Genetic Algorithm

What is GA

- Developed by John Holland, University of Michigan (1970’s)
  - To understand the adaptive processes of natural systems
  - To design artificial systems software that retains the robustness of natural systems
- A **genetic algorithm** (or **GA**) is a search technique used in computing to find true or approximate solutions to optimization and search problems.
- Genetic algorithms are categorized as global search heuristics.
- Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

Key terms

- **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*

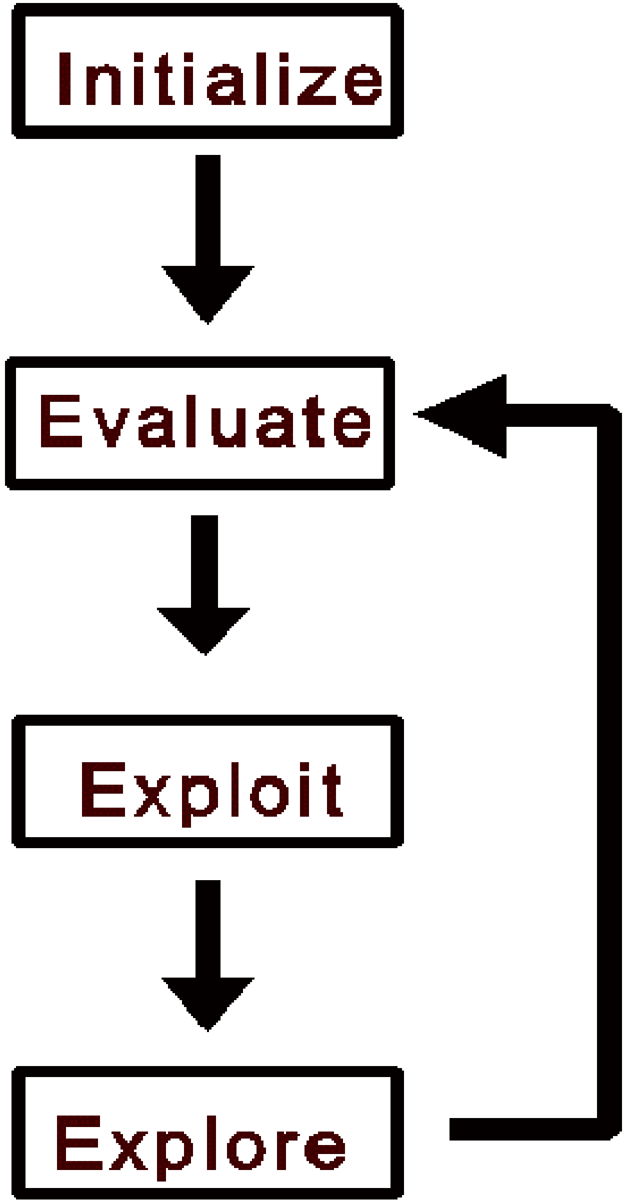
Genetic Algorithm

- heuristic method based on ‘ survival of the fittest ’
- useful when search space very large or too complex for analytic treatment
- in each iteration (generation) possible solutions or individuals represented as strings of numbers

3021 3058 3240

00010101 00111010 11110000
00010001 00111011 10100101
00100100 10111001 01111000

11000101 01011000 01101010

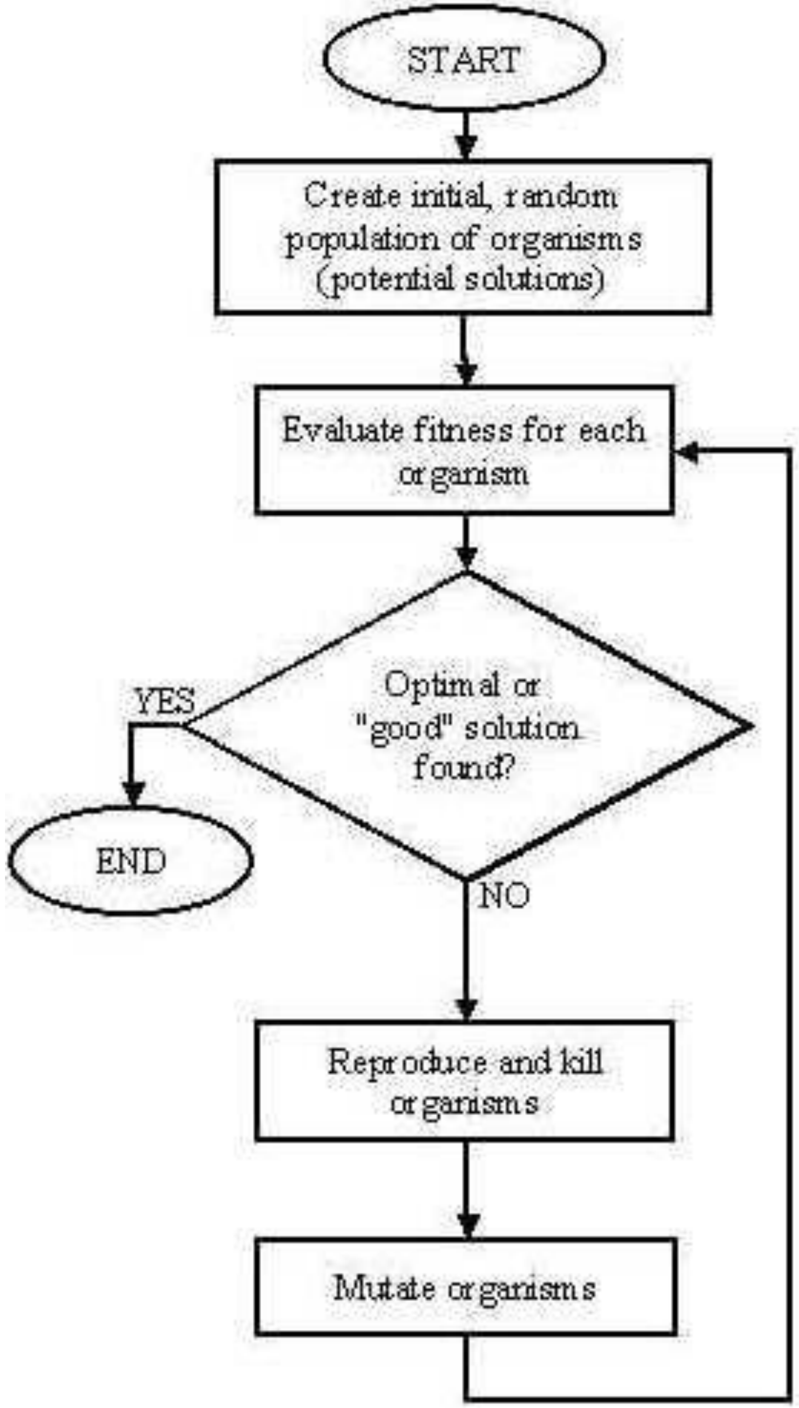


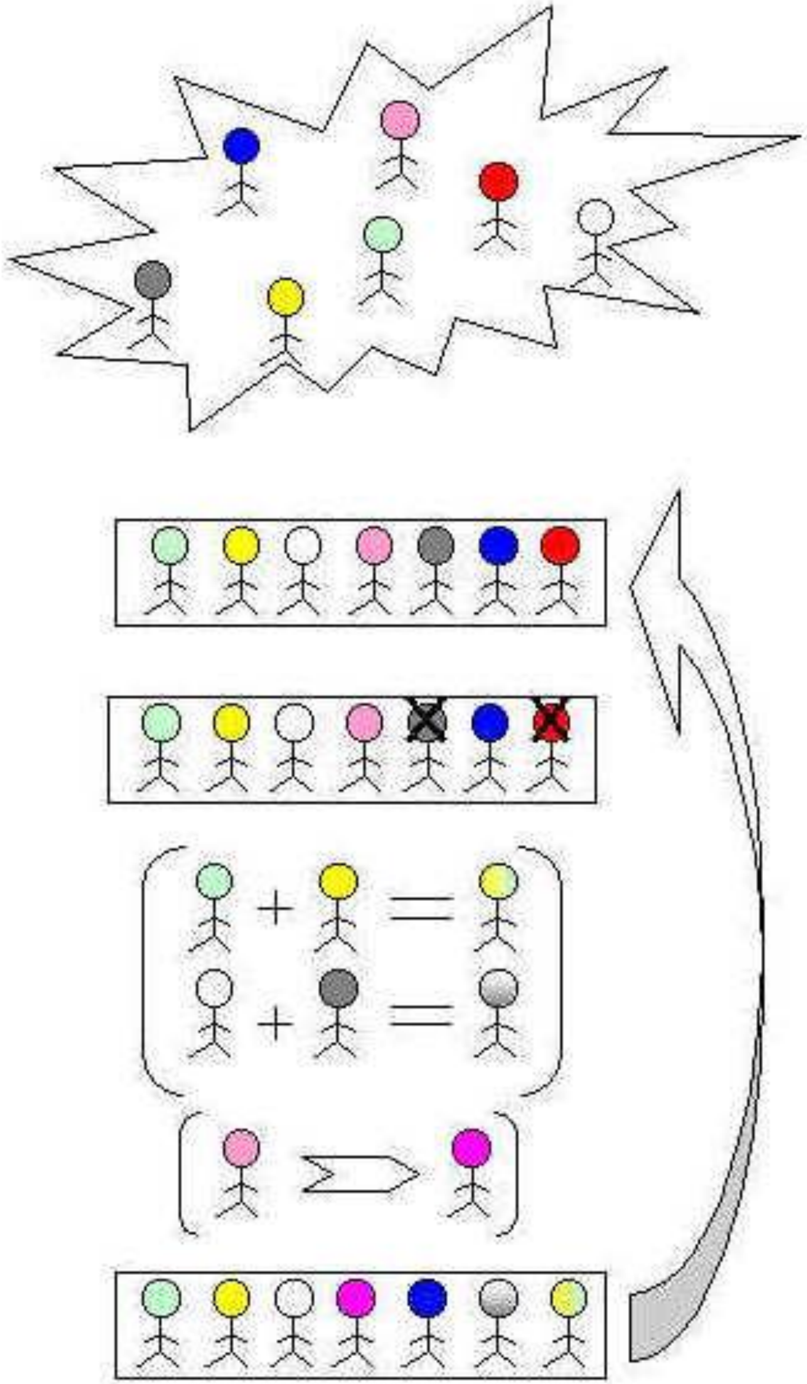
- all individuals in population evaluated by fitness function

- individuals allowed to reproduce (selection), crossover, mutate

Flowchart of GA

http://ib-poland.virtualave.net/ee/genetic1/3geneticalgorithms.htm





Basics of GA

- The most common type of genetic algorithm works like this:
- a population is created with a group of individuals created randomly.
- The individuals in the population are then evaluated.
- The evaluation function is provided by the programmer and gives the individuals a score based on how well they perform at the given task.
- Two individuals are then selected based on their fitness, the higher the fitness, the higher the chance of being selected.
- These individuals then "reproduce" to create one or more offspring, after which the offspring are mutated randomly.
- This continues until a suitable solution has been found or a certain number of generations have passed, depending on the needs of the programmer.

General Algorithm for GA

- **Initialization**
- Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions.
- Traditionally, the population is generated randomly, covering the entire range of possible solutions (the *search space*).
- Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

General Algorithm for GA

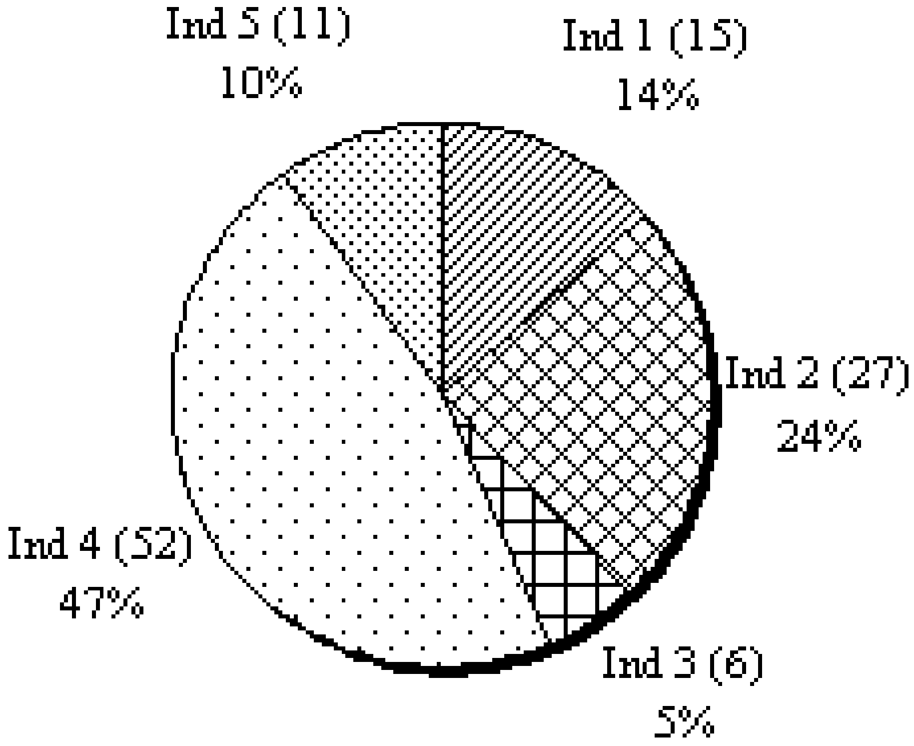
- **Selection**
- During each successive generation, a proportion of the existing population is selected to breed a new generation.
- Individual solutions are selected through a *fitness-based* process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected.
- Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time-consuming.
- Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions. Popular and well-studied selection methods include roulette wheel selection and tournament selection.

General Algorithm for GA

- 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.

Population	*Fitness*
Individual 1	15
Individual 2	27
Individual 3	6
Individual 4	52
Individual 5	11



Randomly generated number = 21

Individual 2 is selected

Roulette Wheel Selection

General Algorithm for GA

Roulette Wheel’s Selection Pseudo Code:

```
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
```

General Algorithm for GA

- **Reproduction**
- The next step is to generate a second-generation population of solutions from those selected through genetic operators:

  crossover (also called recombination), and/or mutation.
- For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously.
- By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each child, and the process continues until a new population of solutions of appropriate size is generated.

General Algorithm for GA

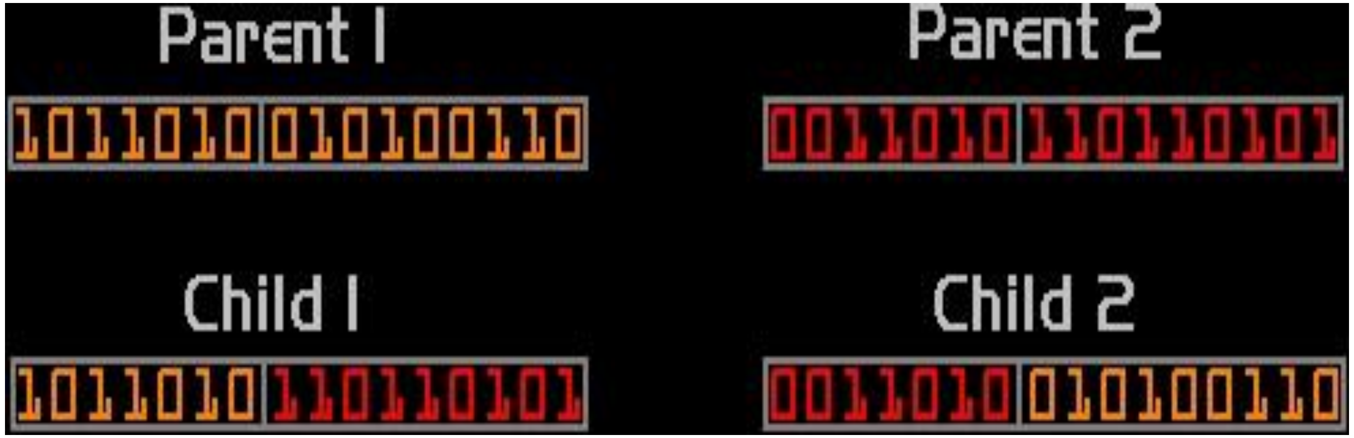
- These processes ultimately result in the next generation population of chromosomes that is different from the initial generation.

- Generally, the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

Crossover

- the most common type is single point crossover. In single point crossover, you choose a locus at which you swap the remaining alleles from on parent to the other. This is complex and is best understood visually.
- As you can see, the children take one section of the chromosome from each parent.
- The point at which the chromosome is broken depends on the randomly selected crossover point.
- This method is called single point crossover because only one crossover point exists. Sometimes only child 1 or child 2 is created, but oftentimes both offspring are created and put into the new population.
- Crossover does not always occur, however. Sometimes, based on a set probability, no crossover occurs, and the parents are copied directly to the new population. The probability of crossover occurring is usually 60% to 70%.

Crossover



Parent 1
1011010 010100110
Parent 2
0011010 110110101
Child 1
1011010 110110101
Child 2
0011010 010100110

Mutation

- After selection and crossover, you now have a new population full of individuals.
- Some are directly copied, and others are produced by crossover.
- In order to ensure that the individuals are not all exactly the same, you allow for a small chance of mutation.
- You loop through all the alleles of all the individuals, and if that allele is selected for mutation, you can either change it by a small amount or replace it with a new value. The probability of mutation is usually between 1 and 2 tenths of a percent.
- Mutation is fairly simple. You just change the selected alleles based on what you feel is necessary and move on. Mutation is, however, vital to ensuring genetic diversity within the population.

Mutation

General Algorithm for GA

- **Termination**
- This generational process is repeated until a termination condition has been reached.
- Common terminating conditions are:
  - A solution is found that satisfies minimum criteria
  - Fixed number of generations reached
  - Allocated budget (computation time/money) reached
  - The highest-ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
  - Manual inspection
  - Any Combinations of the above

GA Pseudo-code

Choose initial population

Evaluate the fitness of each individual in the population

Repeat

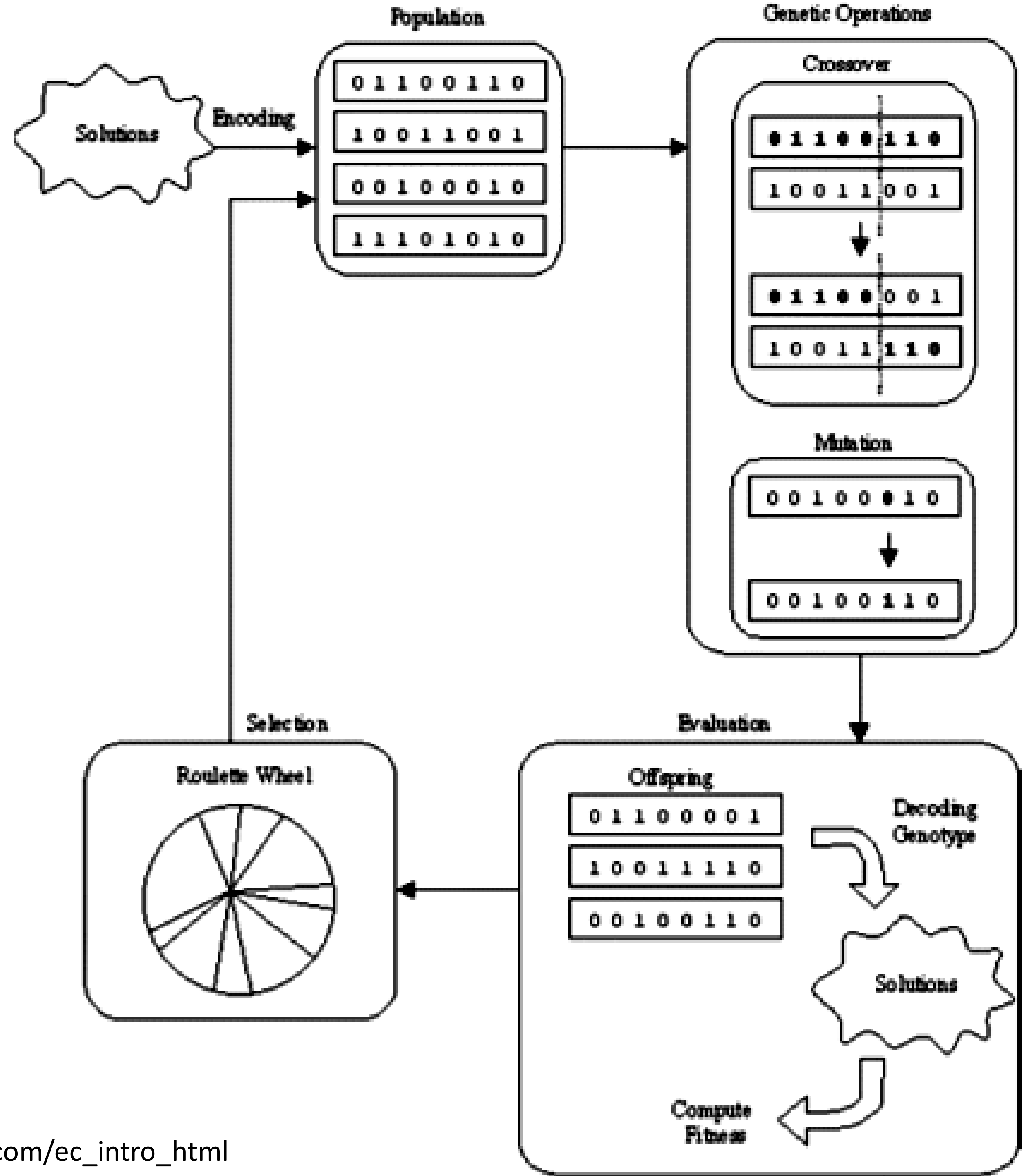
    Select best-ranking individuals to reproduce

    Breed new generation through crossover and mutation (genetic operations) and give birth to offspring

    Evaluate the individual fitnesses of the offspring

    Replace worst ranked part of population with offspring

Until <terminating condition>



Population
Genetic Operations
Crossover
Solutions
Encoding
0 1 1 0 0 1 1 0
1 0 0 1 1 0 0 1
0 0 1 0 0 0 1 0
1 1 1 0 1 0 1 0
Mutation
Selection
Evaluation
Roulette Wheel
Offspring
0 1 1 0 0 0 0 1
1 0 0 1 1 1 1 0
0 0 1 0 0 1 1 0
Decoding Genotype
Solutions
Compute Fitness

Some Applications

Decision making / decision support systems

Engineering component / equipment design

Engineering process optimization

Portfolio optimization

Route optimization; optimal layout; optimal packing

Schedule optimization

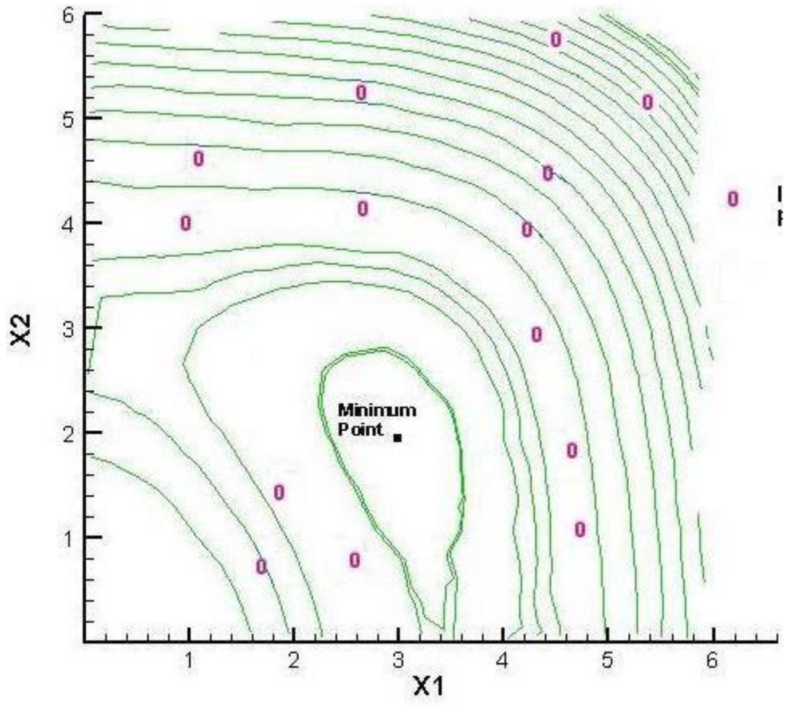
Protein structure analysis

Genetic Algorithms

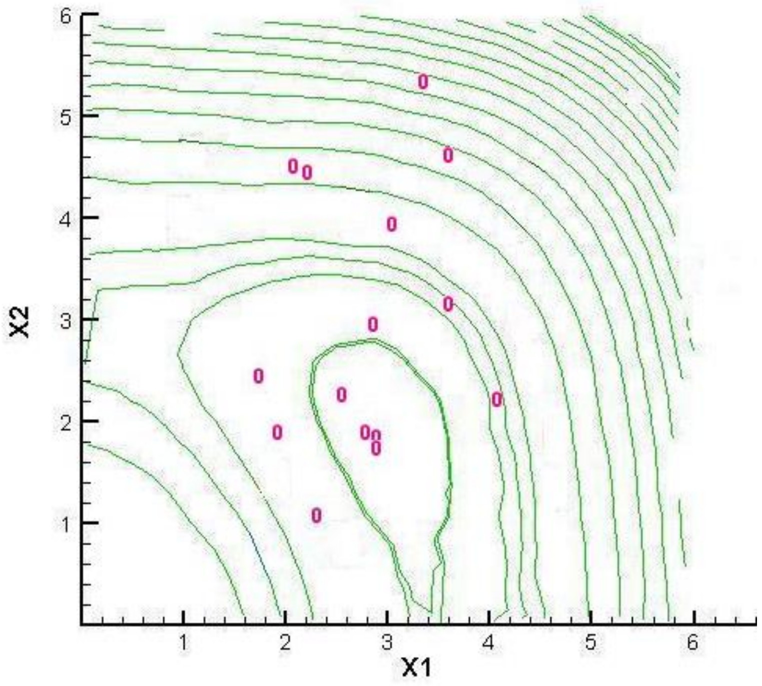
Results from a small example:

Minimize $$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

$$0 \le x_1, x_2 \le 6$$

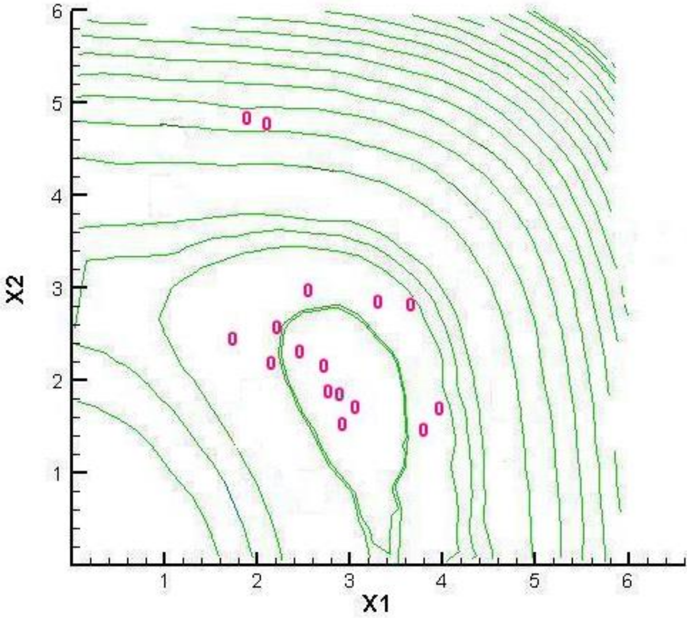


Initial Population

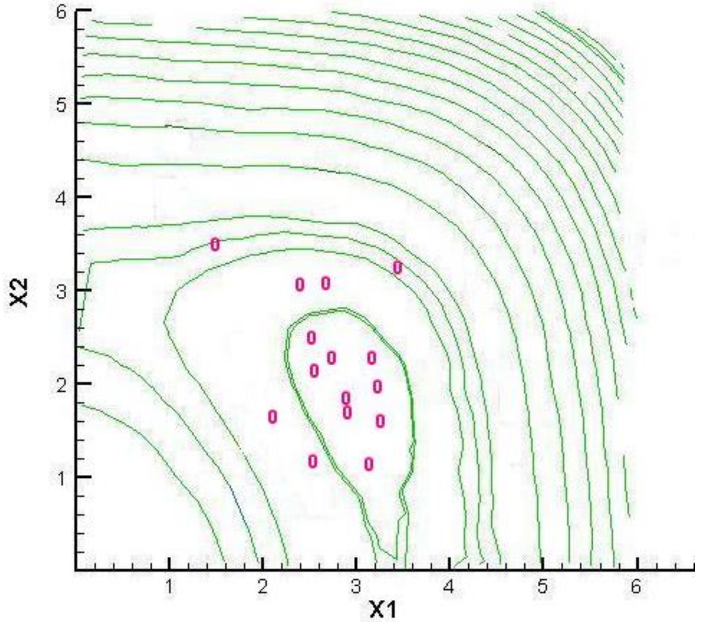


Generation 10

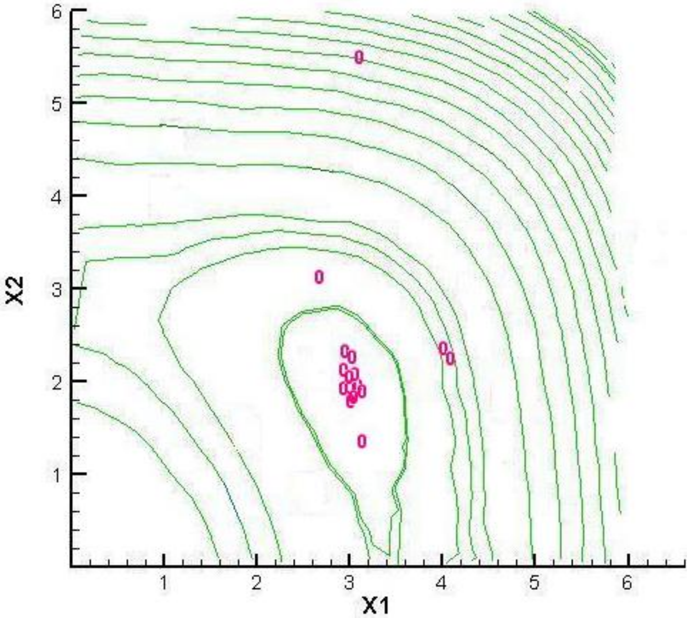
Genetic Algorithms



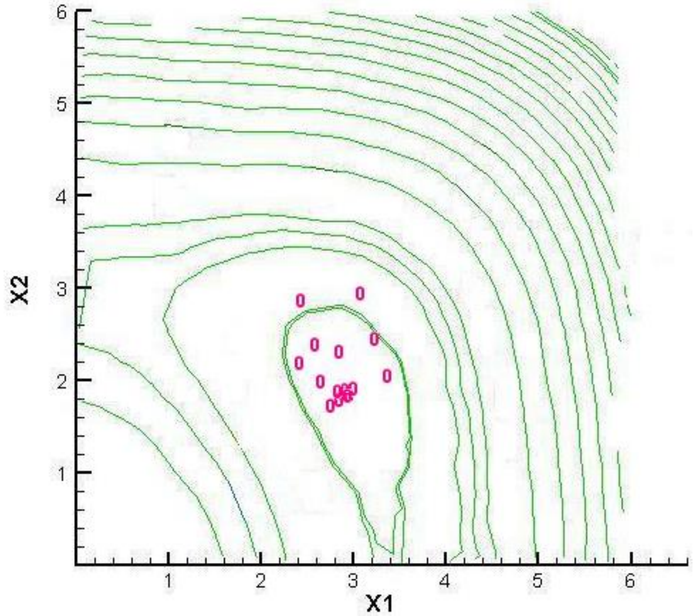
Generation 20



Generation 30



Generation 40



Generation 50