

Cloud Service Ranking and Selection

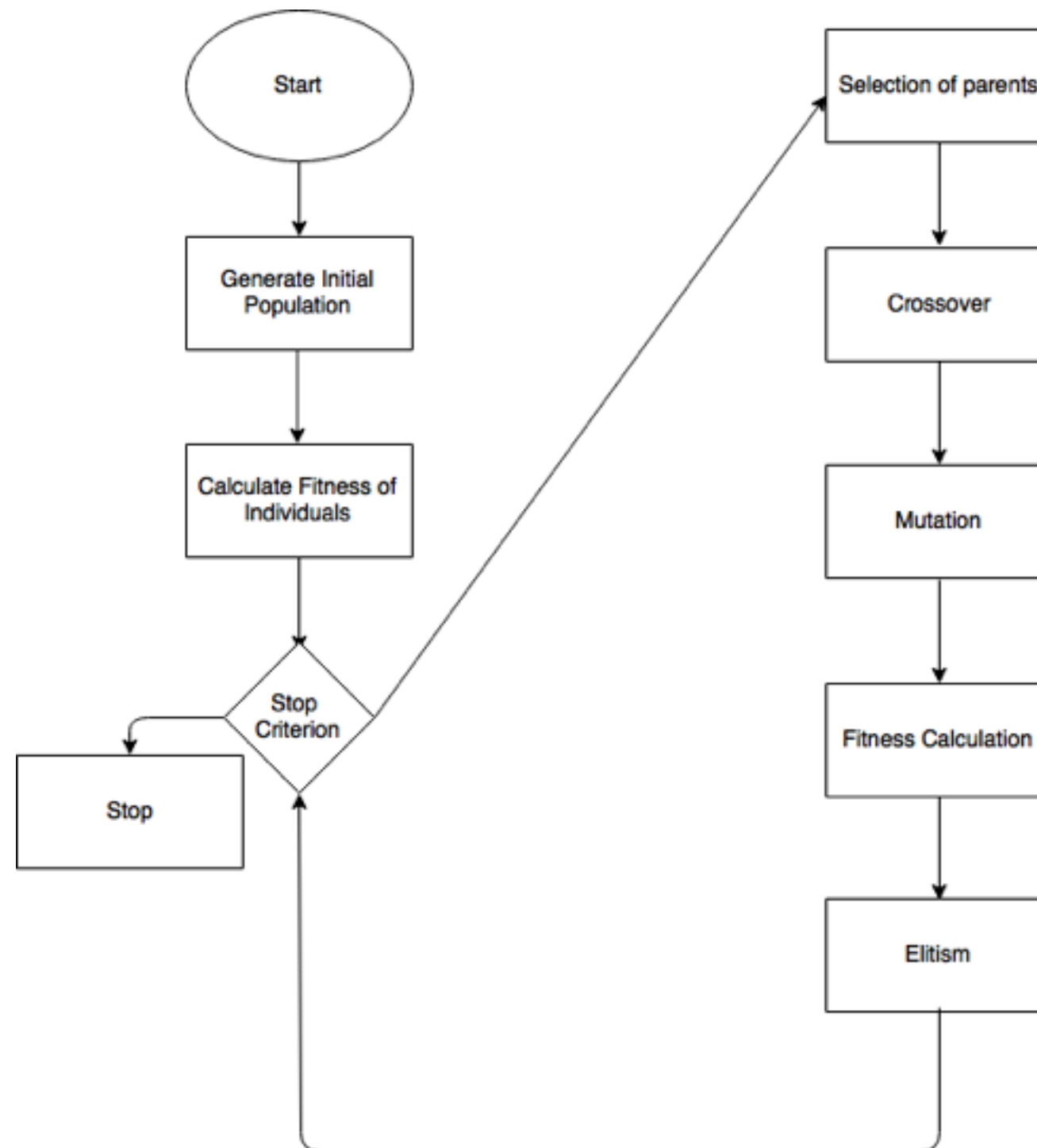
- A Genetic Algorithm Approach

Genetic Algorithm

In the field of artificial intelligence, a **genetic algorithm** (GA) is a search heuristic that mimics the process of natural selection.

This heuristic (also sometimes called a metaheuristic) is routinely used to generate useful solutions to **optimization and search problems**.

Working of the Algorithm



GA In Cloud Service Selection

- Known Details:
 - From the SLAs provided by each of the Cloud Service Providers, a table (ranking table) is formed.
 - This is based on the parameters that the different SLAs promise.

Sample Ranking Table

	Param1	Param2	Param3	Param4	Param5
CS1	3	1	3	3	-
CS2	2	-	1	-	2
CS3	4	2	-	4	1
CS4	-	3	2	1	-
CS5	1	4	-	2	3

'-' - No Rank
CS - Cloud Service
Param*i* - *i*'th parameter

Example Chromosome

- Each Cloud Service is represented using a chromosome (in GA) of the form $(p_1p_2p_3p_4p_5)$.
- Thus,
 - CS1 - 31330
 - CS2 - 20102
 - CS3 - 42041
 - CS4 - 03210
 - CS5 - 14023

where 0 represents no rank.

User Parameters

- User provides a set of parameters to be met (subset of 5 parameters in this case) with priority.

	Param1	Param2	Param3	Param4	Param5
User's Req	1	3	2	-	4

- This is taken in chromosome form $d_1d_2d_3d_4d_5$. (13204).
- 13204 is the desired chromosome from GA.

Fitness Function

- In GA, a fitness function is required to determine how close the generated chromosomes are to the desired one.
- The fitness function calculates the Euclidean Distance of the generated chromosome (from GA) with the desired chromosome $d_1d_2d_3d_4d_5$ (13204).
- For a generated chromosome $g_1g_2g_3g_4g_5$ (say 24501), the distance is calculated as follows :
 - $\sqrt{\sum (g_i - d_i)^2}$

In this example, $\sqrt{((2-1)^2 + (4-3)^2 + (5-2)^2 + (0-0)^2 + (1-4)^2)} = \sqrt{20}$

Fitness Function

- In this way, each of the newly generated chromosome is either discarded (distance is high) or chosen for the next generation (distance is low).
- Each of the chosen chromosome are now exposed to mutation and crossover each with its own probability of occurring.

Mutation

- A chromosome can go through mutation and the result is nothing but some values in the chromosome string changes (randomly).
- For example, a chromosome 23345 can undergo mutation to become 24342.

Crossover

- Two chromosomes that are chosen for the next generation can undergo crossover with a certain probability.
- For example, two chromosomes 23401 and 40123 can undergo crossover and form 23123.

Overall Picture

- Thus, with the above defined fitness function (using euclidean distance), probability for mutation and crossover the process is as follows.
- Start with a initial population of random chromosomes of cloud services.
 - Calculate the fitness function for each of them and discard those that are poor.
 - Allow the remaining ones to undergo mutation and crossover to end up with a new set of chromosomes.
 - Repeat the above steps a certain number of times or till only one chromosome remains.
 - This chromosome is the solution.
 - Finally, since this chromosome need not exactly match an existing cloud service, the cloud service closest (in Euclidean distance) is selected and recommended to the user.