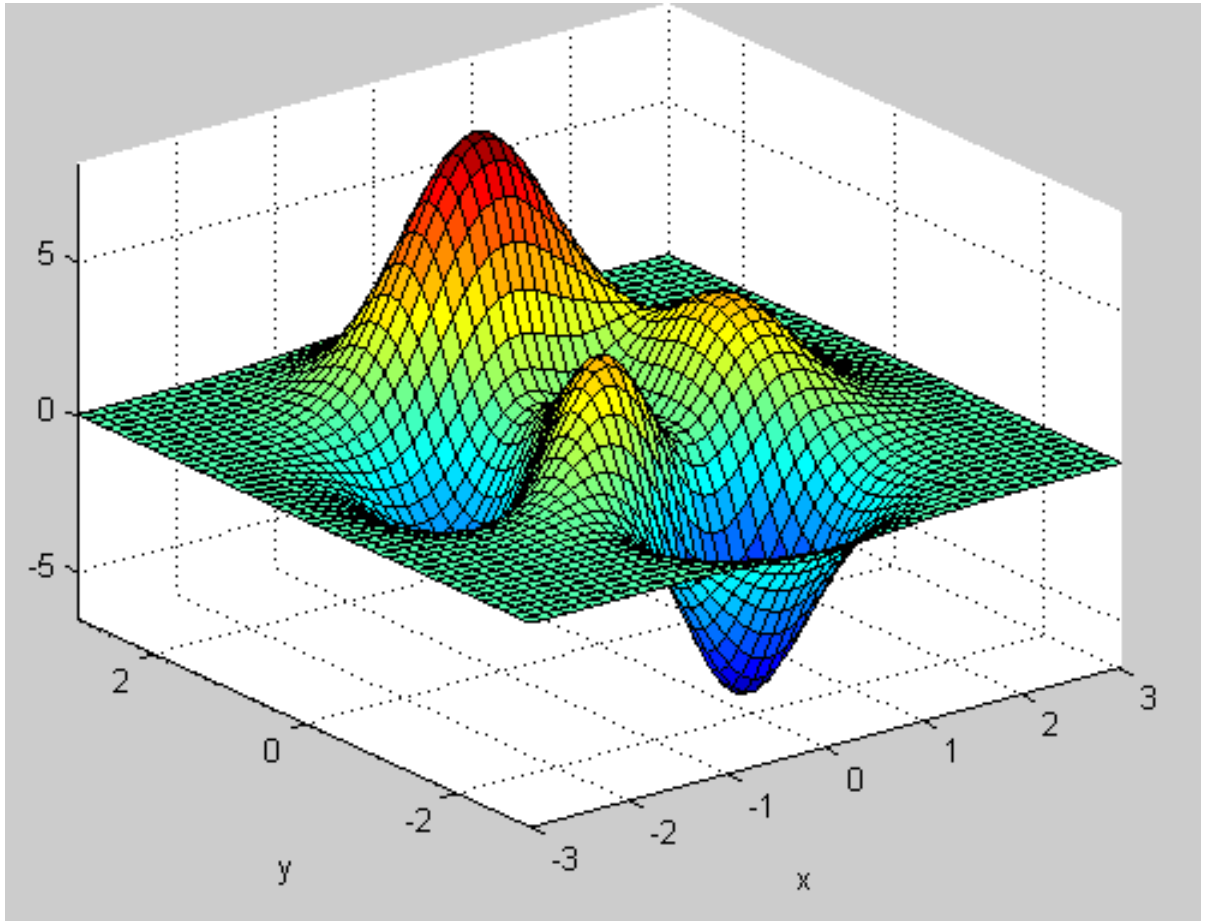
Universidade do Minho
Escola de Engenharia
Departamento de Informática

Genetic and Evolutionary Algorithms

Mestrado Integrado em Engenharia Informática
Mestrado em Engenharia Informática
Perfil SI :: Computação Natural

ISLab
Synthetic Intelligence Lab

Optimization and Search



Tjalling C. Koopmans (Nobel Memorial Prize in Economic Sciences (1975)):
– “best use of scarce resources”
– “Mathematical Methods of Organizing and Planning of Production”

Roger Fletcher (Mathematician (1987)):
”The subject of optimization is a fascinating blend of heuristics and rigour, of theory and experiment.”

ISLab
Synthetic Intelligence Lab

Optimization

Many real-world problems involve maximizing or minimizing a value:

- How can a car manufacturer get the most parts out of a piece of sheet metal?
- How can a moving company fit / transport the maximum majority of furniture in a truck of a given size?
- How can a telephone company route calls to get the best use of its lines and connections?
- How can a university schedule its classes to make the best use of classrooms without conflict?

ISLab
Synthetic Intelligence Lab

Optimization and Search Methods

- Brute-force search (Exhaustive search);
- Uninformed Search;
- Heuristic Search;
- Hill climbing;
- Gradient ascent;
- Simulated annealing;
- …

Optimization

Basic concepts:

- A numerical representation of x for all possible solutions to a given problem;
- $f(x)$, as a function that tells us how good a solution to this problem is;

- The possibility of finding
  - $\max_x f(x)\ if\ bigger\ f(x) is\ better\ \ (benefit)$

  - $\min_x f(x)\ if\ smaller\ f(x) is\ better\ \ (cost)$

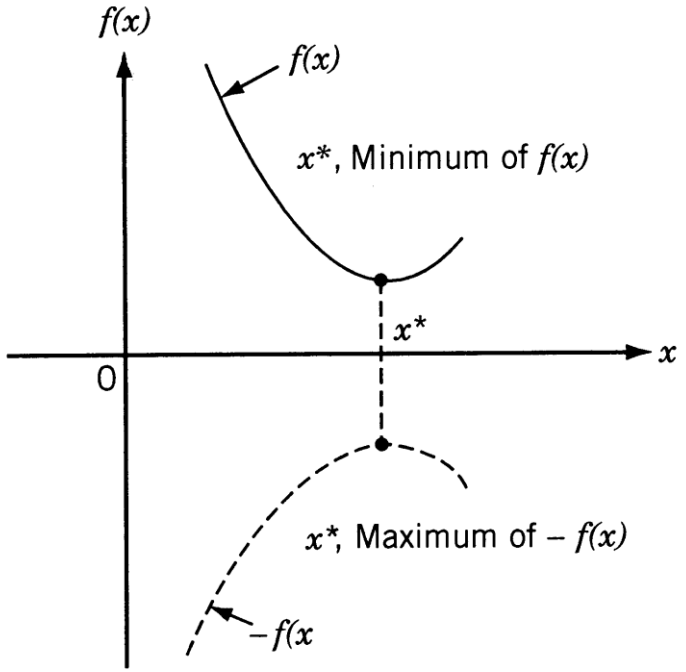
ISLab
Synthetic Intelligence Lab

Optimization Problems

- A numerical representation of X for all possible solutions to the problem
  $x = (x_1, \ldots, x_n)$
  Controllable variables (linearly independent)

- $f_0 : R^n \rightarrow R$
  Objective function

- $g_i : R^n \rightarrow R: (I = 1,\ldots,m)$
  Restrictions



Optimization and Search

Continuous Optimization

“finding the maxima and minima of functions, possibly subject to constraints”

“In **continuous optimization**, the variables in the model are allowed to take on any value within a range of values, usually real numbers. ...”

Discrete Optimization

“looking thoroughly in order to find an item with specified properties among a collection of items”

“As opposed to continuous optimization, the variables used (or some of them) are restricted to assume only discrete values, such as the integers.”

ISLab
Synthetic Intelligence Lab

The famous one!

Image Source: https://www.linkedin.com/pulse/applying-traveling-salesman-problem-business-russ-penlington/

Travelling Salesman Problem

Given the coordinates of n cities, find the shortest closed tour which visits each once and only once.

ISLab
Synthetic Intelligence Lab

Optimization and Search

- In many optimization problems, the path to the goal is irrelevant! **The goal is the solution itself**!
- State space = set of complete configurations!
- Iterative Algorithms maintain a single state (current) or a population of states and try to improve it!
- Iterative Improvement Algorithms:
  - Hill-Climbing Search
  - Simulated Annealing
  - Tabu Search
- Solution Population:
  - Genetic Algorithms
  - Particle Swarm Optimization
  - Ant Colony Optimization
- Strategy:

  Start as an initial solution / population of initial solutions to the problem and make changes in order to improve its quality.

Optimization and Search

Individual Based

Few examples:

- **Hill-Climbing Search**
  - Choose a state randomly from the state space
  - Consider all neighbors in that state
  - Choosing the best neighbor
  - Repeat the process until there are no better neighbors
  - The current state is the solution
- **Simulated Annealing**
  - Similar to Hill-Climbing Search but admits to exploring worse neighbors
  - Temperature that is successively reduced defines the probability of accepting worse solutions
- **Tabu Search**
  - Similar to Hill-Climbing Search, it explores neighboring states but eliminates the worst (taboo neighbors)
  - Deterministic algorithm

ISLab
Synthetic Intelligence Lab

Optimization and Search

Population Based

Few examples:

- **Particle Swarm Optimization**
  - Various departure states (swarm)
  - The neighborhood is explored and kept, the best solution and the best state
  - States are moving in the direction of the best solution found so far
  - The speed of movement depends on the distances to the best solution and the best state and the state position
- **Ant Colony Optimization**
  - Starting several states (ants colony)
  - The probability of a path being better is determined from the number of “ants” that pass through it
- **Genetic Algorithms**
  - Definition of the state as a chromosome
  - Generate solutions (chromosomes) from an initial state population
  - Reproduction, Mutation and Selection

ISLab
Synthetic Intelligence Lab

Genetic and Evolutionary Algorithms

- An **iterative** procedure that maintains a **population** of structures that are **candidates for solutions**, for specific domains.
- With each increment of time (**generation**), the structures of the current population are evaluated in terms of their ability to be valid solutions for the problem domain, forming a **new population** of candidate solutions, based on their evaluation, developed by the application of **genetic operators** (selection, crossover, mutation, purification, among others).

ISLab
Synthetic Intelligence Lab

Genetic and Evolutionary Algorithms

- Genetic Algorithms configure adaptive search processes in a space of solutions, by applying operators modeled according to the concept of inheritance, inherent to the theory of the evolution of species, by Charles Darwin;
- Belong to the class of probabilistic algorithms, distinguishing:
  - the search method they use;
  - for the specific treatment of the great places;
- Application areas:
  - in problems that involve the **improvement of solutions** (optimization problems);
  - problems whose **calculation** of solutions is **difficult or** even **impossible**.

Adaptation in Natural and Artificial Systems

- In the early 1970s, John Holland developed research aimed at verifying whether aspects of natural evolution could be incorporated into algorithms that allow the automatic resolution of problems;
- According to John Holland, each solution to a problem could be understood as an individual in a population, so that all individuals in the population would be equivalent to a set of possible solutions to a problem, each individual would be represented by its genotype, that is, individuals would be represented by chromosomes;
- Such populations would generally evolve based on the reproduction process, assuming that mutations could occasionally occur;
- In this way the population was renewed through generations, which would correspond to cycles of the algorithm:
- The process was repeated until we arrived at an individual (solution) that had the minimum desired characteristics or until a time limit imposed the end of the algorithm.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. University of Michigan Press, Ann Arbor, MI, USA.

ISLab
Synthetic Intelligence Lab

Basic ideas

- Search in some search spaces using traditional search methods would be intractable;
- This usually occurs when candidate states/solutions have a very large number of successors;
- Genetic algorithms are a random heuristic search strategy;
- Basic idea: Simulate natural selection, where the population is made up of candidate solutions;
- The focus is on evolving a population from which strong and diverse candidates can emerge through mutation and crossover (reproduction).

Genetic Algorithm

Genetic Algorithm is a stochastic process in which successor states are generated by combining two parent states instead of modifying a single state.

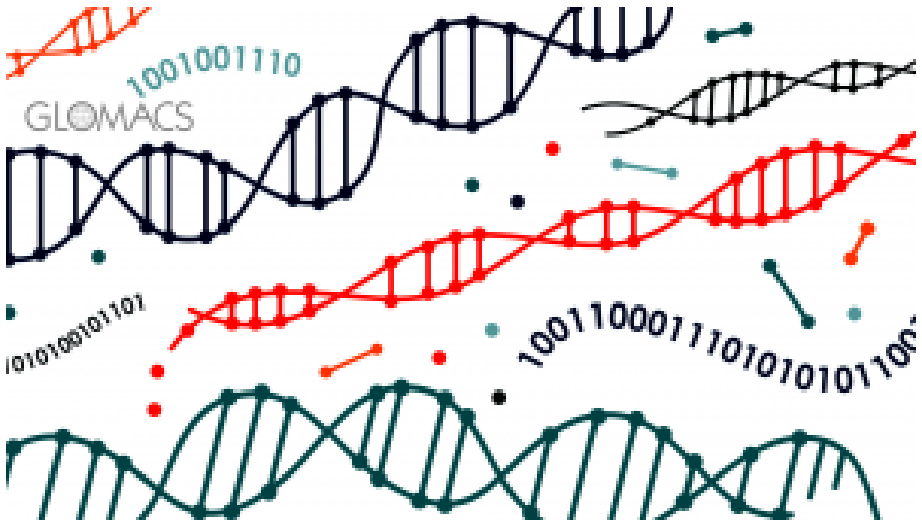


Image Source: http://glomacs.com/articles/genetic-algorithms

ISLab
Synthetic Intelligence Lab

Pseudo-Code

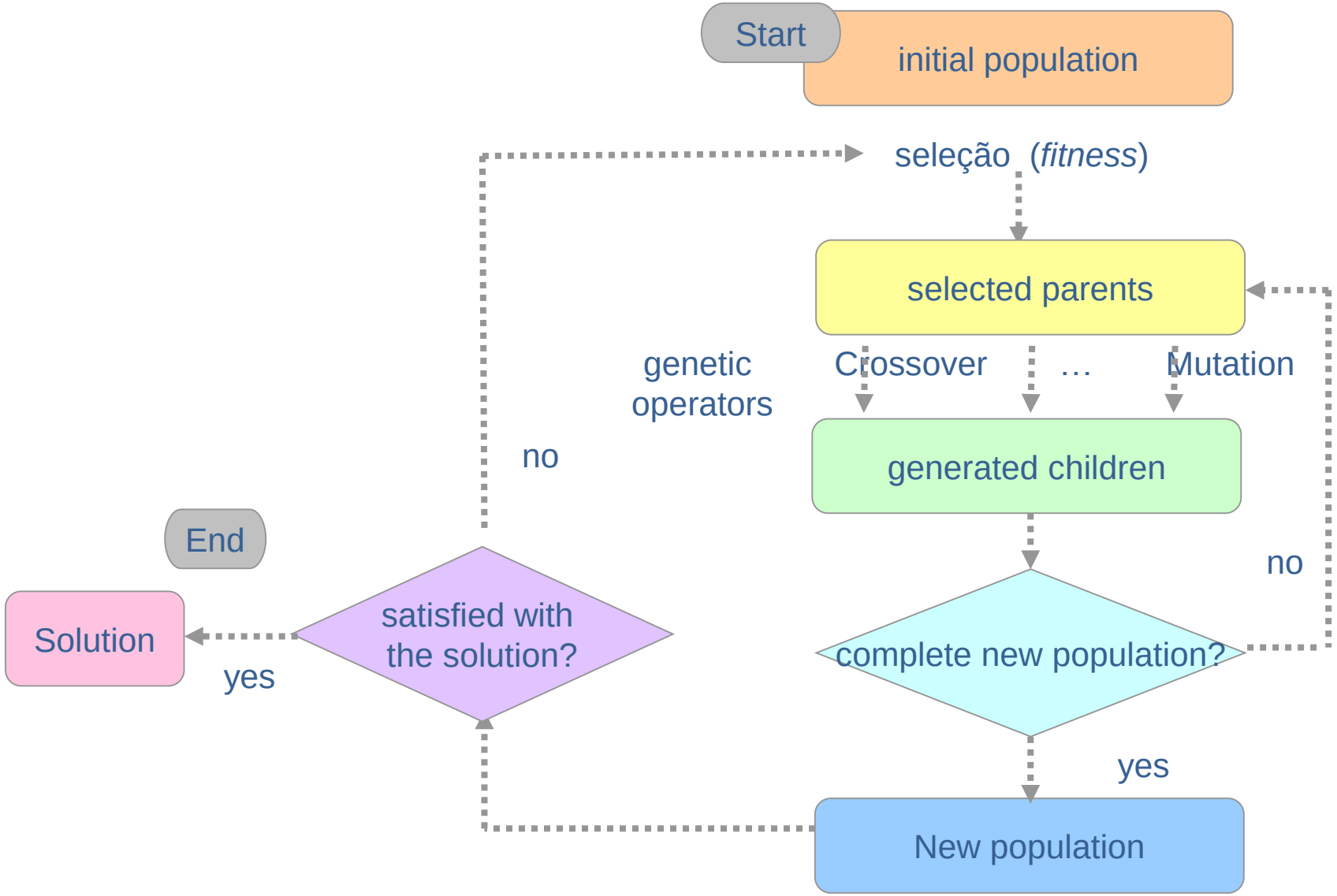
- Choose initial population (e.g., random)
- Repeat until “terminated”
  - Evaluate each individual's fitness
  - Prune population (typically all; if not, then the worst)
  - Select pairs to mate from best-ranked individuals
  - Replenish population (using selected pairs)
    - Apply crossover operator
    - Apply mutation operator
  - Check for “termination criteria”

(number of generations, amount of time, minimum fitness threshold satisfied, fitness has reached a plateau, other)

    - Loop, if not terminating

ISLab
Synthetic Intelligence Lab

Genetic Algorithm



ISLab
Synthetic Intelligence Lab

Basic Components

- Representation of solutions
  - Important to choose this representation well
  - More work here means less work on reproduction roles
- Fitness Function
- Reproduction functions:
  - Mutation
  - Crossover
- Solution Testing
- Some parameters:
  - Population Size
  - Generations limit
  - Selection Method
  - Intersection Method
  - Mutation Method
  - Elitism

ISLab
Synthetic Intelligence Lab

Representation Model

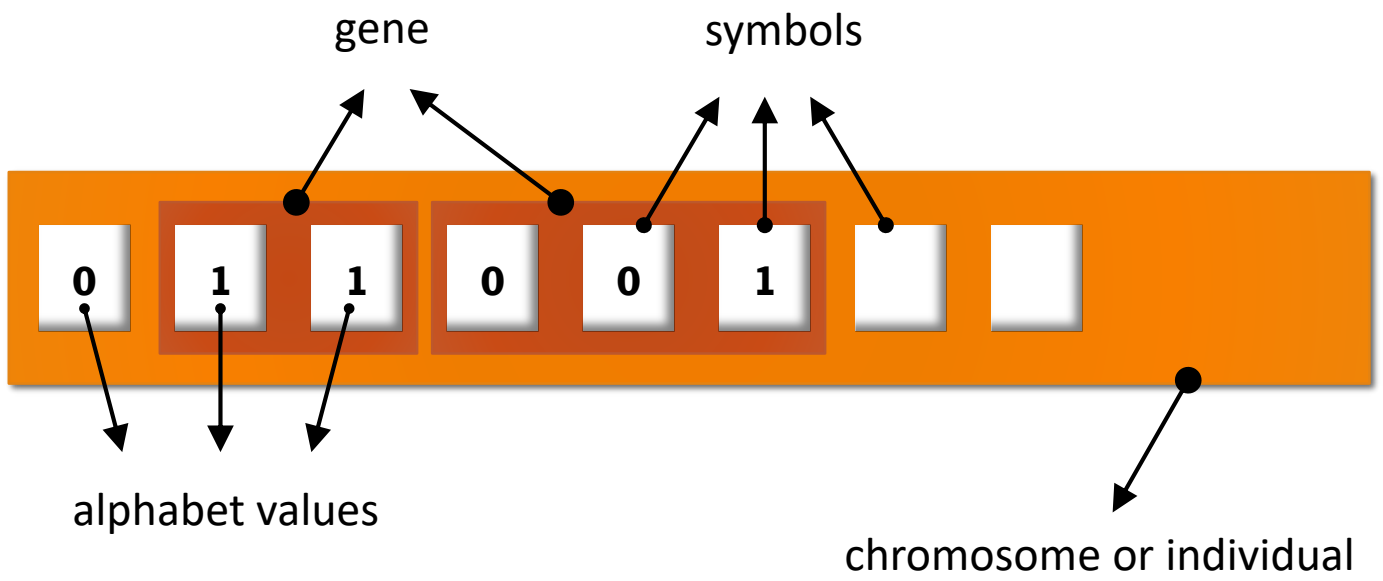
The **environment**, **inputs** and **outputs** are represented by **sets of symbols**, of fixed size, of a given alphabet:

- { 0; 1 }
- { X; Y; Z }
- { i; ii; iii; iv; iv; v; ... }
- $\mathbb{R}$
- ...

- We want to code candidates in a way that facilitates namely, mutation and crossing;
- The typical representation of the candidate is a binary string;
- This chain can be considered as a candidate's genetic code - hence the term “genetic algorithm”!;
- Other representations are possible, but usually make crossing and mutation more difficult.

Representation Model

Each point, in the domain of the environment, can be considered an individual, represented by a sequence (string) generated from the alphabet, constituted in the form:



ISLab
Synthetic Intelligence Lab

Representation Model

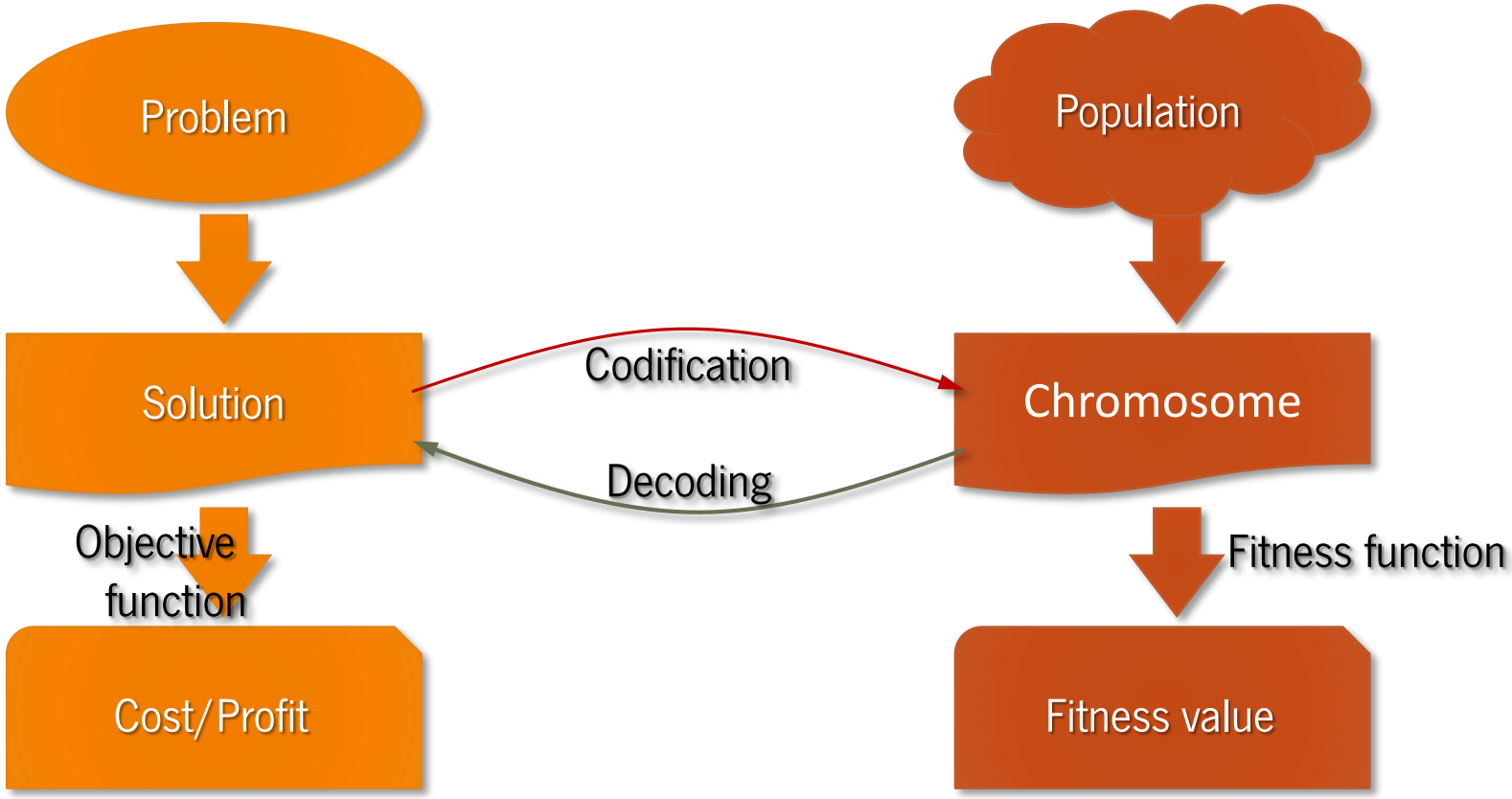
- At a given point in time, the system is characterized by a **population** (set of strings), representing the **current set of solutions** to the problem;
- The **evolution** is based on an **adaptive strategy**, by applying a process of **measuring the performance** (**fitness**) of the population (set of solutions);
- **Time** is measured at discrete intervals, called **generations**, defining the originated state transitions.

Representation of the Problem

- The basic entity for the representation of knowledge is the chromosome, composed of genes;
- Assuming the use of a binary encoding:
  - To represent 8 values (e.g., the 7 colors of the rainbow plus black):
    - use 3 symbols in 1 gene $\rightarrow 2^3 = 8$
  - To represent 5 values (e.g., to represent the fingers of a hand or the colors of the eye iris):
    - use 3 symbols in 1 gene $\rightarrow 2^3 = 8 > 5$
    - use 2 symbols in 1 gene $\rightarrow 2^2 = 4 < 5$

ISLab
Synthetic Intelligence Lab

Solutions Evaluation

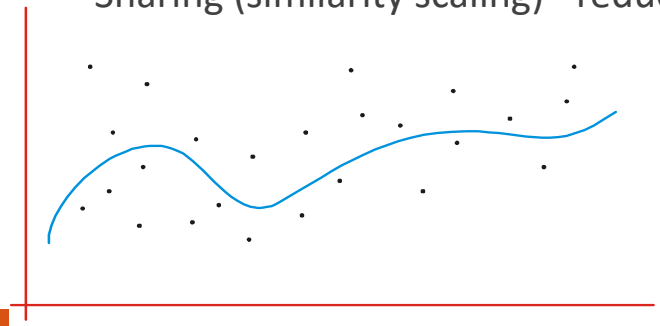


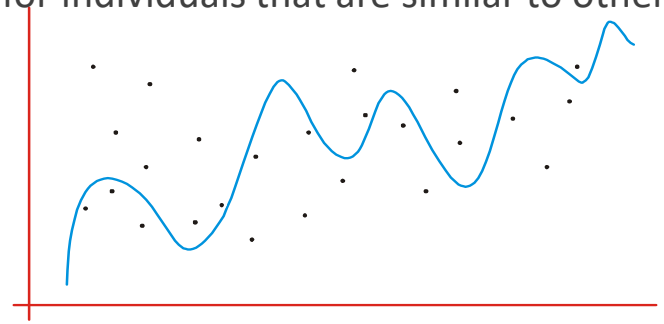
Fitness function

- Analogous to a heuristic that estimates how close a candidate is to be a solution;
- In general, the fit function should be consistent for the best performance.
- However, even if it is, there are no guarantees given that this is a probabilistic algorithm!

$\mathcal{F}($ [0] [1] [1] [0] [0] [1] $) = X$

- Fitness is computed for each individual
  - To help maintain diversity or differentiate between similar individuals, raw objective scores are sometimes scaled to produce the final fitness scores
  - Rank - no scaling
  - Linear scaling (fitness proportionate) - normalizes based on min and max fitness in the population
  - Sigma truncation scaling - normalizes using population mean and std. dev., truncating low-fitness individuals
  - Sharing (similarity scaling) - reduces fitness for individuals that are similar to other individuals in the population





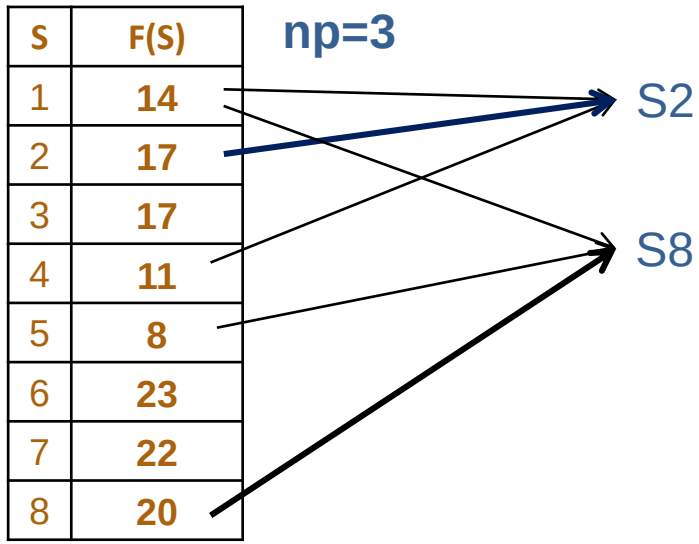
Selection

- The selection of parents for reproduction must ensure that individuals with a higher aptitude value have, proportionally, more offspring;
- Ideally, each individual should be represented by a number of descendants equivalent to the ratio between his aptitude value and the average value of the population;
- Select the best individuals can result in premature convergence, so the best selection schemes are designed to maintain a diverse population:
  - Rank - pick the best individuals;
  - Roulette wheel- the probability of selection is proportional to fitness;
  - Tournament - an initially large number are selected via roulette wheel, then the best ranked are chosen;
  - Stochastic - various methods of replenishment of less fit stock (useful) or initial selection (not useful);
  - Elite - in combination with other selection schemes, always keep the fittest individual around.
  - …

Selection

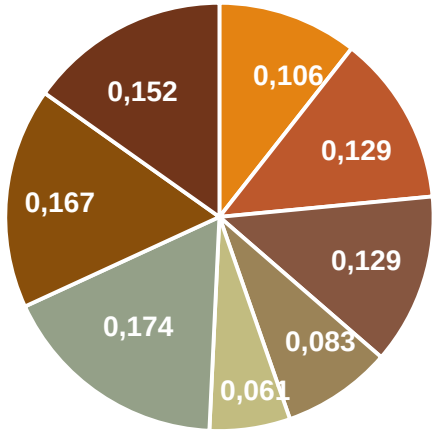
Select the best generation of the current population for parents

S	F(S)
1	14
2	17
3	17
4	11
5	8
6	23
7	22
8	20



Tournament

- Set number of participants (np)
- Random selection of participants



S	Prb(S)
1	0,106
2	0,129
3	0,129
4	0,083
5	0,061
6	0,174
7	0,167
8	0,152

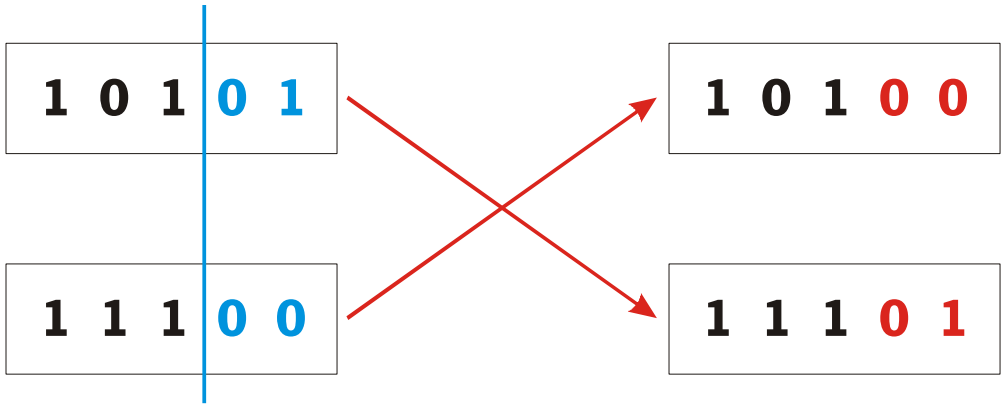
Roulette wheel

- Assign relative solution value
- Selection of participants according to the distribution of probabilities

ISLab
Synthetic Intelligence Lab

Crossover

- The genetic operator Crossover is an analogy with the sexual reproduction of living beings;
- The purpose of the crossing is to obtain, in the descendants, a combination of the genetic material of the parents (inheritance);
- The Crossover operator is applied to two individuals in the population, producing two other individuals for the population of the next generation;
- The crossing rate parameter (Cr) is used to define the probability of application of this operator.



ISLab
Synthetic Intelligence Lab

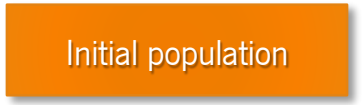
Mutation

- The genetic operator Mutation is intended to be an analogy to the genetic mutation of living beings;
- The mutation is a unary genetic operator: it applies to an individual in the population to generate a new individual for the next population;
- Explores parts of the space that crossover might miss and helps prevent premature convergence;
- The Mutation Rate (Mr) defines the probability of occurring in a given position for a given individual in the population.

ISLab
Synthetic Intelligence Lab

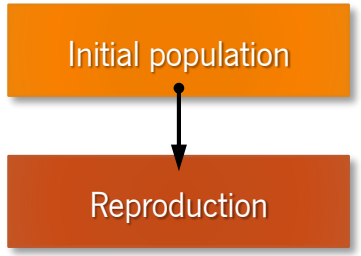
Execution Cycle

- Initial Population :
  - The initial population is chosen at random.

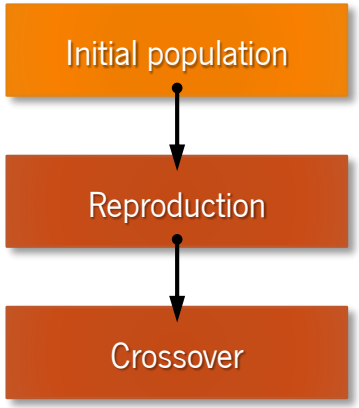


Execution Cycle

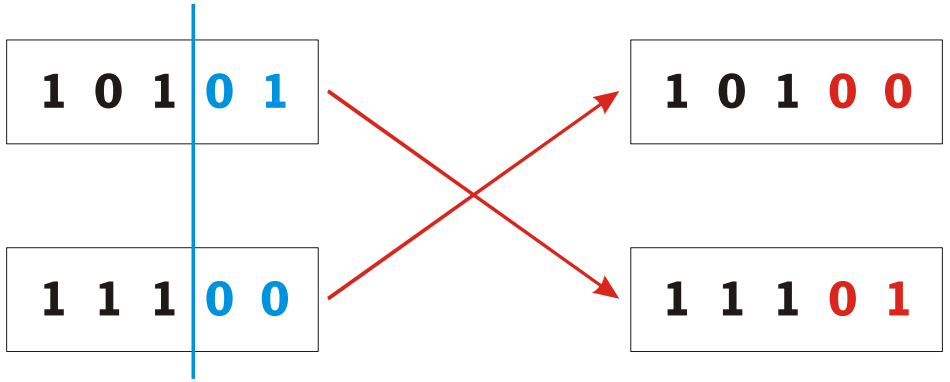
- Reproduction:
  - **Reproduction** is made up of **Evaluation** and **Selection**:
    - the **Evaluation** consists of the application of the evaluation function to all individuals in the population;
    - the next generation structures are obtained by **selecting** the members of the previous generation, through a process that must guarantee the choice of the structures with the best performance.



Execution Cycle



- Crossover:
  - In pairs of elements, the values are crossed from the same position;



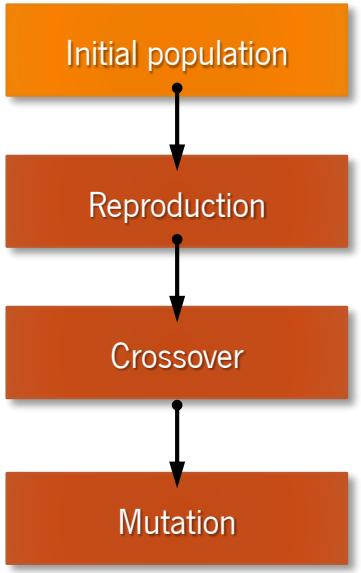
  - this operation allows to create new development points, within the same problem.

ISLab
Synthetic Intelligence Lab

Execution Cycle

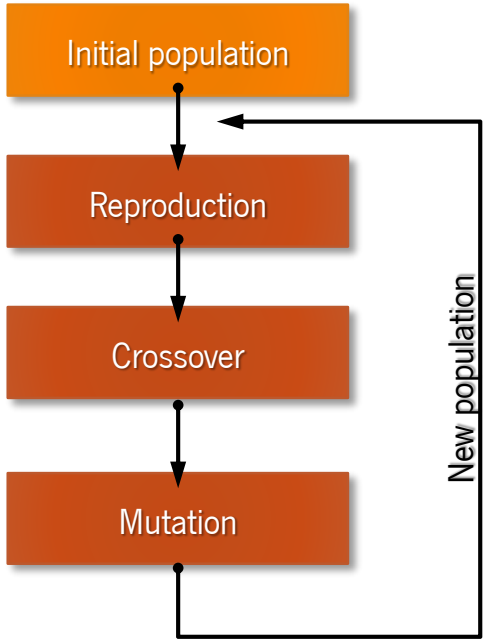
- Mutation:
  - Creates new elements, modifying one or more symbols, chosen at random;

  - Applies with low probability;
  - Ensures that it is always possible to reach any position of the problem;
  - Prevents total loss of information through selection and elimination.



Execution Cycle

- The **New Population**, obtained by the application of genetic operators, is again submitted to the same Cycle of Execution.



ISLab
Synthetic Intelligence Lab

End of Cycle

- The generational process is repeated until a completion condition is reached.
- Common termination conditions are:
  - Optimal solution has been reached;
  - A solution is found that meets minimum criteria;
  - Fixed number of generations affected;
  - Allocated budget (computation time/money) reached;
  - The suitability of the highest-ranking solution is reaching or has already reached a level such that successive iterations no longer produce better results;
  - Manual inspection;
  - Combinations of the above ideas.

ISLab
Synthetic Intelligence Lab

Outcome

- Genetic algorithms represent a wide range of global optimization methods;
- They do not use derivative functions to search for optimal solutions, which avoids “the temptation” to fall into local *minimum or maxima*;
- The resolution of a problem using GA’s proposes a set of solutions:
  - especially useful in multi-purpose optimization problems;
  - enhances parallelism in the search for solutions.

ISLab
Synthetic Intelligence Lab

Genetic Parameters

- **Population Size (N):**
  - a small population can cause poor performance, as it does not adequately cover the problem space, giving rise to local solutions;
  - a large population can avoid the previous problems, but it can affect the computational efficiency of the system.
- **Crossover Rate (Cr):**
  - amount of chromosomes used for the crossing: N x Cr
- **Mutation Rate (Mr):**
  - the mutation is used to increase population variability;
  - each gene has a finite probability of changing;
  - a low mutation rate allows a gene to "freeze" in a value;
  - a high rate of mutation results in a random search for solutions;
  - where L is the length of the chromosome, Mr x N x L mutations will occur.
- **Replacement Rate (Generation Gap - Gr):**
  - controls the percentage of the population to be replaced in each generation;
  - population structures are replaced by N x Gr;
  - if Gr = 1, it means that the entire population must be replaced during each generation.

ISLab
Synthetic Intelligence Lab

Genetic Parameters

- **Selection Strategy**:
  - the reproduction is done based on the proportion of the fitness value of the structures of the current population;
  - the structures with the best performance are those that pass to the next generation.
- **Evaluation Function**:
  - it serves to maintain the diversity of the population during evolution;
  - the emergence of a dominant “super chromosome” in the population must be avoided, so that the problem can be efficiently explored.

ISLab
Synthetic Intelligence Lab

When and Where

- Multimodal functions;
- Discrete or continuous functions;
- Highly dimensional functions, including combinatorics;
- Non-linear dependence on parameters;
- Often used to solve NP problems;
- Do not use GA when another method such as hill-climbing, etc., works well or at least before trying that type of method!

ISLab
Synthetic Intelligence Lab

Knapsack problem

- The knapsack problem is a combinatorial optimization problem;
- It is intended to fill a backpack with objects of different weights and values;
- The goal is to fill the backpack with the highest possible value, not exceeding the maximum weight
- The backpack problem is one of the 21 NP-complete problems of Richard Karp, exposed in 1972;
- The formulation of the problem is extremely simple, but its solution is complex.

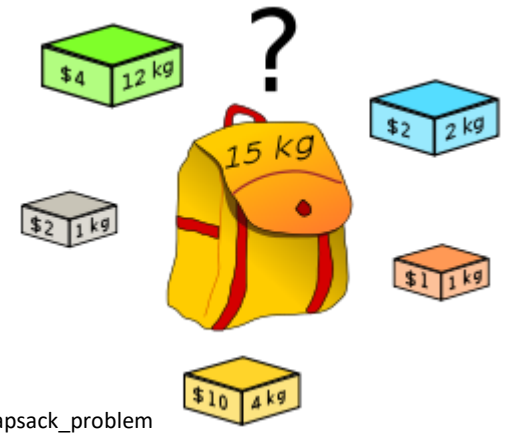


Image Source: https://en.wikipedia.org/wiki/Knapsack_problem

ISLab
Synthetic Intelligence Lab

Knapsack problem

- Example of a one-dimensional (constraint) knapsack problem: which boxes should be chosen to maximize the amount of money while still keeping the overall weight under or equal to 15 kg?
- A multiple constrained problem could consider both the weight and volume of the boxes.

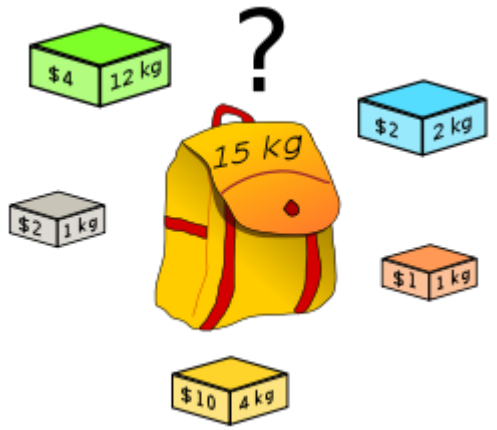


Image Source: https://en.wikipedia.org/wiki/Knapsack_problem

Knapsack problem

Example:

N = 8

Capacity (C) = 50

	1	2	3	4	5	6	7	8
Value	4	3	6	7	2	9	7	6
Weight	12	16	8	21	16	11	6	12

Objective

Maximize $\sum_i O_i \times v_i$

Constraint $\sum_i O_i \times p_i \leq 50$

?

Representation of solutions: Oi

1	1	0	0	0	0	1	0

Knapsack problem

	1	2	3	4	5	6	7	8
Value	4	3	6	7	2	9	7	6
Weight	12	16	8	21	16	11	6	12

Evaluation

$$f(s) = \begin{cases} \sum_i O_i \times v_i & if\ solution\ is\ valid \\ 0 & if\ solution\ is\ not\ valid \end{cases}$$

$$f(s) = \sum_i O_i \times v_i - p(s)$$

where $$p(s) = \begin{cases} \alpha\left(\sum_i O_i \times v_i\right) - C & if\ solution\ is\ not\ valid \\ 0 & if\ solution\ is\ valid \end{cases}$$

The penalty for a valid solution is zero and is proportional to the degree of violation of the restrictions

ISLab
Synthetic Intelligence Lab

Knapsack problem

	1	2	3	4	5	6	7	8
Value	4	3	6	7	2	9	7	6
Weight	12	16	8	21	16	11	6	12

Create and evaluate the initial population

Solution								Value	Weight	p(s)	f(s)
1	1	0	0	0	0	1	0	14	34	0	14
0	0	0	1	1	0	1	1	22	55	5	17
0	1	0	1	0	0	1	0	17	43	0	17
1	0	0	0	0	0	1	0	11	18	0	11
0	1	1	1	1	0	1	0	25	67	17	8
1	0	0	1	0	0	1	1	24	51	1	23
0	1	0	1	0	1	1	0	26	54	4	22
1	1	0	0	0	0	1	1	20	46	0	20

ISLab
Synthetic Intelligence Lab

Example Recombination

Recombinant solutions

- Combining the genetic material of two parents to generate new solutions
- Objective: To combine interesting features of two solutions

Cut-off recombination

- Align the two parents
- Select a random cut point
- Combine complementary sections to obtain descendants

ISLab
Synthetic Intelligence Lab

Example Recombination

Initial population

Solução (progenitores)								Value	Weight	p(s)	f(s)
1	1	0	0	0	0	1	0	14	34	0	14
0	0	0	1	1	0	1	1	22	55	5	17
0	1	0	1	0	0	1	0	17	43	0	17
1	0	0	0	0	0	1	0	11	18	0	11
0	1	1	1	1	0	1	0	25	67	17	8
1	0	0	1	0	0	1	1	24	51	1	23
0	1	0	1	0	1	1	0	26	54	4	22
1	1	0	0	0	0	1	1	20	46	0	20

Recombinação

Solution (children)								Value	weight	p(s)	f(s)
0	0	0	1	1	0	1	0	16	43	0	16
1	0	0	0	0	0	1	1	17	30	0	17
0	1	0	1	0	0	1	1	23	55	5	18
1	1	0	0	0	1	1	0	23	45	0	23

ISLab
Synthetic Intelligence Lab

Example Mutation

Changing the value of a gene (binary mutation)

Creates variability in the set of solutions

Introduces changes in genetic material

Objective: Introduce diversity in the population

Children population

Solution (Children)								Value	Weight	p(s)	f(s)
0	0	0	1	1	0	1	0	16	43	0	16
1	0	0	0	0	0	1	1	17	30	0	17
0	1	0	1	0	0	1	1	23	55	5	18
1	1	0	0	0	1	1	0	23	45	0	23

Mutation

Solution (Children)								Value	Weight	p(s)	f(s)
0	0	0	1	1	0	1	0	16	43	0	16
1	0	0	0	0	1	1	1	26	41	0	26
0	1	0	1	0	0	1	1	23	55	5	18
1	1	0	0	0	1	1	0	23	45	0	23

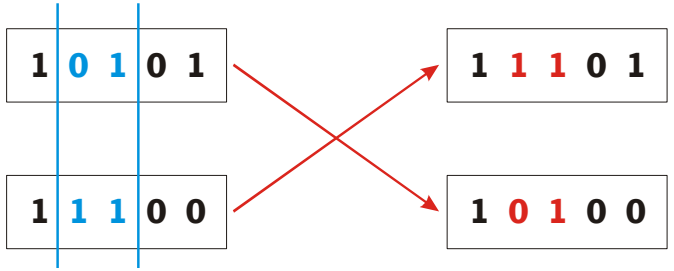
ISLab
Synthetic Intelligence Lab

Other approaches: Representation

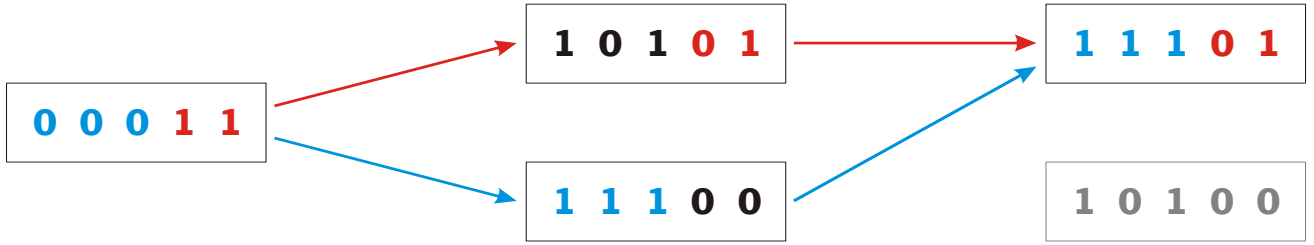
- Real Representations:
  - Closer to real problems;
  - Crossing alters “genes” and not symbols of the genes;
  - The Mutation will replace the value of a gene with another random value.
- Integer Representations:
  - For situations where the domain of solutions is discreet;
  - The Crossing has the same meaning as before;
  - Mutation implies greater computational power: random or sequential choice.
- Permutation Based Representations:
  - The order of genes on the chromosome is not relevant;
  - Each individual is built by permuting a set of values;
  - Repeated values are not allowed;
  - The length of the chromosome is given by the cardinality of the alphabet.
- Ordinal Representations:
  - They are a “manipulation” of Permutation-Based Representations;
  - It has the objective of allowing the use of the genetic operators of Crossover and Mutation of binary representations;
  - The value of each gene is an integer in the range [1, N-i + 1].

Other approaches: Crossover

- Two-point crossing:
  - Individuals form "rings";
  - Two points are used to mark the beginning and end of the crossing material.



- Uniform crossing:
  - A binary mask, generated at random, is used;
  - In the position where the mask has the value "1", the descendant adopts the value of the first parent; when the value is "0", the value of the second parent is copied;
  - For the second descendant, the procedure is reversed.



ISLab
Synthetic Intelligence Lab

Pros and Cons

Pros

- Faster (and lower memory requirements) than searching in a very large search space systematically;
- Easy, because if the representation function and the suitability of candidates is correct, a solution can be found without any explicit analytical work.

Cons

- Random - it is not ideal nor is it a complete algorithm;
- It can get stuck in the local max/min, although crossover and mutation can help mitigate that;
- It can be difficult to figure out how best to represent a candidate as a string of bits (or another);
- Heavier than Hill-Climbing or other algorithms (“individual-based”)

ISLab
Synthetic Intelligence Lab

References

Holland, John H. “Genetic Algorithms.” *Scientific American* 267 (1): 66–73, 1992.

David Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning”, Addison Wesley, 1989.

Michalewicz, Zbigniew. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Science & Business Media, 2013.

Yang, Xin-She. *Nature-Inspired Optimization Algorithms*.

Elsevier, 2014.

Miguel Rocha, José Neves, “Computação Genética e Evolucionária”, Universidade do Minho, 2000

ISLab

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Genetic and Evolutionary Algorithms

Mestrado Integrado em Engenharia Informática
Mestrado em Engenharia Informática
Perfil SI :: Computação Natural