Alptekin Orbay 2017700090
Gözde Berk 2017700027

# IE-517 HEURISTIC METHODS IN OPTIMIZATION

## Final Project Report

*Solving the Traveling Salesman Problem with Profits*

In this project, we choose simulated annealing (SA) as a meta-heuristic method. The solution representation is permutation representation. It starts with 1 and ends with 1 again as it is a tour. 1 is the depot. The other elements are customer ids between two 1s without repetition. Since the number of customers that must be visited is not fixed , our method must find the optimal number of customers visited leading to the optimal objective value. So, SA is applied in two different way. In our method, a outer SA is designed to find the optimal customers that gives the best results. So, the neighbourhood structure is inserting a element, deleting a element or exchanging a element with one that are not involved in the current solution. To compare the candidate solutions, the optimal values of each neighbourhood must be found. For this purpose, a inner SA is applied again. Permutation representation is used as solution representation and 2-exchange is used as neighbour structure. Inner SA is used to find the best solution with this configuration that is the set of chosen cities for the current situation.

Briefly, our algorithms first generate an initial solution. For low profits, the initial solution includes less customers and for high profits, the initial solution starts with more customers. Then, the initial solution is optimized with inner SA. Then, with same probability, a solution is proposed with insertion, deletion or exchange. Also, for intensification, with same probability with others, the current solution is maximized with inner SA. Insertion and deletion enables to find the optimal number of customers. Exchange enables to find the best customers with restriction of customer number. Note that, an insertion and a deletion move gives the same effect, but directly using exchange can make jumps that an insertion and a deletion cannot. By the way, insertion and deletion works randomly. They do not depend on the profits to increase diversification.

For hyperparameters like number of epochs, initial and final temperature, we do several experiments. The higher number of epochs improve results very little relatively. However, for each dataset, the number of epochs are different. For larger dataset, the combinations are much higher, so we arrange the epoch number higher. It results higher CPU times. Though, the CPU time are not increased exponentially as combinations increases exponentially.

Alptekin Orbay 2017700090
Gözde Berk 2017700027

| Instance | Best Value | No. | Sequence of customers visited | CPU Time (s) |
|---|---|---|---|---|
| eil51-LP | 27,28 | 18 | [ 1, 32, 11, 16, 50, 9, 49, 10, 33, 45, 15, 37, 17, 4, 47, 12, 46, 51, 27, 1] | 368 |
| eil51-HP | 706.84 | 49 | [ 1, 22, 8, 26, 31, 28, 3, 36, 35, 20, 29, 2, 16, 50, 21, 34, 30, 9, 49, 10, 39, 33, 45, 15, 44, 42, 19, 41, 13, 25, 14, 24, 43, 7, 23, 48, 6, 27, 51, 46, 12, 47, 18, 4, 17, 37, 5, 38, 11, 32, 1] | 201 |
| eil76-LP | 115.34 | 59 | [ 1, 73, 33, 63, 16, 51, 17, 40, 12, 26, 67, 76, 75, 4, 68, 6, 2, 62, 28, 74, 30, 48, 47, 36, 71, 60, 70, 20, 37, 5, 29, 45, 27, 52, 34, 46, 8, 35, 53, 11, 66, 65, 38, 10, 58, 72, 39, 9, 25, 50, 32, 44, 3, 24, 49, 23, 56, 41, 42, 43, 1,] | 531 |
| eil76-HP | 1246.09 | 74 | [ 1, 43, 42, 64, 41, 56, 23, 49, 24, 18, 50, 55, 25, 9, 39, 72, 58, 10, 38, 65, 66, 11, 59, 14, 53, 7, 35, 8, 19, 54, 13, 57, 15, 5, 37, 20, 70, 60, 71, 69, 36, 47, 48, 21, 61, 22, 62, 28, 74, 2, 30, 4, 45, 29, 27, 52, 46, 34, 67, 26, 76, 75, 68, 6, 51, 17, 12, 40, 32, 44, 3, 16, 63, 33, 73, 1] | 258 |
| eil101-LP | 187.42 | 76 | [ 1, 69, 52, 18, 60, 83, 8, 45, 17, 84, 5, 61, 16, 86, 44, 14, 42, 57, 2, 87, 97, 95, 92, 37, 98, 100, 91, 85, 93, 59, 99, 96, 6, 94, 13, 58, 53, 101, 27, 28, 26, 40, 21, 73, 72, 74, 22, 75, 56, 39, 25, 55, 54, 80, 68, 77, 3, 79, 33, 81, 78, 34, 35, 71, 66, 20, 32, 90, 10, 62, 19, 47, 48, 7, 88, 31, 70, 1] | 1133 |
| eil101-HP | 1626.72 | 100 | [ 1, 70, 30, 32, 90, 63, 64, 49, 36, 47, 19, 11, 62, 10, 31, 88, 7, 82, 48, 46, 8, 45, 17, 84, 5, 60, 83, 18, 52, 89, 6, 96, 99, 93, 59, 92, 98, 37, 100, 91, 85, 61, 16, 86, 38, 44, 14, 42, 43, 15, 57, 2, 87, 97, 95, 94, 13, 58, 40, 26, 12, 54, 4, 21, 73, 72, 74, 22, 41, 75, 56, 23, 67, 39, 25, 55, 24, 29, 80, 68, 77, 3, 79, 78, 34, 35, 71, 65, 66, 20, 51, 9, 81, 33, 50, 76, 28, 53, 101, 27, 69, 1] | 1087 |