Evolutionary Computation - lab assignment 6

Szymon Bujowski, 148050, source: https://github.com/bujowskis/put-evolutionary-computation

Problem description

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.

Implemented algorithms and pseudocodes

Notes

- 50% was the required number of nodes in a solution, and “required number of nodes” will be referred to this way in the pseudocode
- “**Total move cost**” is the total change in objective function after adding a node, that is, its edge distance and additional cost
- Pseudocode is simplified to the required configuration of Local Search - that is: Steepest LS, Random starting solution, two edges exchange intra moves
- LS with candidate moves was used

Multiple Start Local Search (MSLS)

1. Generate 200 Local Search solutions (starting from random solution)
2. Return the solution with the lowest objective function value

Iterated Local Search (ILS)

1. Generate a Local Search solution (starting from random solution) - remember it as the best solution known so far
2. While total running time of the algorithm is less than that of MSLS
2.1. Generate a perturbed solution - perturbation is a random segment exchange, repeated 5 times
2.1.1. Select a random segment of 3 consecutive nodes from the best solution known so far
2.1.2. Select a random segment of 3 consecutive nodes from the best solution known so far, such that it does not overlap segment from step 2.1.1.
2.1.3. Insert segment from step 2.1.1. in place of segment from step 2.1.2., and vice versa
2.1.3.Ad.1. Suppose the best solution known so far is [1,**2,3,4**,5,6,**7,8,9**], and segments [**2,3,4**], [**7,8,9**] are selected - then the perturbed solution after 1 random segment exchange is [1,**7,8,9**,5,6,**2,3,4**]
2.2. Perform Local Search starting from the perturbed solution
2.3. If the result from step 2.2. has lower objective function, update the best solution known so far
3. Return the best solution known so far

Results

Note: All best solutions were checked with the solution checker

Statistics

Method	**TSPA** avg (min - max)	**TSPB** avg (min - max)
NN at any	73173 (71179 - 75450)	45870 (44417 - 53438)
Local Search (steepest, greedy, two_edges)	71998 (70110 - 74270)	45171 (44140 - 51146)
Local Search (steepest, random, two_edges)	74222 (71725 - 78624)	48627 (45619 - 52245)
Multiple Start Local Search (MSLS)	72324 (71458 - 72825)	46801 (45825 - 47480)
Iterated Local Search (ILS)	69498 (69259 - 70321)	43858 (43568 - 44484)

Running times (s)

Method	**TSPA** avg (min - max)	**TSPB** avg (min - max)
NN at any	0.15 (0.14 - 0.19)	0.16 (0.15 - 0.53)
Local Search (steepest, greedy, two_edges)	0.08 (0.07 - 0.08)	0.08 (0.07 - 0.08
Local Search (steepest, random, two_edges)	0.22 (0.18 - 0.42)	0.22 (0.17 - 0.32)
Multiple Start Local Search (MSLS)	47.48 (42.77 - 53.01)	45.31 (44.61 - 45.89)
Iterated Local Search (ILS)	47.37 (47.26 - 47.51)	47.39 (47.26 - 47.54)

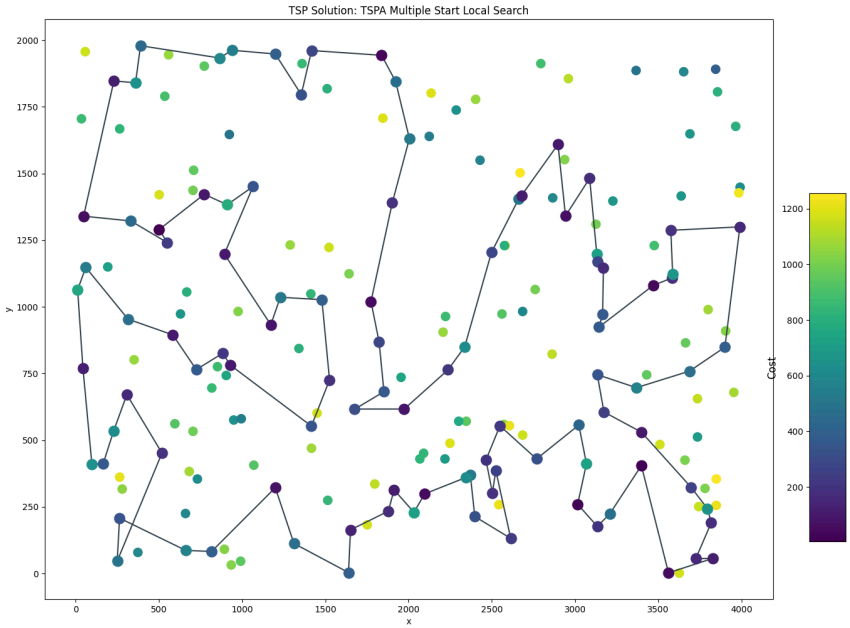
Number of Local Search runs

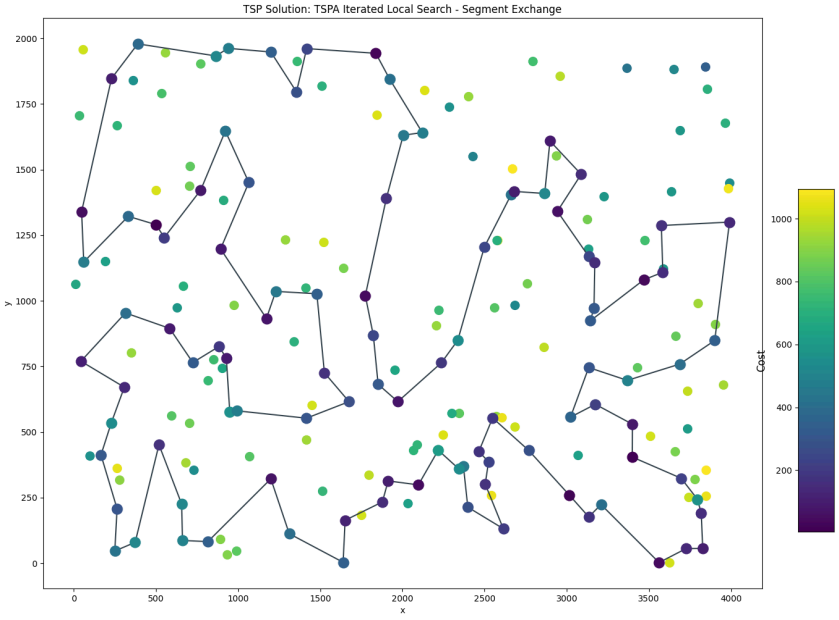
Method	**TSPA** avg (min - max)	**TSPB** avg (min - max)
Multiple Start Local Search (MSLS)	200 (200 - 200)	200 (200 - 200)
Iterated Local Search (ILS)	901 (815 - 951)	901 (875 - 940)

Best solutions

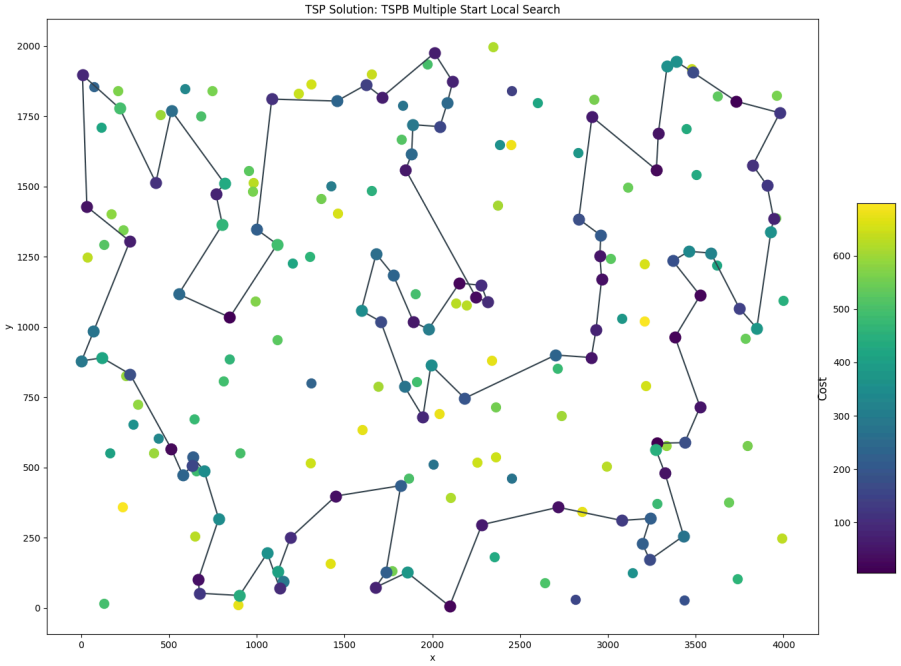
Note: additional cost is depicted using a color scale

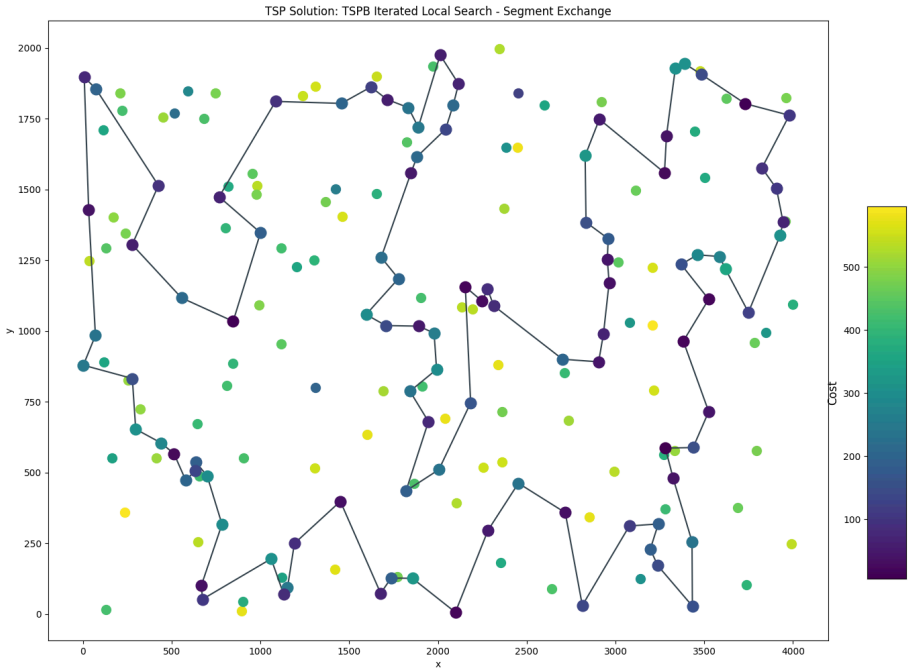
TSPA


TSP Solution: TSPA Multiple Start Local Search
x
y
Cost


TSP Solution: TSPA Iterated Local Search - Segment Exchange
x
y
Cost

TSPB


TSP Solution: TSPB Multiple Start Local Search
y
x
Cost


TSP Solution: TSPB Iterated Local Search - Segment Exchange
y
x
Cost

Conclusions

- On average, MSLS is still worse than LS in configuration (steepest, greedy, two_edges)
- ILS ran for the same duration as MSLS outperforms all considered methods by a significant margin
- Small perturbations from local optima:
    - Enable ILS to overcome the limitation of LS and “fix” problematic segments of the solution
    - Lead to faster running times of LS, because most of the solution is “already in its local optimum” - consequently, significantly more solutions can be explored compared to MSLS (avg 901 vs 200)