GENETIC ALGORITHM

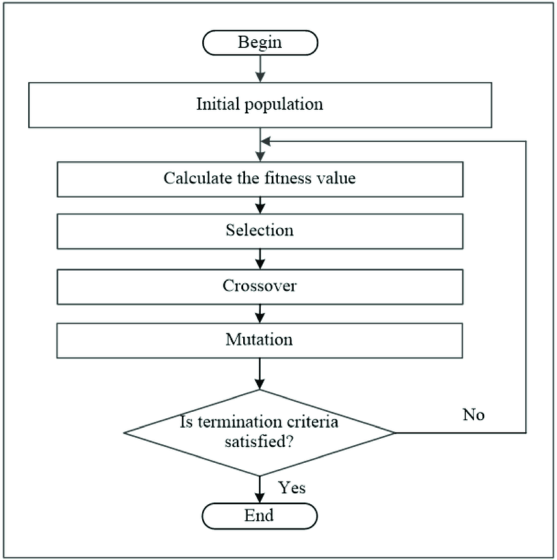
Dr. Sumanto Dutta
School of Computer Applications
Kalinga Institute of Industrial Technology (KIIT)

March 27, 2024

Contents

- Genetic Algorithm

Genetic Algorithm



Main features of Genetic Algorithms (GA)

Genetic Algorithms (GA) use principles of natural evolution. There are five important features of GA:

- **Encoding :** possible solutions of a problem are considered as individuals in a population. If the solutions can be divided into a series of small steps (building blocks), then these steps are represented by genes and a series of genes (a chromosome) will encode the whole solution. This way different solutions of a problem are represented in GA as chromosomes of individuals.
- **Fitness Function** represents the main requirements of the desired solution of a problem (i.e. cheapest price, shortest route, most compact arrangement, etc). This function calculates and returns the fitness of an individual solution.
- **Selection operator** defines the way individuals in the current population are selected for reproduction. There are many strategies for that (e.g. roulette–wheel, ranked, tournament selection, etc), but usually the individuals which are more fit are selected.
- **Crossover operator** defines how parents' chromosomes are mixed to obtain genetic codes of their offspring (e.g. one–point, two–point, uniform crossover, etc). This operator implements the inheritance property (offspring inherit genes of their parents).
- **Mutation operator** creates random changes in genetic codes of the offspring. This operator is needed to bring some random diversity into the genetic code. In some cases GA cannot find the optimal solution without mutation operator (local maximum problem).

```
Algorithm GA(Population,Fitness_Function)
repeat
    new_population :=   ;
    for i in 1 to size(Population) do
        Parent1 := SELECT-OPR(Population,Fitness_Function) ;
        Parent2 := SELECT-OPR(Population,Fitness_Function) ;
        Child := CROSSOVER(Parent1, Parent2) ;
        if P_c less than limit_probability then
            Child := MUTATION(Child) ;
        new_population.add(Child) ;
    Population := new_population ;
until some individual is fit enough, or total generation reached;
return the best individual in the Population, according to Fitness_Function
```

Example

Suppose a genetic algorithm uses chromosomes of the form x = abcdefgh with a fixed length of eight genes. Each gene can be any digit between 0 and 9. Let the fitness of individual x be calculated as:
$f(x) = (a + b)(c + d) + (e + f)(g + h)$
and let the initial population consist of four individuals with the following chromosomes: $x1 = 65413532$, $x2 = 87126601$, $x3 = 23921285$, $x4 = 41852094$

- Evaluate the fitness of each individual, showing all your workings, and arrange them in order with the fittest first and the least fit last.
- Perform the following crossover operations:
  - Cross the fittest two individuals using one–point crossover at the middle point.
  - Cross the second and third fittest individuals using a two–point crossover (points b and f).
  - Cross the first and third fittest individuals (ranked 1st and 3rd) using a uniform crossover.
- Suppose the new population consists of the six offspring individuals received by the crossover operations in the above question. Evaluate the fitness of the new population, showing all your workings. Has the overall fitness improved?

Thank You.