Assignment 6: Multiple start local search (MSLS) and iterated local search (ILS)

Authors: Kiril Andrukh, 162069; Uladzislau Lukashevich, 155671.
Source code: link

Description of the problem

Multiple start local search (MSLS) and iterated local search (ILS)

The goal of the task is to implement two simple extensions of local search:

Multiple start local search (MSLS) – we will use steepest local search starting from random solutions. Iterated local search (ILS).

You can use basic steepest local search with edge exchange as intra-route move or version with list of moves (if it was implemented successfully). The perturbation for ILS should be designed by you and precisely described in the report. You should aim at obtaining better results with ILS than with MSLS.

Computational experiment: Run each of the methods (MSLS and ILS) 20 times for each instance. In MSLS perform 200 iterations of basic local search. For ILS as the stopping condition use the average running time of MSLS. For ILS report also the number of runs of basic local search. For ILS as the starting solution (one for each run of ILS) use random solution. Note that the results of a single run of MSLS is the best solution among a given number of runs of local search.

Reporting results: Use tables as in the previous assignment. For ILS add a table with the number of runs of basic LS. Report the best solutions for each instance as a lists of nodes. Include results of the best construction heuristics(s) and the best version(s) of local search.

The outline of the report as previously.

Multiple Start Local Search

Pseudocode:

```
    Initialize empty ratings list

    For each run in range(num_runs):
        Generate random initial solution
        Apply local search to solution with steepest strategy and edge-based
intra-search
        Add solution and its cost to ratings

    Find and return the best solution (lowest cost) from ratings
```

Iterated Local Search

Pseudocode:

```
Perturbation function:
     Get solution as input
     Set number of exchanges to 5

     For each exchange:
         Select random start position for first 3-node segment
         Select random start position for second non-overlapping 3-node segment
         If valid positions exist:
             Extract both segments
             Swap their positions based on which comes first
     Return perturbed solution


ILS:
     Initialize empty ratings list

     For each run in range(num_runs):
         Generate random initial solution and apply local search to it
         Perturbate the generated solution using perturbation function

    Perform local search starting from perturbed solution
    If the solution obtained from Local Search is not worse than the current best:
        Set the solution as the current best

    Find and return the best solution (lowest cost) from ratings
```

Function performance

Method	Dataset A	Dataset B
Random generation, edge, steepest	73855.835(70939-77610)	48296.625(45319-50992)
Multiple start local search	72010.48(70553-72972)	46477.23(45212-47381)
Iterated local search	70797.655(69875-72440)	45494.965(44070-47548)

Average running time

Method	Dataset A	Dataset B
Random generation, edge, steepest	3.3 s	3.17 s
Multiple start local search	46.43 s	46.24 s
Iterated local search	46.42 s	46.23 s

Number of runs

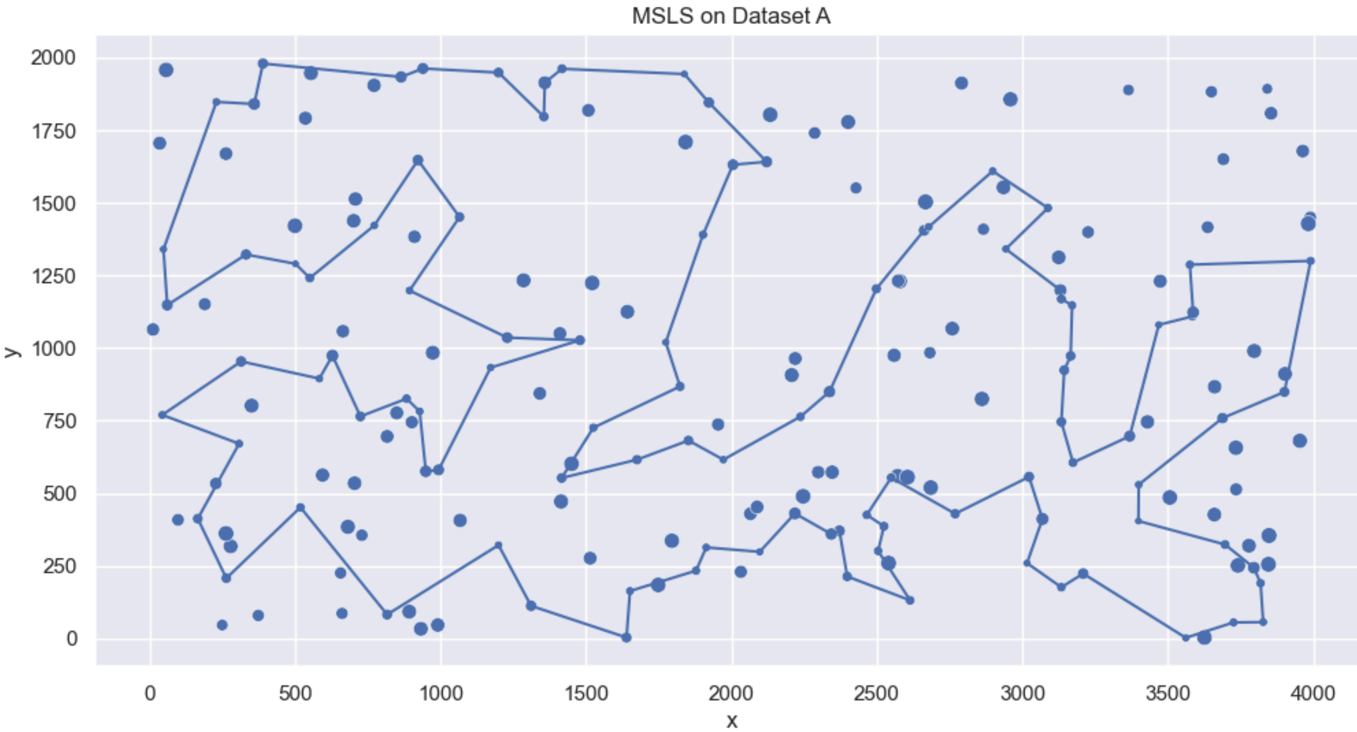
Method	Dataset A	Dataset B
Multiple start local search	200(200-200)	200(200-200)
Iterated local search	642.54(624-658)	645.46(593-666)

Plots

Multiple Start Local Search

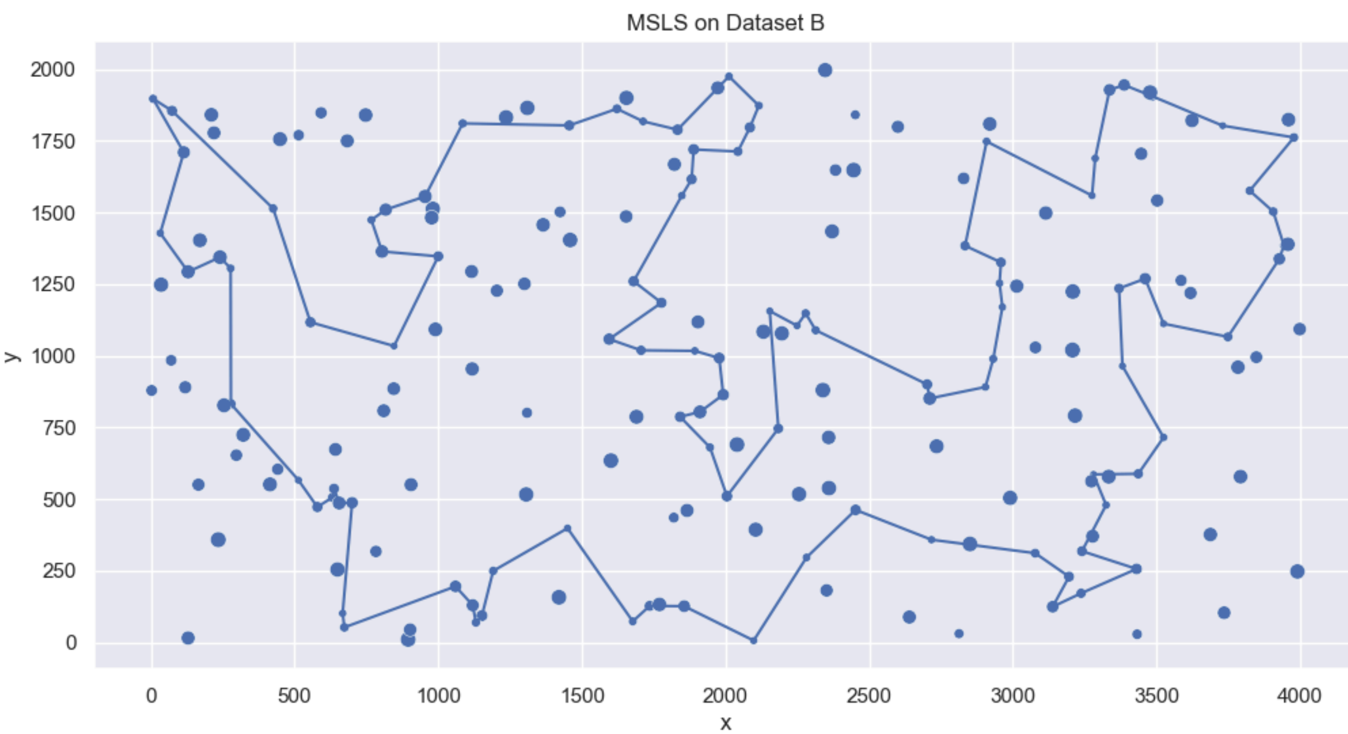
Dataset A:

Best solution: [55, 57, 92, 179, 185, 40, 119, 165, 90, 81, 196, 145, 78, 31, 56, 113, 175, 171, 16, 25, 44, 120, 82, 129, 2, 152, 97, 1, 101, 75, 86, 26, 100, 121, 53, 180, 154, 135, 70, 127, 123, 112, 184, 177, 54, 160, 34, 181, 42, 5, 43, 116, 65, 131, 149, 59, 51, 118, 115, 46, 68, 139, 41, 193, 159, 146, 22, 18, 69, 108, 140, 93, 117, 0, 170, 143, 183, 89, 186, 23, 137, 176, 80, 151, 162, 133, 79, 63, 94, 124, 148, 9, 62, 144, 14, 49, 3, 178, 106, 52]



Dataset B:

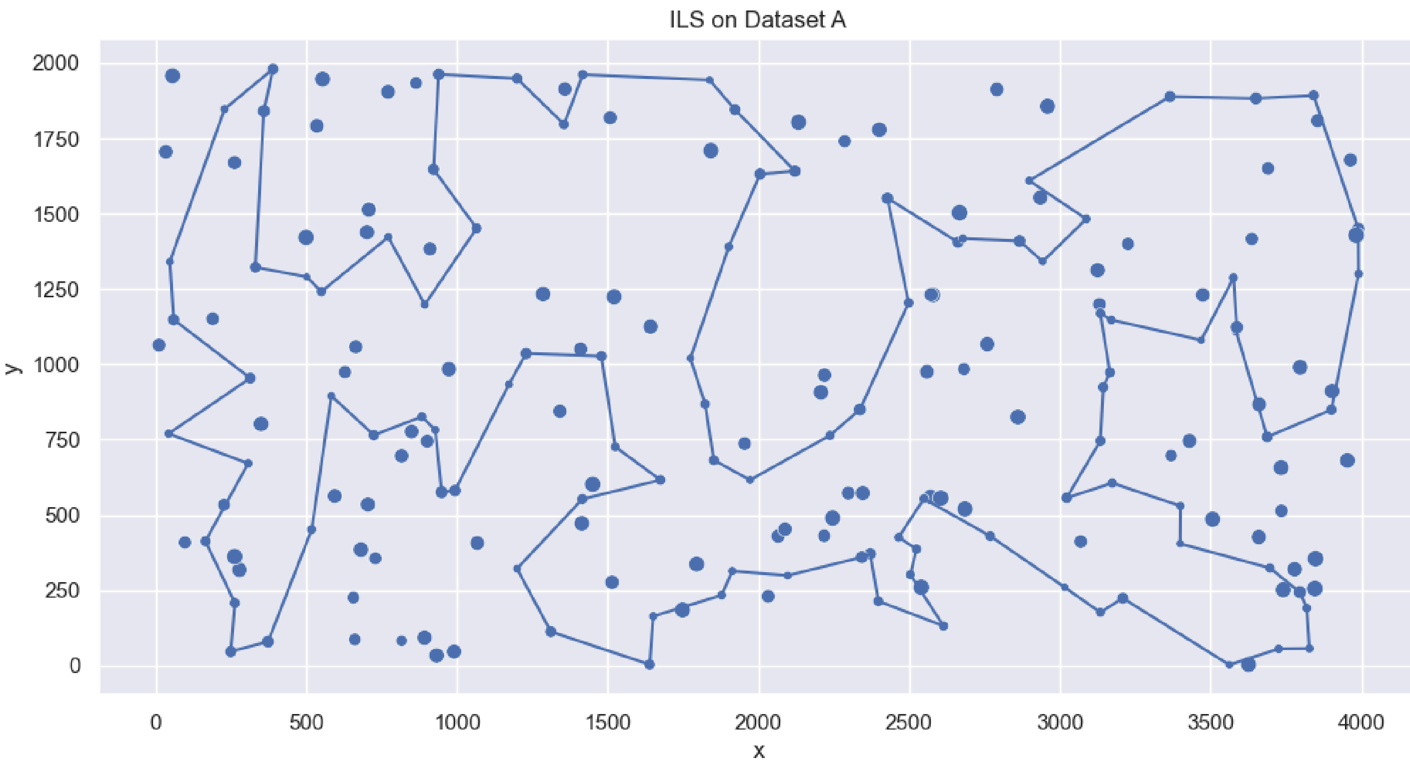
Best solution: [77, 141, 91, 61, 36, 177, 5, 78, 175, 142, 45, 80, 190, 73, 164, 54, 31, 193, 117, 1, 135, 32, 102, 63, 100, 40, 107, 122, 131, 121, 51, 125, 90, 191, 71, 147, 6, 188, 169, 132, 70, 3, 15, 145, 13, 195, 168, 139, 11, 182, 138, 33, 160, 144, 56, 104, 8, 82, 111, 29, 0, 109, 35, 143, 159, 106, 124, 62, 18, 55, 34, 152, 183, 140, 4, 149, 28, 20, 60, 148, 47, 94, 66, 179, 185, 130, 95, 86, 166, 194, 176, 113, 103, 114, 127, 165, 89, 163, 153, 81]



Iterated Local Search

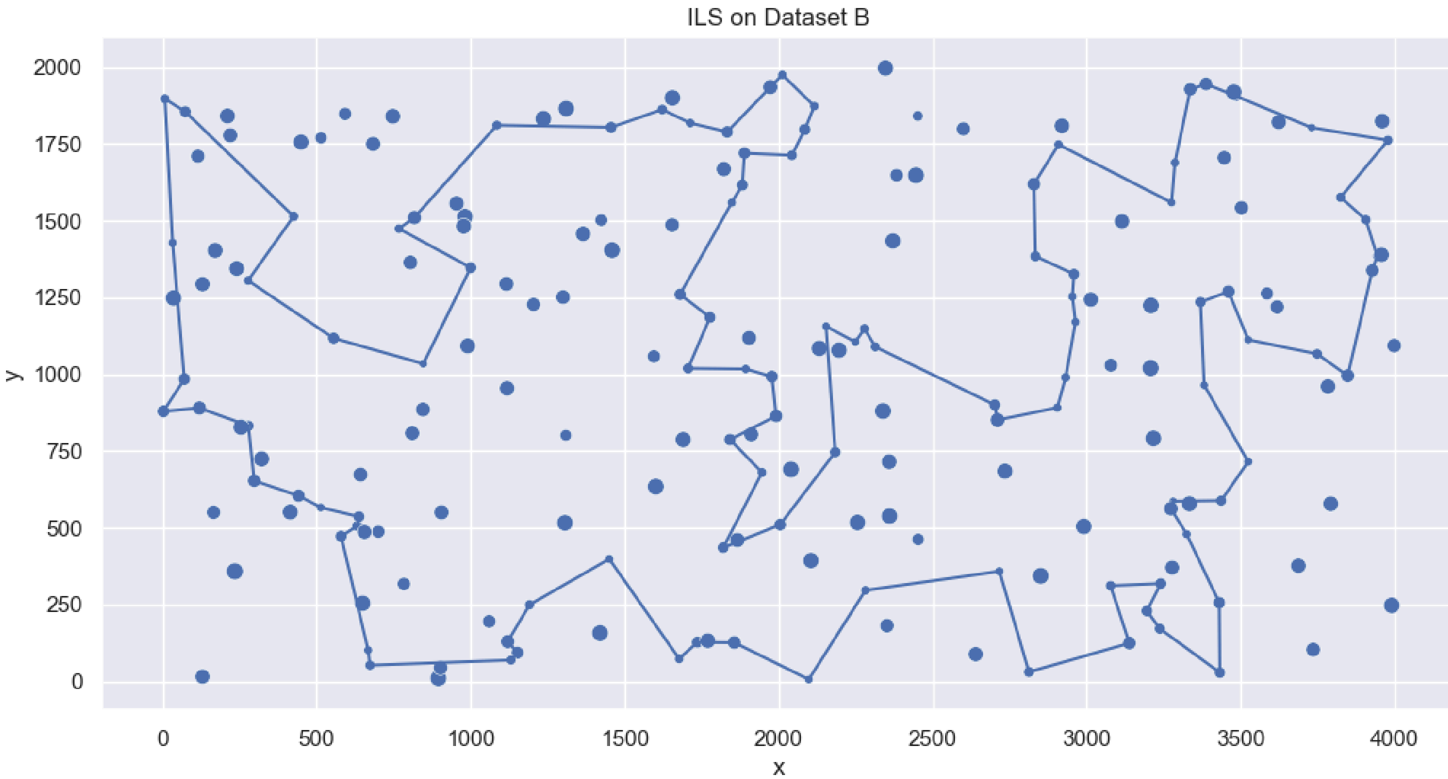
Dataset A:

Best solution: [116, 65, 131, 149, 59, 118, 51, 151, 133, 162, 123, 127, 70, 135, 154, 180, 53, 100, 26, 86, 75, 101, 1, 97, 152, 2, 120, 44, 25, 16, 171, 175, 113, 56, 31, 78, 145, 92, 129, 57, 55, 52, 178, 106, 185, 165, 119, 40, 196, 81, 90, 27, 164, 7, 21, 144, 14, 49, 102, 62, 9, 15, 148, 124, 94, 63, 79, 80, 176, 137, 23, 186, 89, 183, 143, 0, 117, 93, 68, 46, 115, 139, 41, 193, 159, 69, 108, 18, 22, 146, 181, 34, 160, 54, 177, 10, 190, 184, 42, 43]



Dataset B:

Best solution: [35, 143, 159, 106, 124, 62, 18, 55, 34, 170, 152, 183, 140, 4, 149, 28, 20, 60, 148, 47, 94, 66, 172, 179, 185, 130, 95, 86, 166, 194, 176, 180, 113, 114, 137, 127, 89, 103, 163, 165, 187, 153, 77, 141, 91, 61, 36, 177, 5, 142, 78, 175, 80, 190, 193, 31, 54, 117, 198, 156, 1, 16, 27, 38, 63, 40, 107, 122, 135, 131, 121, 51, 90, 191, 147, 6, 188, 169, 132, 70, 3, 15, 145, 13, 195, 168, 139, 11, 138, 33, 160, 144, 104, 8, 21, 82, 111, 29, 0, 109]



Conclusion:

On average MSLS is better than LS. ILS outperforms all other algorithms. Smaller perturbations lead to faster running times, and leads to more solutions being explored.