Genetic Algorithms

Prof. Alessandro Lucantonio

Aarhus University

21/11/2023

GA - Main features

- Inspired by biological evolution (survival of the fittest, natural selection, and genetic inheritance)
- **Gradient-free/global** optimization
- Good choice when the search space is very large / multi-dimensional problems
- Relatively easy to implement and *parallelize*

Genetic Algorithms

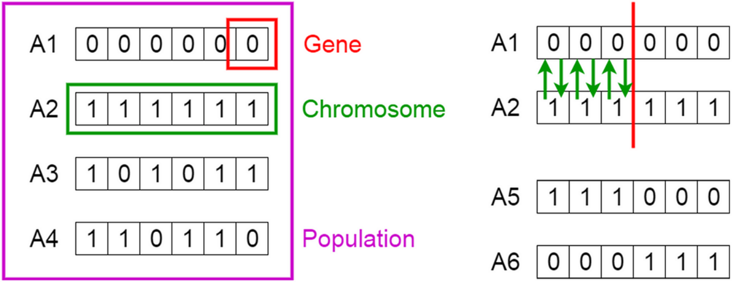
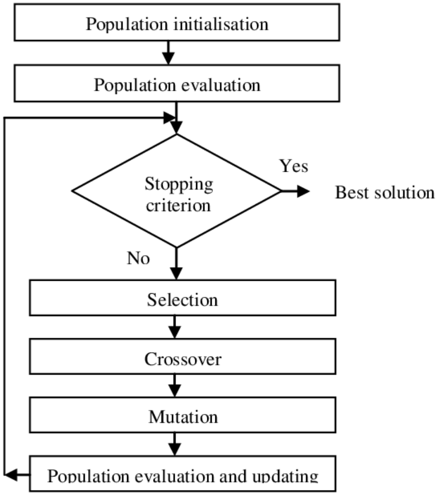


Figure: Genes, chromosomes (binary) and population in a GA.

Basic terminology

- **Population**: a collection of candidate solutions
- **Individual**: a candidate solution to the given problem; it is also called **chromosome**
- **Gene**: the indivisible building block making up an individual
- **Fitness**: a score that measures how good an individual is (as a solution to the given problem)
- **Mutation**: random modification of the genes of an individual
- **Crossover**: combination of the chromosomes of two or more individuals to create a new candidate solution
- **Selection**: choice of individuals to breed the next generation

GA flowchart



GA pseudo-code

1. Randomly generate a population of m parents
2. Repeat until reaching a stopping criterion
   2.1 Compute the fitness for each individual in the current parent population
   2.2 Select p parents form the population
   2.3 Generate m offspring by crossover
   2.4 Probabilistically mutate individuals of the offspring
   2.5 Replace the parent population with the offspring

Note: the best individuals in the parent population may be lost because the offspring population replaces the parent one (non-monotonic fitness). We can preserve the best using **elitist selection**.

Selection mechanisms

Selection pressure: greediness or exploitation pressure.

Some mechanisms ranked from high to low selection pressure:

- **Tournament**: randomly select k ($k = 2, 3$, typically) individuals using a uniform probability and then select the best (or worst) individual from the competitors as the winner (or loser). If p individuals need to be selected, p tournaments are performed.
- **Fitness-proportional**: each individual is assigned the probability f_i/f_{sum} where f_i is the fitness of individual i and f_{sum} is the total fitness of all the individuals in the current selection pool
- **Uniform**: select the parents using a uniform probability distribution

Crossover and mutation

Single-point crossover: call L the number of genes; randomly select a crossover point between genes i and $i+1$ and copy genes $1 \dots i$ from parent 1 and genes $i+1 ... N$ from parent 2.

- Parent 1: A B | C D E F
- Parent 2: a b | c d e f
- Child: A B c d e f

Gaussian mutation: suppose that the chromosome is made of real numbers; for each gene to be mutated (randomly selected or according to a specific criterion), draw a number from a Gaussian distribution and modify it by adding the extracted number to it.

Picture an individual as a point in an N-dim gene space: children produced by crossover correspond to vertices of the N-dim rectangle defined by the two parents. *Mutation* produces a child by forming a cloud around the parent: it provides a source of useful gene values, while crossover explores the lattice they define.

GA design - Exploration vs exploitation

High-level goal: effective balance between exploration of new regions of the search space and exploitation of the already explored regions.

- *Crossover* and *mutation* are the primary source of **exploration**, while *selection* controls **exploitation**
- Strong selection pressure should be balanced by more explorative reproductive (crossover/mutation) operators
- Size of parent population measures the degree of parallel search (increase for multi-peaked landscapes)

Model Predictive Control - Main features

MPC uses a **model** of the system to **predict** future states $x_i \in \mathbb{R}^n$, $i = 0, \ldots, N-1$ and optimize **control** inputs $u_i \in \mathbb{R}^m$, $i = 0, \ldots, M-1$ over a defined *time horizon*.

Key components:

- **Predictive Model**: Represents the dynamics of the system (linear/non-linear, discrete/continuous).
- **Horizon**: Consists of a prediction horizon (N future time-steps) and a control horizon (M future time-steps).
- **Cost function**: Defined to quantify the desired system performance, typically involves minimizing *error* (of system trajectory with respect to desired/*setpoint* $\bar{x}_i$) and *control effort* (here $N = M$):

$$J = \sum_{i=0}^{N-1} [s\|x_i - \bar{x}_i\|^2 + q\|u_i\|^2]$$

with s and q weights of the cost function terms.

Model Predictive Control - Algorithm

Basic steps:

1. Measure current state of the system;
2. Predict future states x_i using the model;
3. Solve an optimization problem to minimize the cost function J with respect to the controls u_i, subject to constraints (such as bounds on the controls).
4. Apply u_0 and go to step 1.

We can use a **genetic algorithm** to solve the optimization problem.