Michał Kałmucki 151944
Dominik Ludwiczak 151948

Problem description

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.

Algorithm MSLS:

```
1. Generate random solution
2. Set random soluton as best one
3. For i in range(20):
    - Run Local Search on best solution
    - If LS solution is better than best one:
      - Set LS solution as best one
```

Algorithm ILS

```
1. Generate random solution
2. Run Local Search with random solution
3. Set best solution to LS solution
4. While runtime < 10000ms:
    - Perturb best solution
    - Run Local Search on perturbed solution
    - If LS solution is better then best one:
      - Set LS solution as best one
```

Perturbation

```
1. For i in range(3):
     - Select random index from solution
     - Select random node that is not in solution
     - Remove selected index from solution
     - Add random not used node at random place in solution
```

Results:

TSPA Instance

Method	Instance	Min Distance	Max Distance	Average Distance	Execution Time (ms)	Avergae time (ms)	Average LS runs
MSLS	TSPA	71615	78519	73886.605	2717553	10503.35	20
	TSPB	45111	51523	48358.664	1658511	10437.215	20
ILS	TSPA	69400	71285	70198.045	2004059	--	241.785
	TSPB	43649	46946	44650.46	2003919	--	266.39
STEEPEST-EDGES-RANDOM	TSPA	71756	78077	74003.145	86739	--	--
	TSPB	46019	51211	48491.05	96498	--	--

Source code link: Github

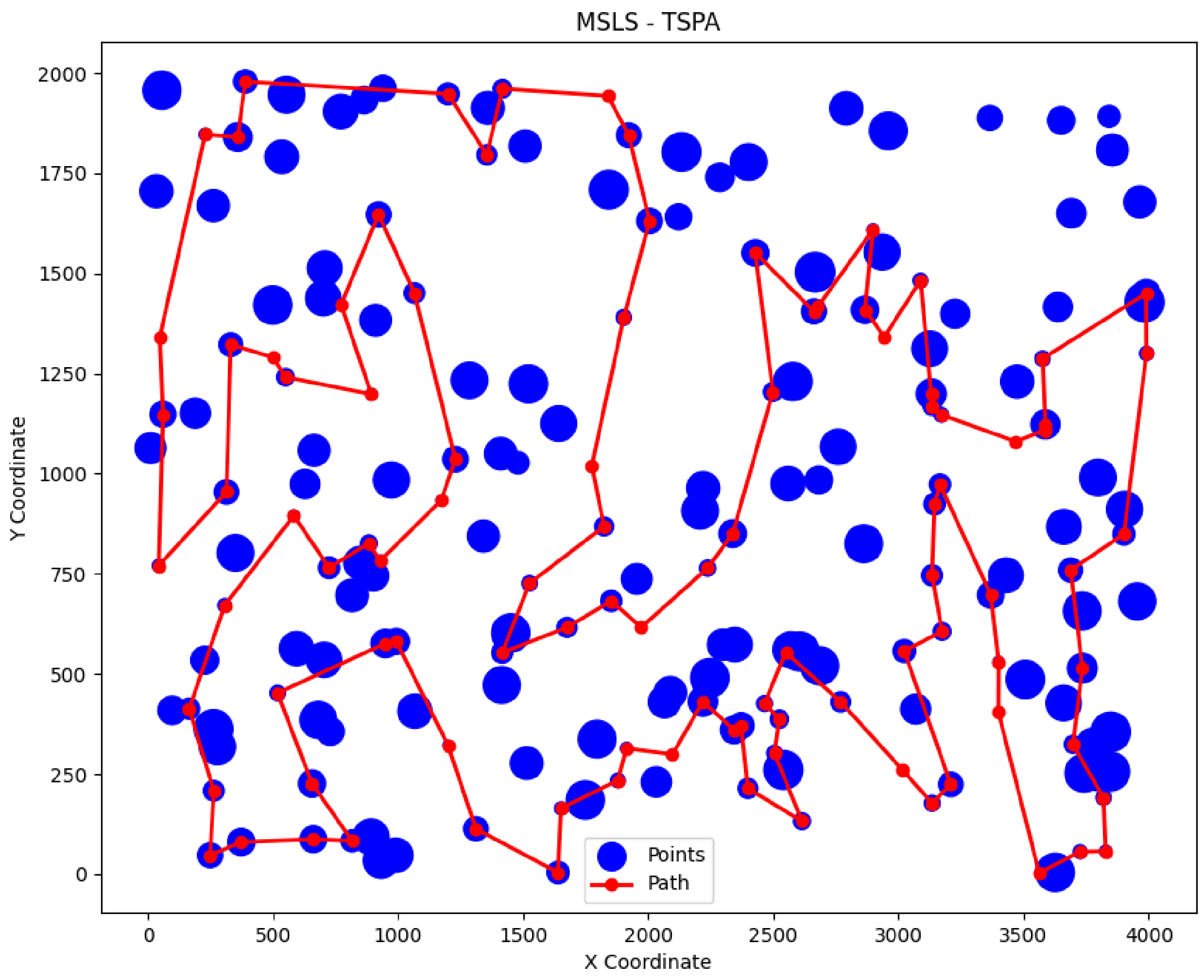
Conclusions:

Running local search multiple times on the smae starting solution can provide us slightly different outcomes, but with enough such starts it should find better solution than local search run once. In the ILS method running local search on perturbed solutions until runtime is not bigger than average runtime of MSLS we can achieve better solutions, because we introduce some sort of randomization in perturbations and we can run

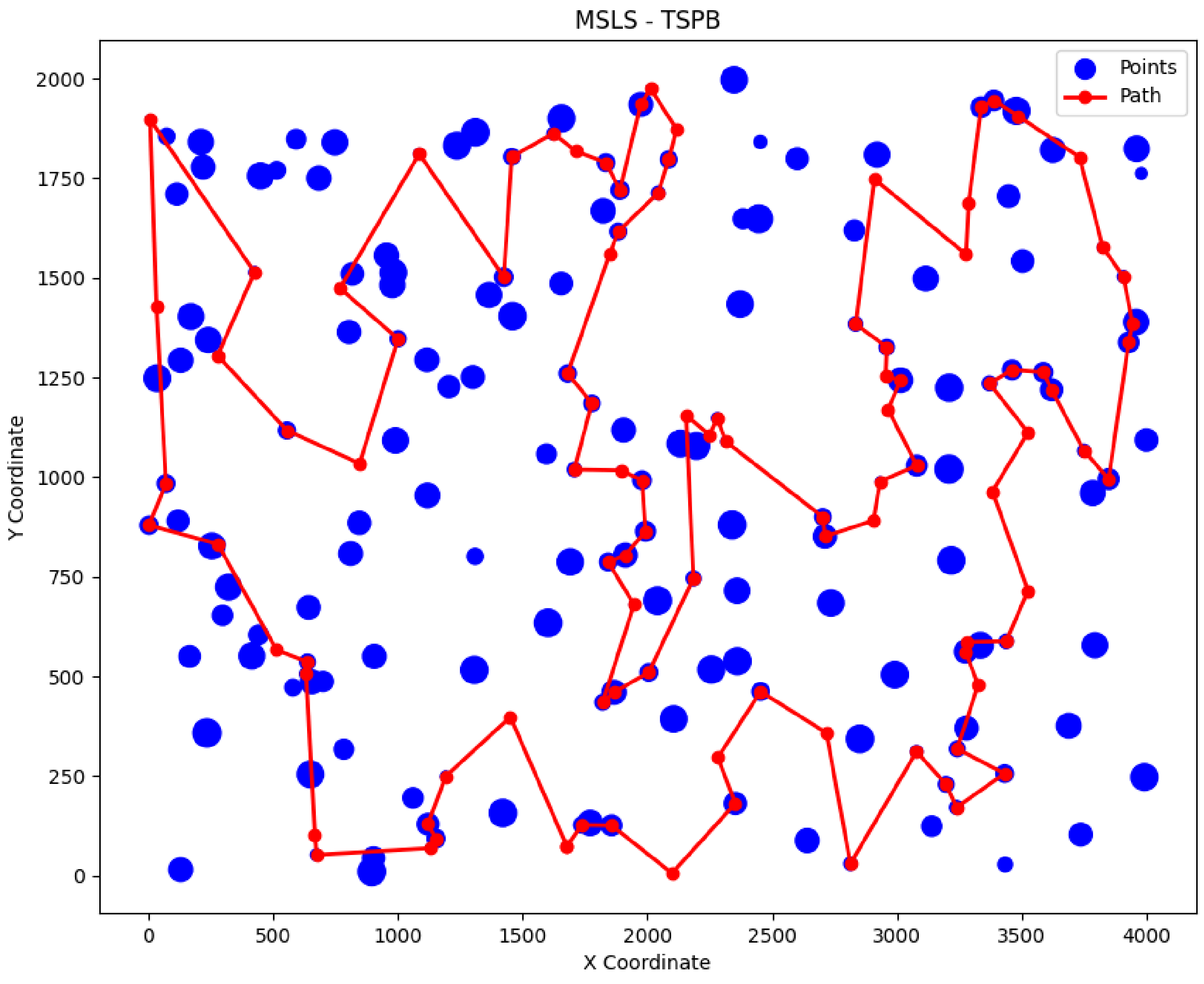
LS more times, here on average around 250 runs. Above conclusions shows us that ILS with good perturbations is able to perform better.

Graphs:

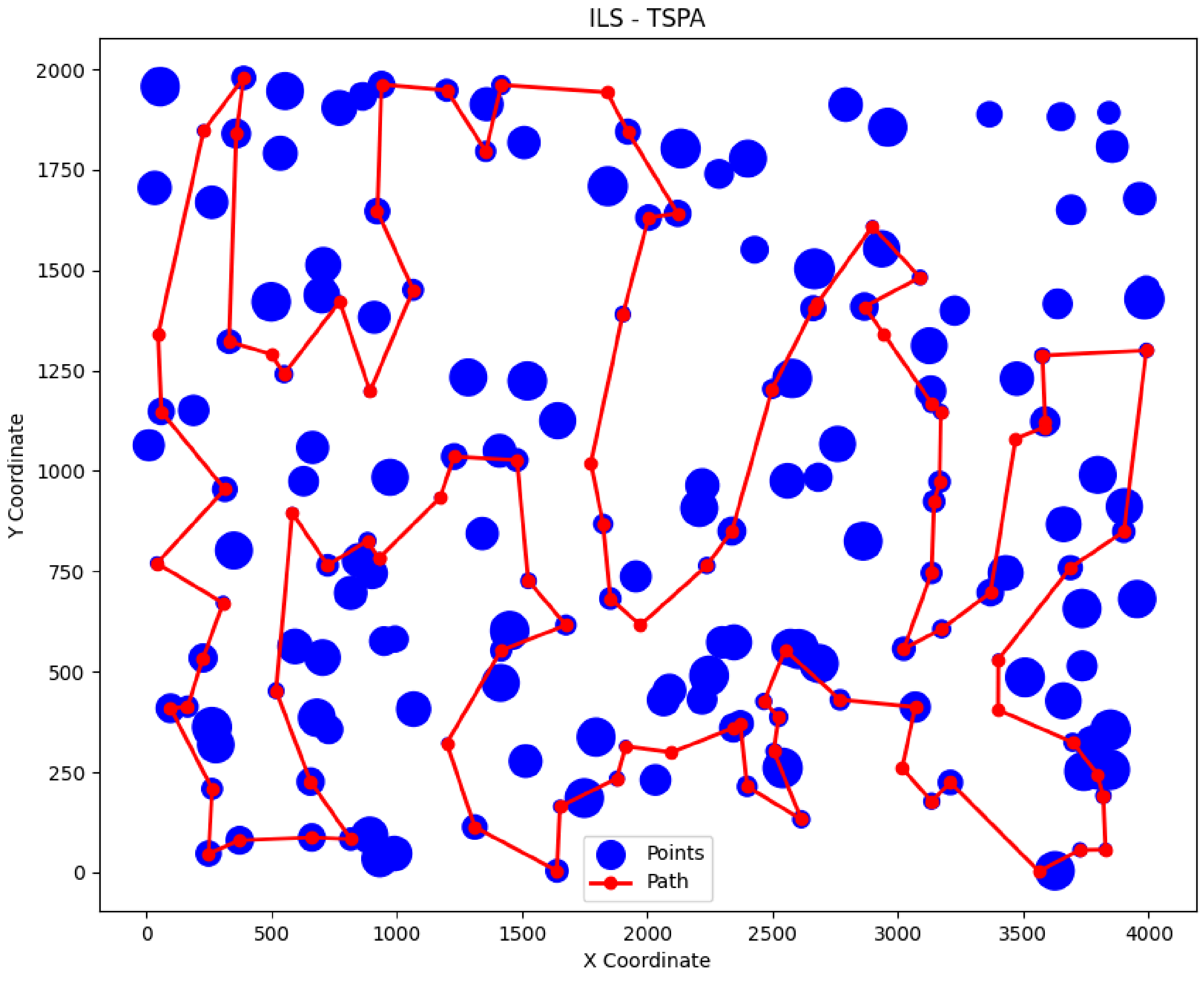
MSLS TSPA:



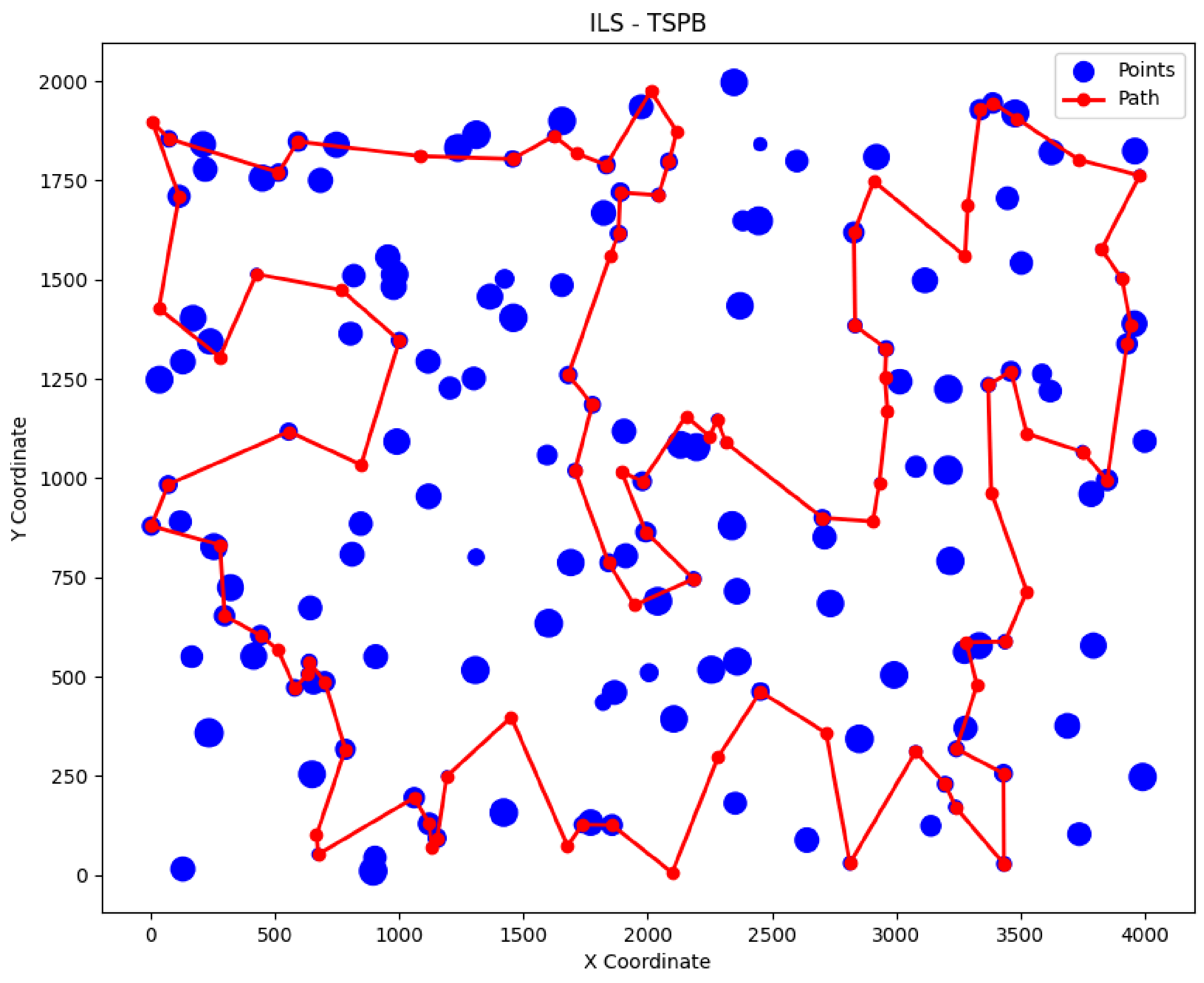
MSLS TSPB:


MSLS - TSPB
Points
Path
X Coordinate
Y Coordinate
0
500
1000
1500
2000
2500
3000
3500
4000
0
250
500
750
1000
1250
1500
1750
2000

ILS TSPA:


ILS - TSPA
Y Coordinate
X Coordinate
Points
Path
0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000
2500
3000
3500
4000

ILS TSPB:


ILS - TSPB
Points
Path
Y Coordinate
X Coordinate
2000
1750
1500
1250
1000
750
500
250
0
0
500
1000
1500
2000
2500
3000
3500
4000