Data Structures

Homework 4. Genetic Algorithm

Ali Mazlumov

20130841

1. **Encoding**. First of all, I have changed the encoding scheme from Position based encoding to K-array encoding. When I changed encoding, I had to revise how Initial Population is drawn. So there is no more need for checking feasibility of initial population, because genotype generated is random permutation of numbers, so there will not be any duplications (same numbers in one array).
2. **Fitness**. Fitness is calculated in pretty much the same way – inverse of the distance.
3. **Select Parents**. Selection of parents will be the same – ranking based selection.
4. **Crossover**. Crossover is the Key Component in my algorithm. For that purpose, I have referenced [1]. This paper proposed the new algorithm for crossover which is called – Greedy Subtour Crossover (GSX). They have showed by experiments that solution for TSP can popup from local minima more effectively than SA (simulated annealing). So I just decided to try out this algorithm. I think that algorithm works pretty fast and more accurate than the original crossover, it was also very good at exploration phase (escaping from local minima).
5. **2 opt Swap**

2optSwap(route, i, k) {

1. take route[0] to route[i-1] and add them in order to new\_route

2. take route[i] to route[k] and add them in reverse order to new\_route

3. take route[k+1] to end and add them in order to new\_route

return new\_route;

}

1. **Conclusion**. My modified GA with GSX works a lot faster due to optimization of calculation phase as well as crossover stage (GSX).
2. **References**

[1] Sengoku, Hiroaki, and Ikuo Yoshihara. "A fast TSP solver using GA on JAVA." Third International Symposium on Artificial Life, and Robotics (AROB III’98). 1998.

[Attached PDF File]

[2] https://en.wikipedia.org/wiki/2-opt



