Genetic Algorithms

Rui Mendes

Introduction

Characteristics

- Inspired in living systems
- Borrow ideas from evolution

Examples

- Genetic Algorithms
- Genetic Programming

Genetic Algorithm

- Optimization algorithm
- Inspired in natural selection
- Uses a population of individuals
- Each individual encodes a solution
- There are three operators:
    1. Selection
    2. Crossover
    3. Mutation

Encoding

- Process that represents a solution in a given alphabet
- There are several encodings, e.g.:
  - binary
  - integer
  - real numbers
  - permutations
- Usually, they are linear sequences of values (chromossomes composed of genes)

Creating new solutions

- The genetic operators for creating new solutions are:
  - Recombination or Crossover
  - Mutation

Crossover

- Combines several solutions
- Usually, it takes two parents and creates two offspring
- Usual methods:
  - One point crossover
  - Two point crossover
  - Uniform crossover

One point crossover

- A cutting point is randomly selected
- Each solution takes a part from each parent

Parents

010|110
011|101

Offspring

010|101
011|110

Two point crossover

- Two cutting points are randomly selected
- The central parts are exchanged

Parents

10|11|0
01|10|1

Offspring

10|10|0
01|11|1

Uniform crossover

- The value of the first parent is randomly copied to one of the offspring
- The value of the second parent is copied to the other one

Parents	Offspring
10110	00100
01101	11111

Mutation

- A position is randomly selected inside the string
- The value is randomly selected among the possible values

Before

10110

After

10010

Fitness

- Quantifies the quality of a solution
- It is usually a *floating point number*
- In order to compare the quality of two solutions, one simply has to compare the fitness values

Selection

- Process for choosing which individuals will generate offspring or will be copied into the new population
- It usually selects individuals:
  - According to their fitness;
  - By partial or total ordering of the solutions
  - By stochastic choice of the individuals according to their fitness

Fitness proportionate selection

- The probability of selecting an individual is proportional to their fitness
- If it is a maximization problem, the probability of selecting an individual is $p_i = \frac{f_i}{\sum_i f_i}$

Selection by ordering

- If using total ordering, the individuals are ranked according to their fitness and the top n individuals are chosen
- In a partial ordering, several individuals are chosen at random and they are ordered (it's called *tournament selection*)

Genetic Algorithm outline

1. Initialize the population
2. While not finished
    1. Choose individuals
    2. Apply genetic operators
    3. Insert offspring into the next population

Generational scheme

- A new population is created in each iteration
- Parents are chosen from the current population
- Offspring are inserted into the new population
- The old population is discarded

Generational scheme

1. Initialize population
2. While not finished
    1. Evaluate individuals
    2. While the new population is not complete
        1. Select two individuals
        2. Apply genetic operators
        3. Insert offspring into the new population

Steady state scheme

- There are not generations
- Several individuals are randomly chosen for the tournament
- The best individuals produce offspring using the genetic operators
- The offspring substitute the worse individuals

Steady state scheme

1. Initialize population
2. While not finished
   1. Choose 4 individuals
   2. Rank the individuals
   3. Apply genetic operators to the best 2 individuals
   4. The offspring substitute the worst 2 individuals

Representations

- There can be several solution representations
- Each representation has several advantages and disadvantages
- There are several possibilities:
  - Integer
  - Real
  - Ordered

Integer representation

- Each individual is a sequence of integers, e.g.,

$$2 \quad 1 \quad 2 \quad 3 \quad 7$$

- Crossover operators defined thus far can be used
- There can be new crossover and mutation operators

Random value mutation

$$2\ 1\ 2\ 3\ 7 \rightarrow 2\ \color{red}{5}\ 2\ 3\ 7$$

Next element mutation

$$2\ 1\ 2\ 3\ 7 \rightarrow 2\ \color{red}{2}\ 2\ 3\ 7$$

Real representation

- Each individual is a sequence of real numbers, e.g.,

$$2.3 \quad -7.4 \quad 1.9 \quad 0.1$$

- There are specific crossover operators:
  - Arithmetic crossover
  - Linear crossover
  - Blend crossover
- And mutation operators:
  - Uniform mutation
  - Gaussian mutation
  - Cauchy modification

Arithmetic crossover

- Given two parents, $\vec{P}_1$ and $\vec{P}_2$
- Create 2 solutions:

  $\vec{S}_1 : \lambda \cdot \vec{P}_1 + (1 - \lambda) \cdot \vec{P}_2$

  $\vec{S}_2 : (1 - \lambda) \cdot \vec{P}_1 - \lambda \cdot \vec{P}_2$
- With λ usually being 0.6

Linear crossover

- Given two parents, $\vec{P}_1$ and $\vec{P}_2$
- Create 3 solutions:

  $\vec{S}_1 : 0.5 \cdot \vec{P}_1 + 0.5 \cdot \vec{P}_2$

  $\vec{S}_2 : 1.5 \cdot \vec{P}_1 - 0.5 \cdot \vec{P}_2$

  $\vec{S}_3 : 0.5 \cdot \vec{P}_1 + 1.5 \cdot \vec{P}_2$
- Choose the two best solutions among $\vec{P}_1$, $\vec{P}_2$, $\vec{S}_1$, $\vec{S}_2$ and $\vec{S}_3$ to become the children

Blend crossover

- Given two parents, x_1 and x_2 where $x_1 < x_2$
- Create solutions using:

$$U[x_1 - \alpha \cdot (x_2 - x_1), x_2 + \alpha \cdot (x_2 - x_1)]$$

- α is usually chosen to be 0.5.

Uniform mutation

- A random mutation where one or more positions are modified
- Modified positions are randomly generated within the entire range
- The modified position has no relationship with the original value
- $x_{new} = U[\min, \max]$

Random modification

- A random mutation where one or more positions are modified
- Modified positions are generated in the vicinity of the original value
- $x_{new} = x + U[-\epsilon, \epsilon]$ or
- $x_{new} = x \cdot (1 + U[-\epsilon, \epsilon])$

Gaussian mutation

- A random mutation where one or more positions are modified
- Modified positions are generated in the vicinity of the original value following the gaussian distribution
- $x_{new} = x + N(0, \sigma)$

Permutation based representations

- Each individual is a sequence of integers representing a permutation
- The integers are in the range $1, \ldots, n$ without repetitions
- Example:

$$2 \quad 1 \quad 5 \quad 4 \quad 3$$

- Genetic operators must preserve these constraints

Order preserving crossover

- Select a cutting position and copy the parts left of the cutting point directly to the corresponding offspring
- Copy the remaining values by using the order in the other parent
- Can also be used with two cutting points

Order preserving crossover

2 1 5 | 3 4	2 1 5 | 4 3
1 4 5 | 3 2	1 4 5 | 2 3

Partially mapped crossover

- Choose random segment from the first parent P_1
- Look for values in the other parent P_2 that were not copied
- For each of these values $x \in P_2$, look at what values $y \in P_1$ that was copied in its place
- Place x in the position occupied by y in P_2, because we know we will not put y there (because it's already in the child)
-

Insertion mutation

- Choose two positions at random
- Move the second position to follow the first

Insertion mutation

2 1 5 3 4 → 2 1 3 5 4

Swap mutation

- Choose two positions at random
- Swap their positions

Swap mutation

$$2\ 1\ 5\ 3\ 4 \rightarrow 2\ 3\ 5\ 1\ 4$$

Inversion mutation

- Choose two positions at random
- Invert the values between them

Inversion mutation

2 1 5 3 4 → 2 3 5 1 4

Scramble mutation

- Pick a random subset of genes
- Randomly change the position of the values between them
- The subset need not be contiguous

Scramble mutation

$2\ 1\ 7\ 5\ 3\ 6\ 4 \rightarrow 2\ 7\ 3\ 6\ 5\ 1\ 4$