5.IMPLEMENTATION

GA($Fitness$, $Fitness\_threshold$, p, r, m)

Fitness: A function that assigns an evaluation score, given a hypothesis.

$Fitness\_threshold$: A threshold specifying the termination criterion.

p: The number of hypotheses to be included in the population.

r: The fraction of the population to be replaced by Crossover at each step.

m: The mutation rate.

- *Initialize population:* $P \leftarrow$ Generate p hypotheses at random
- *Evaluate:* For each h in P, compute $Fitness(h)$
- While $[\max_h Fitness(h)] < Fitness\_threshold$ do

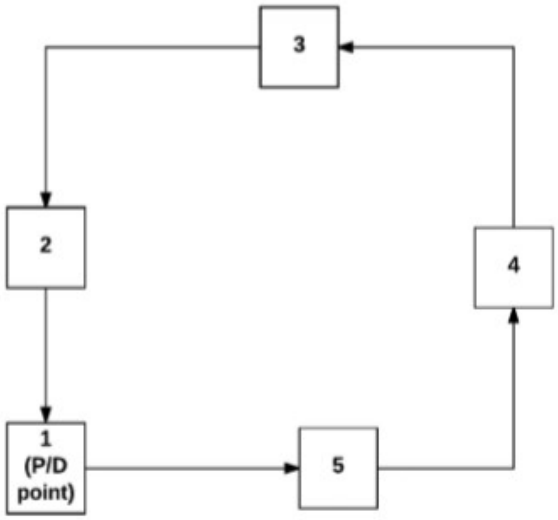
  Create a new generation, P_S:

  1. *Select:* Probabilistically select $(1-r)p$ members of P to add to P_S. The probability $\Pr(h_i)$ of selecting hypothesis h_i from P is given by

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

  2. *Crossover:* Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P, according to $\Pr(h_i)$ given above. For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by applying the Crossover operator. Add all offspring to P_s.
  3. *Mutate:* Choose m percent of the members of P_s with uniform probability. For each, invert one randomly selected bit in its representation.
  4. *Update:* $P \leftarrow P_s$.
  5. *Evaluate:* for each h in P, compute $Fitness(h)$
- Return the hypothesis from P that has the highest fitness.

5.1 Genetic Algorithm



5.2 Path

	1 (P/D point)	2	3	4	5
1(P/D point)	0	10	12	8	15
2	10	0	7	11	9
3	12	7	0	18	13
4	8	11	18	0	6
5	15	9	13	6	0

5.3 Sample shortest distance matrix

Selection: In this step, we determine the chromosomes that are required to be mated to produce the offsprings. These are determined with the help of fitnesses of each chromosome. The fitness is determined with the help of distances between each item in the warehouse.

Chromosome	Fitness score	Probability of being selected	Contiguous Intervals
15432	0.01785714	0.24029	[0 – 0.24029]
32451	0.01960784	0.26385	[0.24030 – 0.50414]
24513	0.01960784	0.26385	[0.50415 – 0.76799]
43521	0.01724138	0.23201	[0.76800 – 1]
Total	**0.07431421**	**1**	

5.4 Selection Probabilities

Fitness = 1/(Distance travelled)

If the fitness is less than the threshold we select them for further process.

Random number	Fallen mapped interval	Corresponding chromosome to be selected (Mating pool)
0.638425	[0.50415 – 0.76799]	24513
0.48754	[0.24030 – 0.50414]	32451
0.045572	[0 -0.24029]	15432
0.439298	[0.24030 – 0.50414]	32451

5.5 Mating Pool Generation

1. **Cross Over**: Crossover is combining pairs of parents to create new individuals or offspring. Two children are produced from two parents through a crossover operator which is a function that gets some of genes from one parent and the rest from the other parent to form two offsprings.

Chromosome pairs in mating pool	Crossover Probability	Random number	Selected for Mating?
(24513 & 32451)	0.7	0.214939833	0.214939833 <= 0.7 ➔ **Yes**
(15432 & 32451)	0.7	0.790672194	0.790672194 > 0.7 ➔ **No**

5.6 Crossover Probability on the Mating Pool

3.Mutation:

Mutation is introducing new genetic material in the population by randomly altering some selected genes of offspring chromosomes generated by crossover operation.

Mated Chromosomes	Mutation Probability	Random number	Selected for Mutating?
34521	0.01	0.394763129	0.394763129 > 0.01 ➔ **No**
52413	0.01	0.008672194	0.008672194 <= 0.01 ➔ **Yes**

5.7 Implementation of Mutation

From one generation to another, population size should remain fixed.

4.Reinsertion :

The difference between population size and the number of chromosomes reproduced through evolution is termed a generation gap.

Reinsertion mechanism is used to decide which chromosomes should survive and be inserted to the next generation, in case of a generation gap.

Chromosomes in mating pool	Fitness score	Rank of chromosome	Selected for Reinsertion?
24513	0.01960784	1	**Yes**
32451	0.01960784	1	**Yes**
15432	0.01785714	3	**No**
32451	001960784	1	**Yes (duplicated)**

5.8 Implementation of Re-Insertion

Having a new generation ready, GA evaluates each chromosome of the new generation using fitness function.