06-MSLS-ILS

December 8, 2024

Made by: Andrei Kulchyk (155489) and Fiodar Piatrovich (155174)

Github

1 Description of a problem

We are given three columns of integers with a row for each node. The first two columns contain x and y coordinates of the node positions in a plane. The third column contains node costs. The goal is to select exactly 50% of the nodes (if the number of nodes is odd we round the number of nodes to be selected up) and form a Hamiltonian cycle (closed path) through this set of nodes such that the sum of the total length of the path plus the total cost of the selected nodes is minimized.

The distances between nodes are calculated as Euclidean distances rounded mathematically to integer values. The distance matrix should be calculated just after reading an instance and then only the distance matrix (no nodes coordinates) should be accessed by optimization methods to allow instances defined only by distance matrices.

2 MSLS

2.1 Algortithm's pseudocode

Algorithm: Multi-Start Local Search (MSLS)

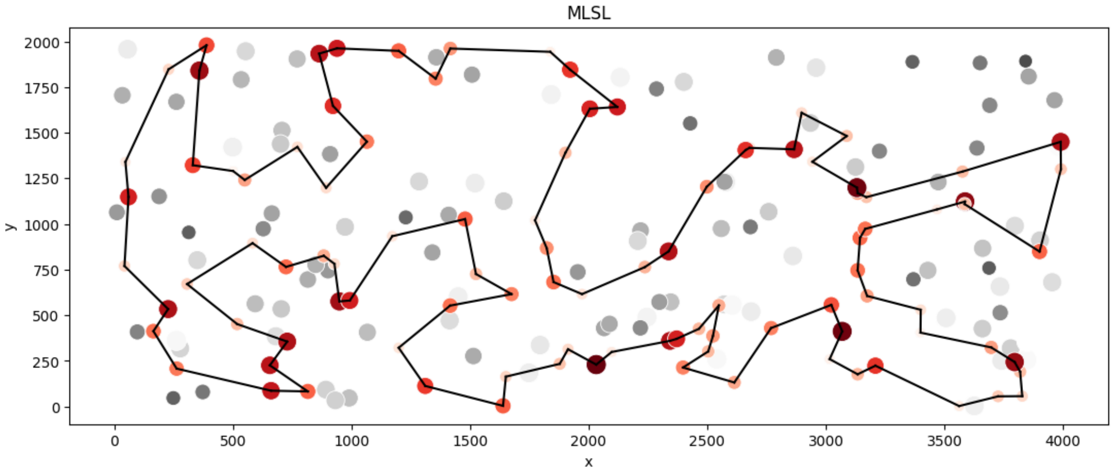
Input: time_budget
Output: Best solution found

Pseudocode:

```
s_best ←
while time_elapsed < time_budget:
    s ← GenerateRandomSolution()
    s ← LocalSearch(s)
    if s_best =  or f(s) < f(s_best):
        s_best ← s
return s_best
```

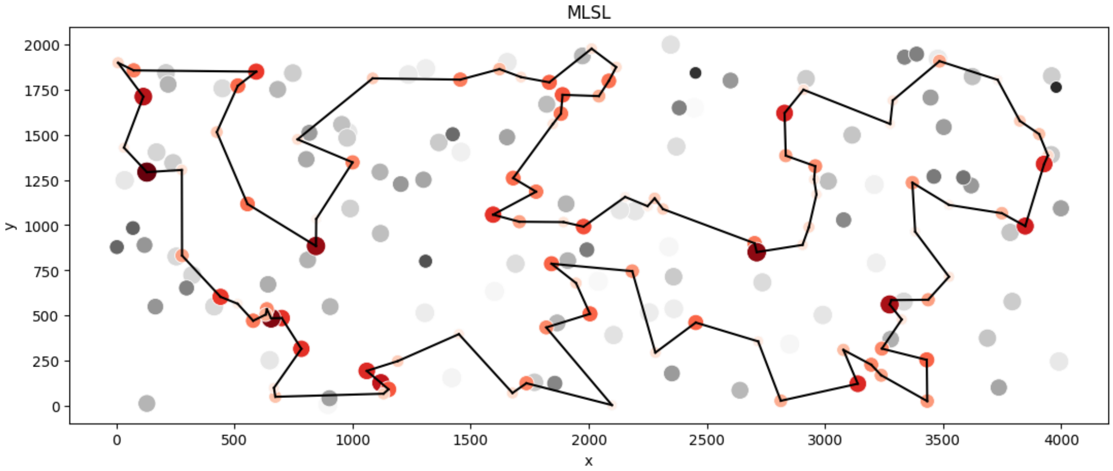
2.2 Results on Dataset A

```
Best solution: [158, 53, 100, 26, 97, 152, 1, 101, 86, 75, 2, 129, 82, 120, 44,
25, 16, 171, 175, 113, 56, 31, 78, 145, 92, 57, 55, 52, 185, 119, 40, 81, 90,
27, 165, 106, 178, 3, 49, 14, 144, 102, 62, 9, 148, 124, 94, 63, 79, 80, 176,
137, 23, 186, 89, 183, 143, 0, 117, 93, 140, 68, 46, 115, 139, 41, 193, 159, 69,
108, 18, 22, 146, 34, 48, 54, 177, 4, 112, 84, 35, 184, 160, 42, 43, 116, 65,
131, 149, 59, 51, 151, 133, 162, 123, 127, 70, 135, 154, 180]
Objective function statistics:
minimum.cost = 70662
mean = 71267.4
maximum.cost= 71693
```



2.3 Results on Dataset B

```
Best solution: [11, 182, 138, 33, 160, 29, 0, 109, 35, 143, 159, 106, 124, 62,
18, 55, 34, 170, 152, 183, 140, 28, 20, 148, 47, 94, 66, 172, 179, 185, 95, 86,
166, 194, 176, 180, 113, 103, 114, 137, 127, 89, 163, 165, 187, 153, 81, 77,
111, 104, 8, 82, 21, 141, 61, 36, 177, 5, 45, 142, 78, 175, 80, 190, 136, 73,
164, 54, 31, 193, 117, 198, 1, 135, 102, 63, 100, 40, 107, 10, 133, 122, 131,
112, 121, 51, 90, 147, 6, 188, 169, 132, 70, 3, 15, 145, 13, 195, 168, 139]
Objective function statistics:
minimum.cost = 45321
mean = 45751.25
maximum.cost= 46137
```



3 ILS

3.1 Algorithm's Pseudocode

Algorithm: Iterated Local Search (ILS)

Input: time_budget
Output: Best solution found

```
1. s ← GenerateRandomSolution()
2. s ← LocalSearch(s)
3. s_best ← s
4. While time_elapsed < time_budget:
    1. s' ← Perturb(s)
    2. s' ← LocalSearch(s')
    3. if Accept(s', s) then s ← s'
    4. if f(s) < f(s_best) then s_best ← s
Return s_best
```

3.2 Perturbations' Definition

The *double bridge* move is an essential tool in optimization for problems where solutions are permutations. It is effective because:

1. It makes large structural changes that help escape local optima.
2. It is computationally simple and easy to implement.
3. It complements local search, creating a balance between exploration and exploitation.
4. It has proven successful in theoretical and practical studies:

- Lin-Kernighan Heuristic Adaptations for the Generalized Traveling Salesman Problem
- Iterated Local Search

While **Node external insertion (exchange)** applied on top, serves as a complementary permutation, removing 5% of internal nodes and adding the same amount of external ones to escape from solution stagnation (statis)

3.3 Perturbations' Pseudocode

Double-Bridge Perturbation:

```
1. n = length(s)
2. Select four distinct positions i, j, k, l with 1 < i < j < k < l < n
3. segment1 ← s[1 … i-1]
4. segment2 ← s[i … j-1]
5. segment3 ← s[j … k-1]
6. segment4 ← s[k … l-1]
7. segment5 ← s[l … n]
8. s' ← Concatenate(segment1, segment3, segment2, segment4, segment5)
```

Return s'

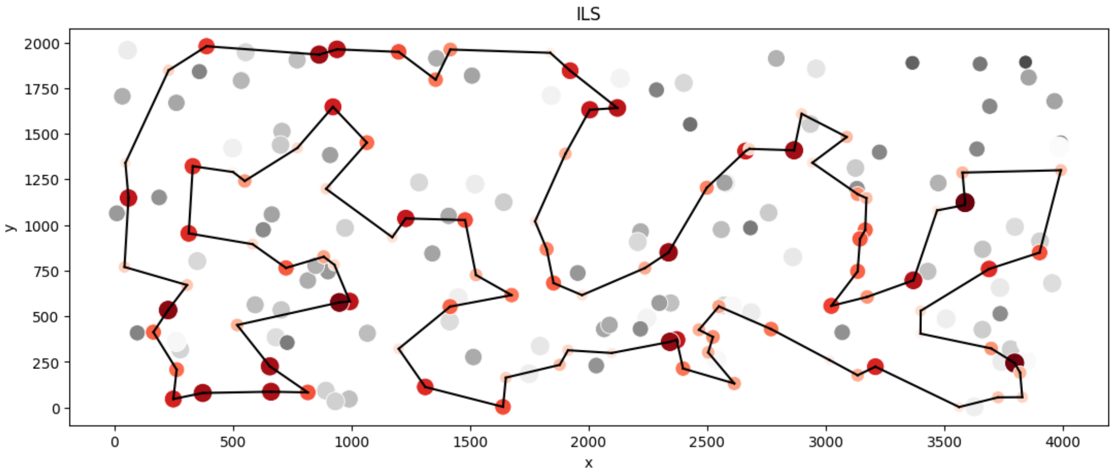
Node insertion Perturbation:

```
1. Choose a node a   selected_nodes
2. Remove a from s and from selected_nodes
3. Choose a node b   non_selected_nodes
4. Insert b into a new position in s
5. Update selected and non-selected nodes
6. s' ← s
7. Repeat n_insertions times
```

Return s'

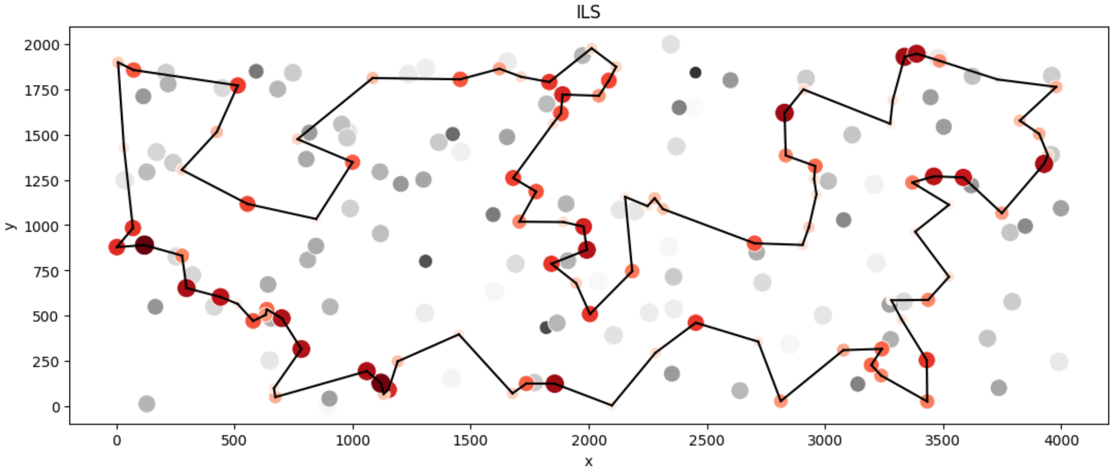
3.4 Results on Dataset A

```
Best solution: [171, 16, 25, 44, 120, 2, 152, 97, 1, 101, 75, 86, 26, 100, 53,
180, 154, 135, 70, 127, 123, 162, 133, 151, 51, 118, 59, 115, 46, 68, 139, 41,
193, 159, 181, 42, 43, 116, 65, 149, 131, 184, 84, 112, 4, 190, 10, 177, 54, 48,
160, 34, 146, 22, 18, 108, 140, 93, 117, 0, 143, 183, 89, 186, 23, 137, 176, 80,
79, 63, 94, 124, 148, 9, 62, 102, 144, 14, 49, 178, 106, 52, 55, 57, 129, 92,
179, 185, 40, 119, 165, 90, 81, 196, 145, 78, 31, 56, 113, 175]
Objective function statistics:
minimum.cost = 69107
mean = 69326.15
maximum.cost= 69765
Mean Number of iterations: 1106.2
Mean Number of sucessfull perturbations: 1106.2
```



3.5 Results on Dataset B

```
Best solution: [77, 141, 91, 61, 36, 177, 5, 78, 175, 142, 45, 80, 190, 136, 73,
54, 31, 193, 117, 198, 156, 1, 16, 27, 38, 63, 40, 107, 133, 122, 135, 131, 121,
51, 90, 147, 6, 188, 169, 132, 70, 3, 15, 145, 13, 195, 168, 139, 11, 138, 33,
160, 144, 104, 8, 82, 111, 29, 0, 109, 35, 143, 106, 124, 62, 18, 55, 34, 170,
152, 183, 140, 4, 149, 28, 20, 60, 148, 47, 94, 66, 179, 99, 130, 95, 185, 86,
166, 194, 176, 113, 114, 137, 127, 89, 103, 163, 187, 153, 81]
Objective function statistics:
minimum.cost = 43462
mean = 43765.6
maximum.cost= 44149
Mean Number of perturbations: 1114.8
```



4 Summary

```
                 Dataset A                                Dataset B  \
                       min       mean      max seconds/iter       min
ILS                69107.0  69326.150  69765.0      2223.00   43493.0
MLSM               70662.0  71267.400  71693.0      2223.00   45321.0
Steepest edge LS   72046.0  74033.715  78801.0         9.54   45393.0


                      mean      max seconds/iter
ILS               43783.05  44312.0      2218.00
MLSM              45751.25   4613.0      2218.00
Steepest edge LS  48264.78  50697.0         9.02
```

5 Conclusion

Due to choosing reasonable and academically prooved perturbations, we achieved results with `ILS` in same time budget better than on `MLSM`