Simulated Annealing and Genetic Algorithm

Comparison for the Traveling Salesman Problem

Ömer Bölükbaşı

omerbolukbasi84@gmail.com

Department of Computer Science

Marmara University

Abstract

This paper contains the implementation of Simulated Annealing (SA) and Genetic Algorithm (GA) on Travelling Salesman problem. In the experiments, the effect of different values of cooling rate and number of iterations in each temperature steps on the SA performance is observed. Results show that with slow annealing performance is higher in SA. Also in fair conditions SA performs better than GA.

1. INTRODUCTION

Travelling Salesman Problem (TSP) is an old NP Complete Problem that has been studied for decades in Computer Science researches. The problem consists of finding the best route over a certain number of cities. The route must include each and every city just one times and ends at the node where it starts.

In this research, the performance effect of cooling rate and the iteration frequency in temperature steps are observed within the experiment 1. And in experiment 2, a GA algorithm is implemented on the same problem. And the performance comparison is done with SA.

2. SIMULATED ANNEALING ALGORITHM

Simulated Annealing(SA) mimics the physical annealing process. In the physical process material is heated and slowly cooled towards a strong crystalline structure instead of metastable states. [1]

In SA, for a calculated probability the solutions worse than the best solutions are also accepted. This property is used to explore different areas in search space. In addition, the solutions better than the best solution are also accepted. The acceptance probability calculation is based on T: Existing Temperature, $eval(v_n)$: fitness of the existing individual, $eval(v_c)$: fitness of the child individual.

$$P(\Delta E) = e^{\frac{eval(v_n) - eval(v_c)}{T}} = e^{\frac{-\Delta E}{T}}$$

In the beginning of the search T is large. If T is large then the probability of accepting worse solution is also high which yields the algorithm acts as a random search (Accepts most of the incoming solutions). In the end of the search T is small. If T is small then SA acts as a local search because it accepts mostly the best solutions.

Temperature decrements whether linear:

$$T_i = T_i - i \times \beta$$

or geometric:

$$T = \alpha \times T \qquad \alpha \in [0,1]$$

3. EXPERIMENTS

3.1 Performance of SA With Respect to Different Cooling Schedules

In this experiment two SA cases will be considered, where the first one has a cooling rate of 0.95 with 10 iterations at each temperature level, while the second case considers a cooling rate of 0.995 with 2 iterations at each temperature level. Run each case 100 times and present the results (the best result, average result, and running time) of each case. Same randomly generated initial solution are selected to compare two cases.

3.1.1 Simulation Parameters

- String Representation: We will consider permutation based representation as in the first project. The initial solution is set randomly.
- Neighbor Generation Strategy: In this project, a neighbor of the current solution is generated by selecting two random positions in permutation encoding representation scheme and swapping the cities of these positions.
- Initial Temperature: It is set to 10000, unless otherwise stated.
- Cooling Schedule: Geometric schedule is considered and cooling rate is equal to 0.95, unless otherwise stated.
- Equilibrium State: To reach an equilibrium state at each temperature, 10 moves must be applied.
- Stopping Condition: The stopping criteria is to reach a final temperature of 0.01.
- Best Solution Preserved. Note that best solution up to now will be preserved, which is not part of SA algorithm.
- Outputs: Best result after termination, total number of iterations (neighbor generations) and running time will be displayed.

3.1.2 Simulation Results on kroA100.tsp

The first part of experiment 1 is done with the input topology of kroA100 which has 100 city and their locations. In Case 1, cooling rate is set to 0.95 and the moves per temperature is set to 10. The total run time, best result and average result are measured as in the following:

```
Case 1, Total Run Time: 6, Best Result: 57281, Average Result:  63395.5
```

In Case 2, the cooling rate is set to 0.995 and the moves per temperature is set to 2. The results are a s follows:

```
Case 2, Total Run Time: 11, Best Result: 53053, Average Result: 53351.5
```

3.1.3 Simulation Results on a280.tsp

The second part of experiment 1 is done with the input topology of a280 which has 280 city and their locations. In Case 1, cooling rate is set to 0.95 and the moves per temperature is set to 10. The total run time, best result and average result are measured as in the following:

```
Case 1, Total Run Time: 717, Best Result: 15376, Average Result: 16790
```

In Case 2, the cooling rate is set to 0.995 and the moves per temperature is set to 2. The results are a s follows:

```
Case 2, Total Run Time: 1556, Best Result: 12697, Average Result: 14157
```

3.2 Performance Comparison of SA with GA

In this case, the results of GA and SA are compared with a fair comparison. A fair comparison can be achieved by providing equal time for both cases.

3.2.1 Simulation Parameters

For SA the parameter in Experiment 1 and Case 2 are used. For GA the parameters are as follows:

Population initialization: Random

Population size: 50

Mutation: Inversion mutation (IVM)

Recombination: Order Crossover (OX)

Parent Selection: Tournament selection with a tournament size of 5.

Survival Selection: Steady-state GA. The offsprings will be written in place of two worst individuals of the population.

3.2.2 Simulation Results on kroA100.tsp

The first part of experiment 2 is done with the input topology of kroA100 which has 100 city and their locations. To be a fair comperation, the total run time must be similar. So, in order to get similar run times, generation count of GA is adjusted to 2970. Results are as follows:

```
Case GA, Total Run Time:  331 , Best Result:  51432 , Average Result: 
58817.98
Case SA, Total Run Time:  354 , Best Result:  43614 , Average Result: 
53241.13
```

3.2.3 Simulation Results on a280.tsp

The second part of experiment 2 is done with the input topology of a280 which has 100 city and their locations. To be a fair comparetion, the total run time must be similar. So, in order to get similar run times, generation count of GA is adjusted to 2274. Results are as follows:

```
Case GA, Total Run Time:  1033 , Best Result:  16300 , Average Result:
17669.23
Case SA, Total Run Time:  1025 , Best Result:  12697 , Average Result:
14157.32
```

4. CONCLUSION

In experiment 1, it is observed that when the cooling rate is 0.995, Case 2 (instead of 0.95) and moves per temperature is 2 (instead of 10), the best result and the average result is better. In Case 2, cooling rate and the speed of iterations are slow. Since it is slow, algorithm can find opportunity to search in high temperatures. In high temperatures poorer solutions can be accepted with high probability. So, the exploration process is longer and the probability of discovering a global optimum is high. The simulation results are also in the same manner.

In experiment 2, within the same duration, in fair conditions SA performs better in both city topologies which shows that the exploration/exploitation balance is much better in SA in comparison to GA.