PERFORMANCE COMPARISON OF GENETIC AND SIMULATED ANNEALING ALGORITHMS APPLIED TO TRAVELLING SALESMAN PROBLEM

Wijayasundara R. and Suthakar S.
Department of Computer Science, Faculty of Science, University of Jaffna
rayanaradha1993gmail.com

SLSWCS Student Workshop on Computer Science

Abstract

The main intentions of this project are,

1. Analyzing the behavior on the Traveling Salesman Problem (TSP) of the two widely used nature-inspired heuristic approaches Genetic Algorithm(GA) and Simulated Annealing Algorithm(SA).
2. Analyzing the impact of parameter values of those two algorithms on different scenarios of TSP problems.

Different size test cases have been analyzed, and the Computation results show that when the number of cities in input increases SA provides better solutions compared to GA.

Data Set

This six standard TSP instances downloaded from the TSPLIB. [2]

Instances	berlien52	eli51	bier127	eli101	pr439	rat575
No. Of Cities	52	51	127	101	439	575
Optimal Solution	7542	426	118282	629	107217	6773

Table 01: Used data instances with a size and TSPLIB provided optimal solution

Methodology - Simulated Annealing Algorithm

- First set the initial temperature and generate a possible route for the TSP problem as the present solution.
- Then the neighbor solution generates from the present solution, by swapping the two cities on it. If the neighbor solution's total distance is less than our current solution's total distance accept it unconditionally. When the neighbor solution isn't better, needed to find the Acceptable probability.

Acceptable probability of neighbor solution = e (ΔE / T)

Where we can define ΔE as ΔE = evl (c) - evl (n)

evl (c) = Total travel distance of current solution.
evl (n) = Total travel distance of neighbor solution.

- We only calculate this acceptable probability when evl (c)< evl (n), which gives ΔE negative values which contain ranges of 0 and 1.
- Then compare this acceptable probability value with a random probability in every iteration of the algorithm and if the probability is high, accept the neighbor solution.
- Then we begin looping until the temperature value becomes less than 1. In every iteration temperature value is reduced according to a cooling rate.

Temperature = Temperature (1-coolingRate)

Methodology - Genetic Algorithm

- **Initial population:** Group of individuals which are possible routes (a route -different sequences of all the cities) to the tsp problem. (*Population Size = Size of the initial population*)
- **Fitness Function:** The fitness function determines how fit a solution is (evaluates how close a given solution is to the optimum solution). The total distance of a solution is calculated by adding the distances between the cities in the solution.

Fitness function = 1 / Total distance of a solution

- Selection: Two pairs of individuals (parents) are selected based on their fitness scores. First, select the subset from the initial population and then get two sequences with the highest fitness values from this subset. (*Tour Size = Size of the selected subset*)
- **Crossover:** In the crossover method needed to select a subset of cities from the first parent, and then add that subset to the offspring. Any missing values are then added to the offspring from the second parent in order that they are found.

Parent 1 :	1	5	8	9	4	2	7	3	3	Parent 2 :	9	2	3	6	5	4	8	1	7
Offspring :						2	7	3		→	9	6	5	4	8	2	7	3	1

- **Mutation Method:** Two locations in the route are selected at random then their positions are swapped when the randomly generated number is less than the predefined fixed number (*Mutation Rate*). Because swap mutation is only swapping pre-existing values, it will never create a list that has missing or duplicate values when compared to the original.

9	5	4	3	2	6	7	8	1

Before Mutation →

9	5	8	3	2	6	7	4	1

After Mutation

- **Termination:** The algorithm terminates when it's performed pre-defined fixed number (*population generation*) of new population generations.

Computational Result

Click to add text

Conducted the experiments 15 times for each instance using both SA and GA with and without Modified Parameters. Then compared the average distance of each instance with a TSPLIB provided optimal solutions.

Figure 01 : Average distance with different population size and mutation rate in GA

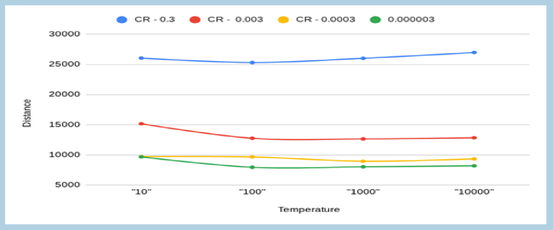

Figure 02 : Average distance with different Cooling Rate and Temperature in SA

Accuracy = (optimal solution / Average distance of 15 experiments) * 100 %

Attribute	Initial Parameters	Modified Parameters
Cooling Rate	0.015	0.000003
Temperature	1000	100

Table 02: SA Algorithm parameters values

Attribute	Initial Parameters	Modified Parameters
Mutation Rate	0.015	0.003
Tour size	5	10
Population Size	100	1000
Population generation	100	100

Table 03: GA Algorithm parameters values

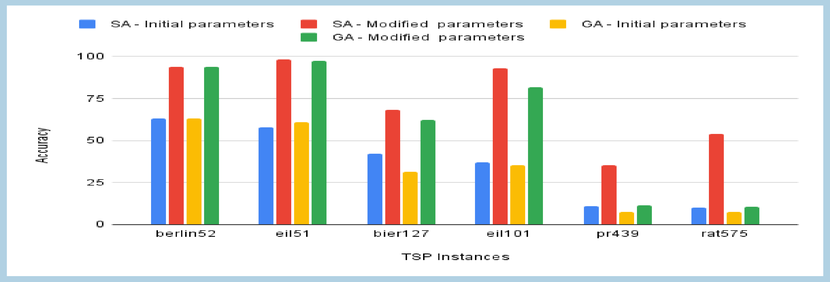

Figure 03 : Accuracy comparison between SA-IP, GA-IP, SA-MP and GA-MP

Discussion & Conclusion

- TSP can be summarized as there are cities and given costs, a traveling salesman is required to go to all cities with a minimum cost. This problem is considered as one of the most famous studied problems in optimization, because when the number of cities increases it becomes difficult to solve.
- In the best-case scenario, both algorithms after parameter tuning (Table 02 / Table 03 modified parameter values) provide optimal solutions for berlin52 and eli51 test cases.
- When increasing the population size GA algorithm converges into the solution which has high accuracy.
- Change of temperature value having less impact on the solution while decreasing cooling rate improved the performance of SA.
- SA provides better solutions compared to GA when the number of cities in input increases.
- Both algorithms can provide optimal solutions if the right set of parameters are set. So, it is needed to conduct a similar kind of experiments to different sizes of data sets to identify matching parameter values.

References

1. Rexhepi., A, Maxhuni., A, and Dika., A, Analysis of the impact of parameters values on the Genetic Algorithm for TSP, International Journal of Computer Science, 2013.
2. Data Set : http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
3. Adewole., A.P, Otubamowo., K, Egunjobi., T.O, A Comparative Study of Simulated Annealing and Genetic Algorithm for Solving the Travelling Salesman Problem, International Journal of Applied Information Systems, 2012.
4. Mukhairez., H, Maghari., A, Performance Comparison of Simulated Annealing, GA and ACO Applied to TSP, International Journal of Intelligent Computing Research, 2015.
5. Panda., M, Performance Comparison of Genetic Algorithm, Particle SwarmOptimization applied to TSP, International Journal of Applied Engineering Research, 2018.