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.

Local Search:

Algorithm:

```
1. Initialize a list of move types.
2. Set foundBetterSolution to true.

3. While a better solution is found:
    - Reset minimum objective change and best move.

    - For each position i in the solution (up to the second-last):
        - Shuffle the list of move types.

        - For each move type:
            - Create a list of possible moves (nodes).
            - If the move type is CHANGE_WITH_NOT_USED, set nodes to elements not
in the solution.
            - Shuffle nodes.

            - For each possible position j in nodes:
                - Simulate the move to calculate its effect (objectiveChange).

                - If the search type is:
                    - Greedy: If the move improves the solution, update best move.
                    - Steepest: If the move is better than previous moves, update
the best move.

    - If no best move is found, stop searching for a better solution.
    - Otherwise, apply the best move to update the solution.
```

Objective change calculation:

Swapping nodes within solution:

```
Calculate `objectiveChange` as follows:
        Remove distances:
            - distance from prevNode1Idx to node1Idx
            - distance from node1Idx to nextNode1Idx
            - distance from prevNode2Idx to node2Idx
            - distance from node2Idx to nextNode2Idx
        Add distances:
            - distance from prevNode1Idx to node2Idx
            - distance from node2Idx to nextNode1Idx
            - distance from prevNode2Idx to node1Idx
            - distance from node1Idx to nextNode2Idx
    Return `objectiveChange`
```

Swapping node from solution with one outside solution

```
Calculate `objectiveChange` as follows:
    Remove costs and distances associated with `node1Idx`:
        - distance from prevNode1Idx to node1Idx
        - distance from node1Idx to nextNode1Idx
        - nodeCosts of node1Idx
    Add costs and distances associated with `node2Idx`:
        - distance from prevNode1Idx to node2Idx
        - distance from node2Idx to nextNode1Idx
        - nodeCosts of node2Idx
Return `objectiveChange`
```

Exchanging edges:

```
Calculate `objectiveChange` as follows:
    Remove distances:
        - distance from node1Idx to nextNode1Idx
        - distance from node2Idx to nextNode2Idx
    Add distances:
        - distance from node1Idx to node2Idx
        - distance from nextNode1Idx to nextNode2Idx
Return `objectiveChange`
```

Results:

TSPA Instance

Method	Min Distance	Max Distance	Average Distance	Execution Time (ms)
GREEDY 2-REGRET WEIGHTED	71,108	73,395	72,130.045	-
GREEDY-EDGES-RANDOM	71351	78486	74209.62	110849
GREEDY-EDGES-REGRET	70571	72695	71492.87	5058
GREEDY-NODES-RANDOM	78156	94451	84799.655	63224
GREEDY-NODES-REGRET	70687	72984	71602.01	4226
STEEPEST-EDGES-RANDOM	71756	78077	74003.145	86739
STEEPEST-EDGES-REGRET	70510	72614	71438.76	6184
STEEPEST-NODES-RANDOM	78755	96702	88257.14	108331
STEEPEST-NODES-REGRET	70626	72950	71658.68	5384

TSPB Instance

Method	Min Distance	Max Distance	Average Distance	Execution Time (ms)
GREEDY 2-REGRET WEIGHTED	47,144	55,700	50,919.565	-
GREEDY-EDGES-RANDOM	45603	51279	48292.18	97457
GREEDY-EDGES-REGRET	45824	55029	50160.145	7225
GREEDY-NODES-RANDOM	52387	68671	60033.585	65873
GREEDY-NODES-REGRET	46360	53761	50089.915	6609
STEEPEST-EDGES-RANDOM	46019	51211	48491.05	96498
STEEPEST-EDGES-REGRET	45867	53667	49888.78	10246
STEEPEST-NODES-RANDOM	55195	73086	62964.445	110889
STEEPEST-NODES-REGRET	46371	55385	50217.305	11671

Source code link: Github

Conclusions:

Conclusions

1. Comparison of GREEDY and STEEPEST Methods:

- Both approaches were comparable on Solution quality and execution time suggesting other factors having bigger impact on experiment results.

2. Neighborhood Type (EDGES vs. NODES):

- Distance Performance: Both EDGES and NODES neighborhoods have distinct effects on solution quality. The EDGES neighborhood methods generally achieve lower minimum and average distances compared to NODES neighborhoods. This suggests that swapping edges between nodes in the TSP solution might be a more effective move strategy for minimizing total travel distance.
- Execution Time: There is no consistent trend in execution time between EDGES and NODES, as it largely depends on the specifics of each method and instance.

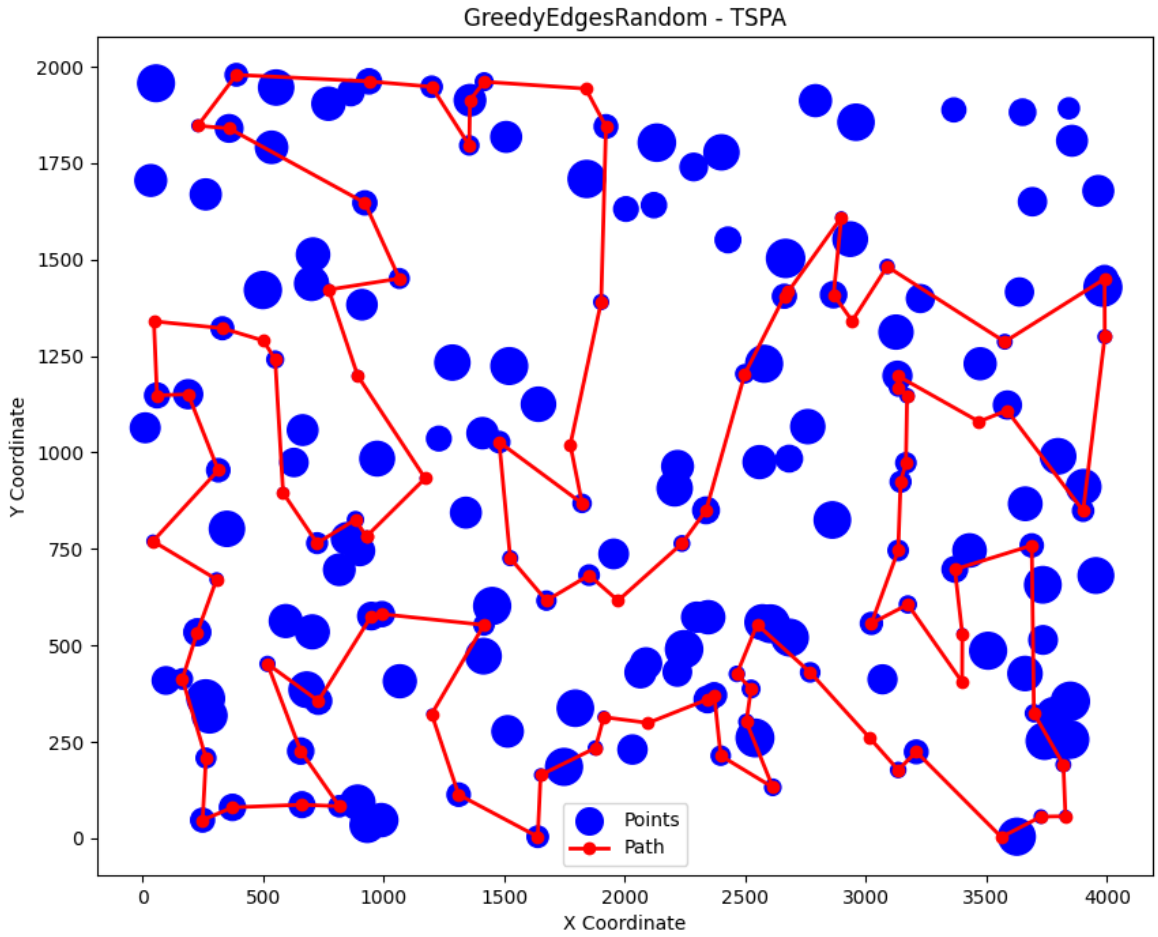
3. Starting Solutions (Regret vs. Random):

- Regret Initialization: Methods starting from a Regret solution (a more strategic initial solution generated with the regret heuristic) consistently achieve better average and minimum distances compared to those starting from a Random solution. This demonstrates the benefit of using a decent starting solution, as it provides a better foundation for local search to refine.
- Random Initialization: The Random starting solution results in higher variance, leading to both higher maximum and average distances. This emphasizes that an unoptimized starting point often requires more iterations to reach an equivalent level of solution quality.

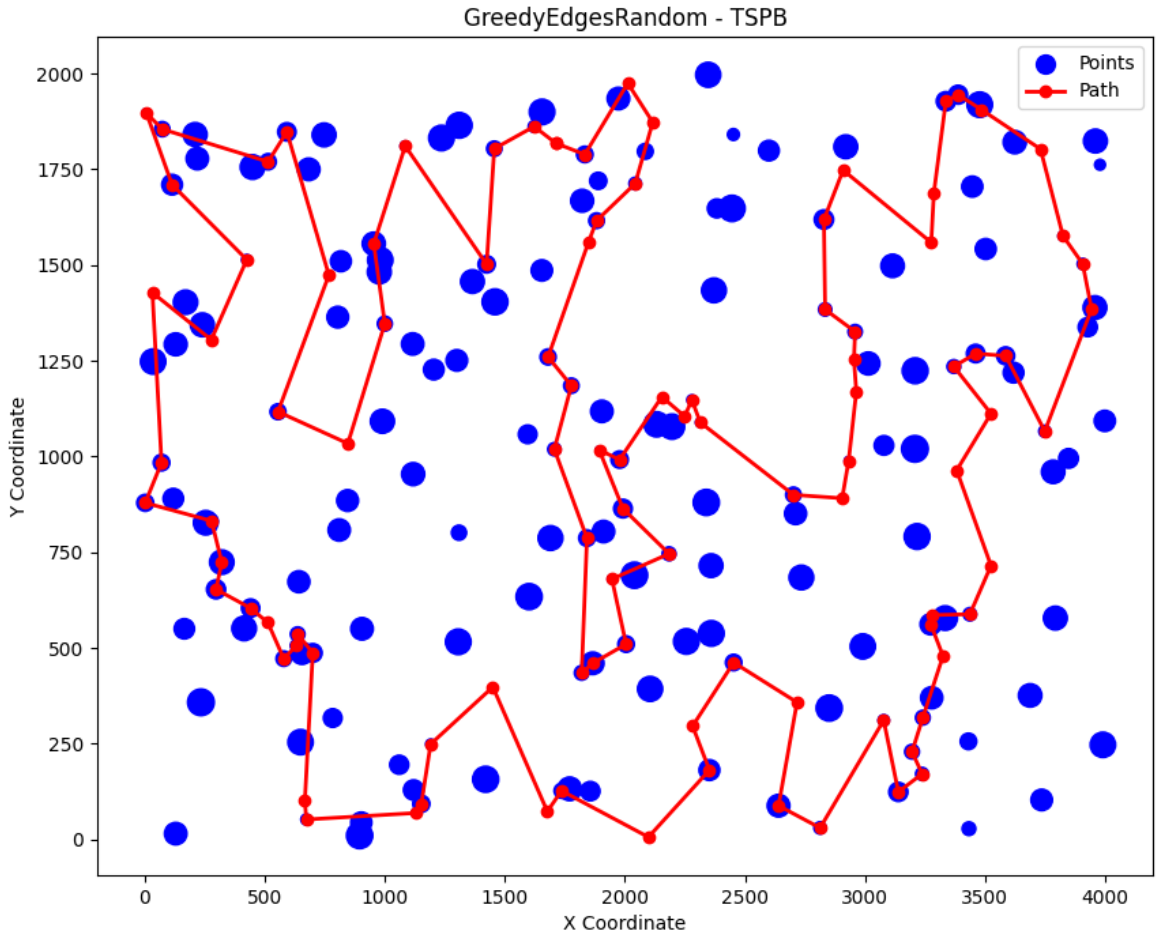
Graphs:

GREEDY:

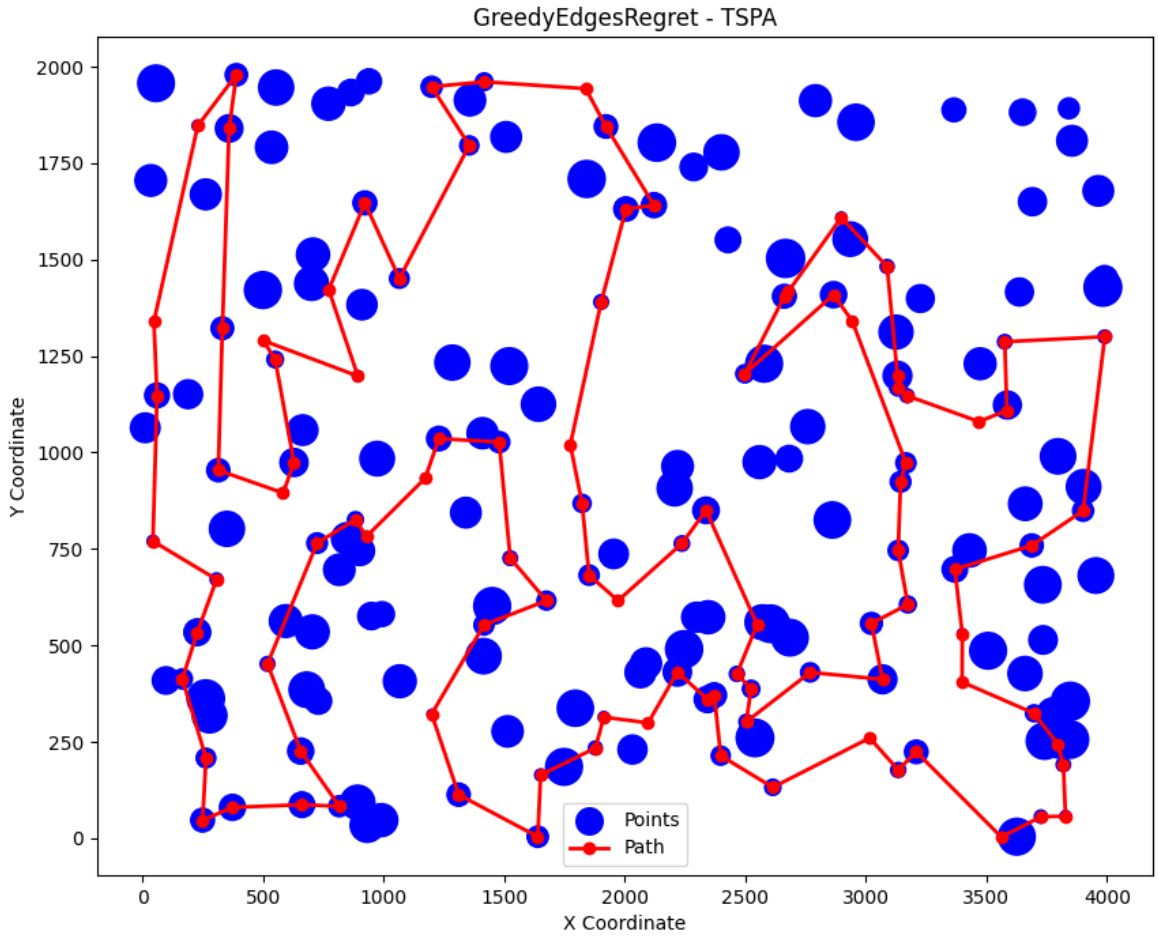
Exchange edges, TSPA, RANDOM:



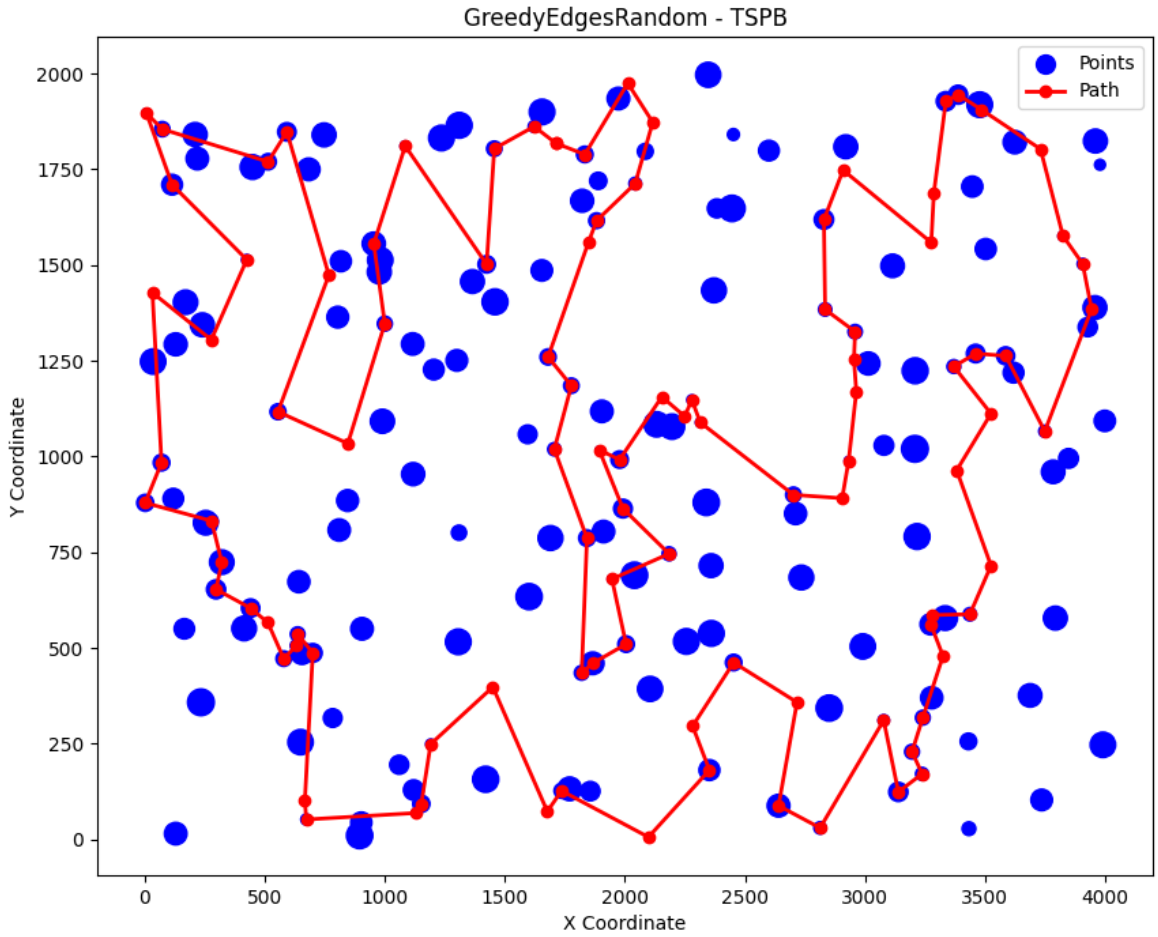
Exchange edges, TSPB, RANDOM:



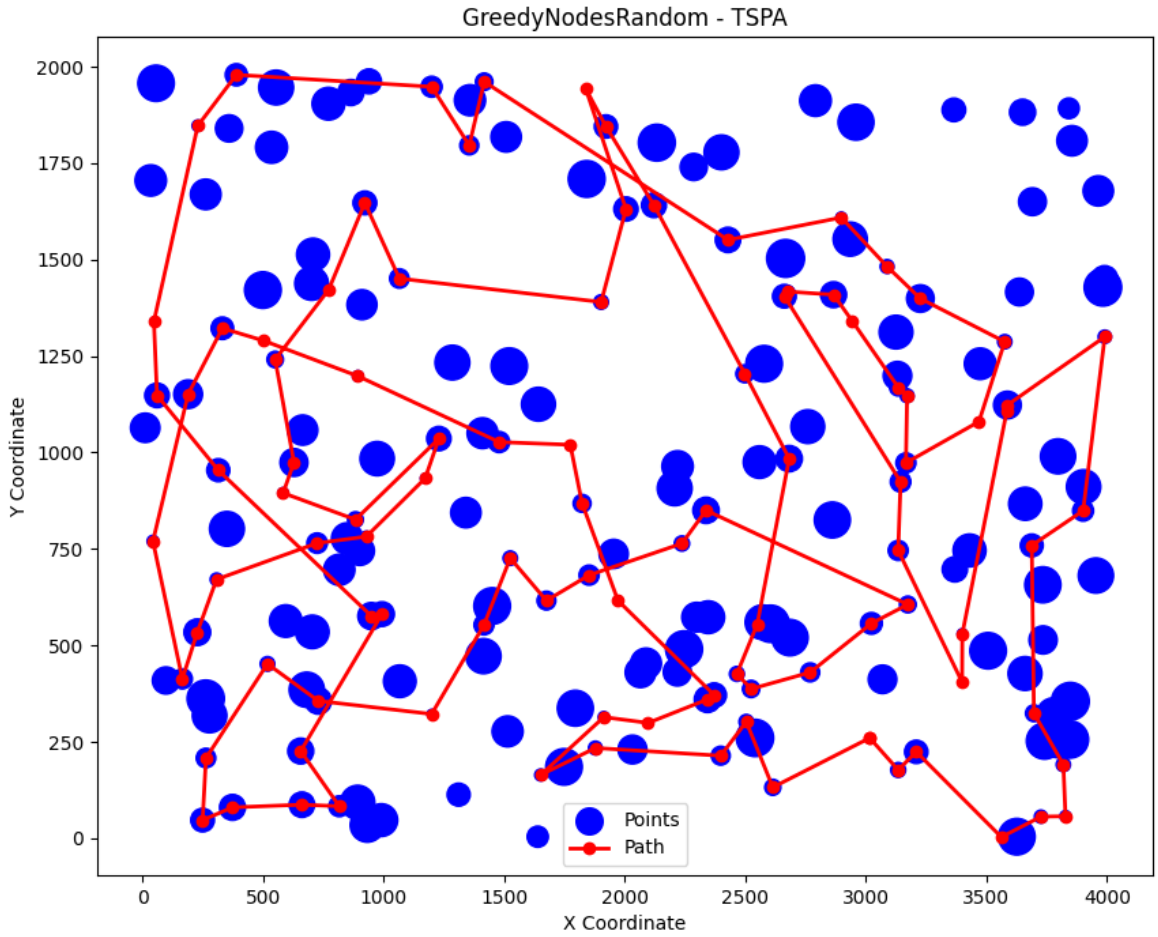
Exchange edges, TSPA, REGRET:



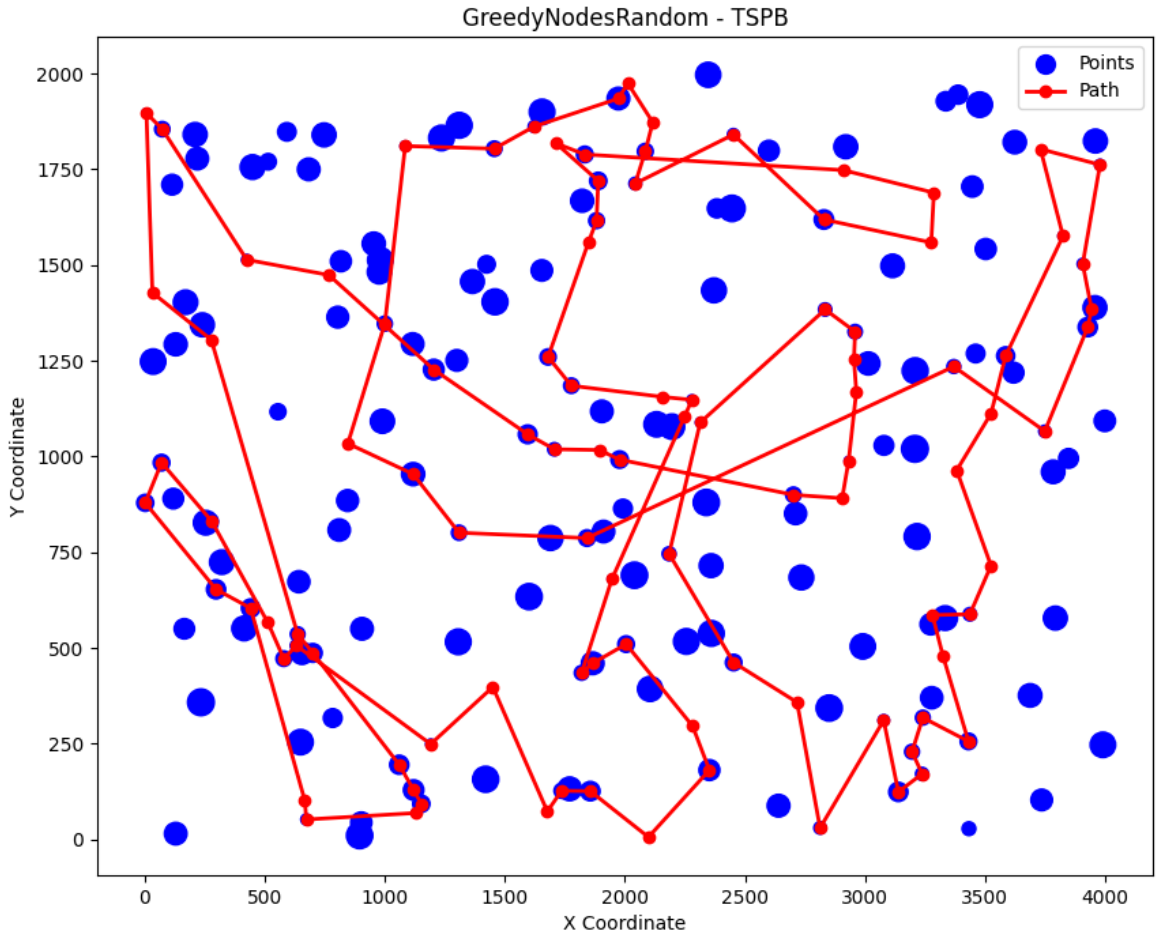
Exchange edges, TSPB, REGRET:



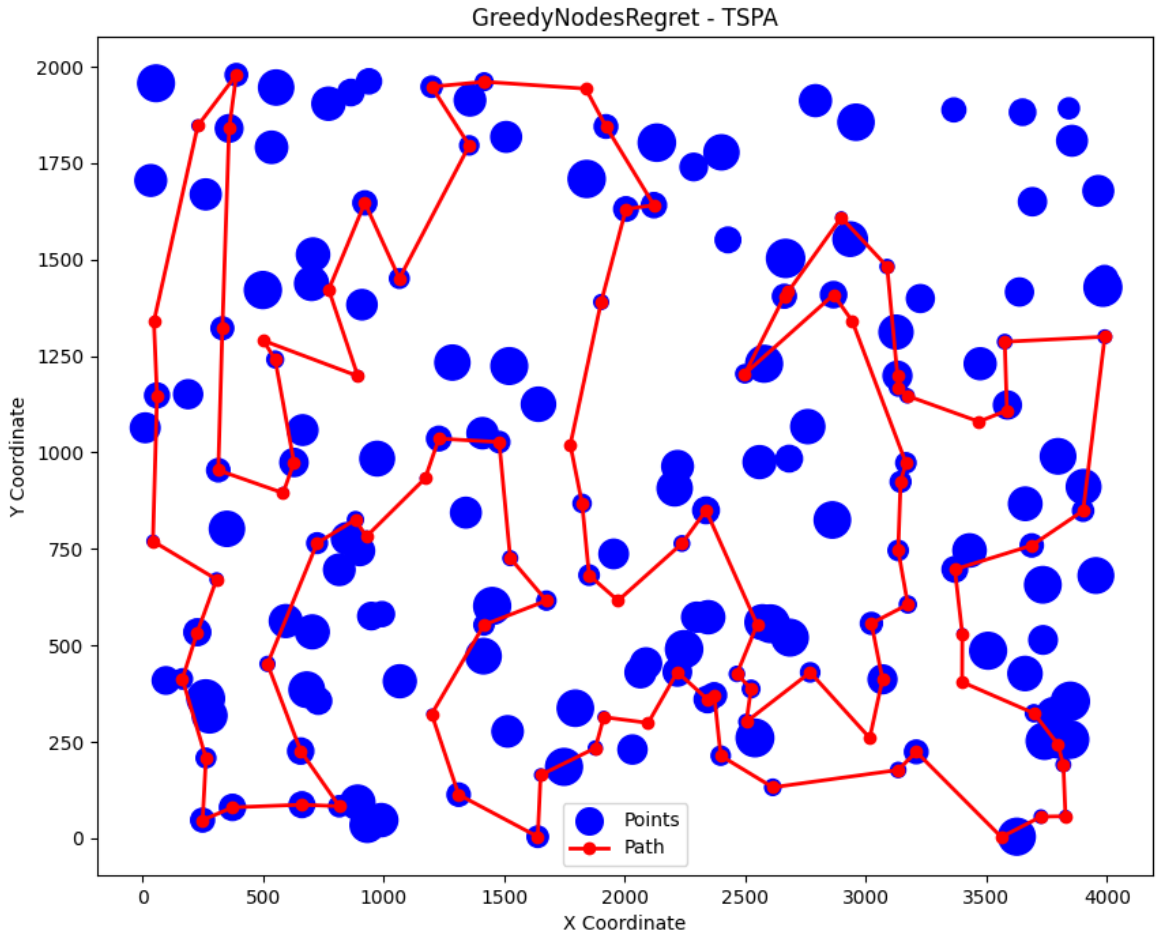
Nodes, TSPA, RANDOM:



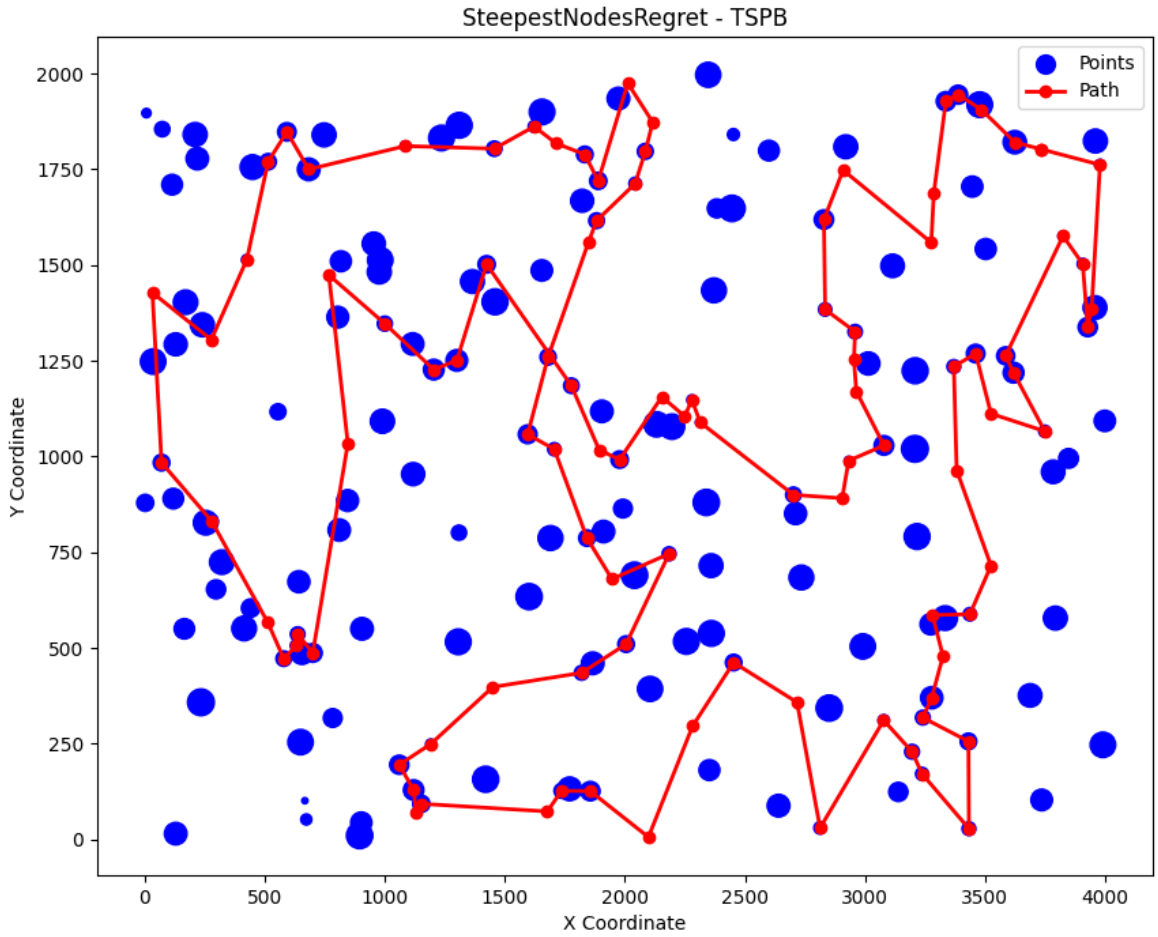
Nodes, TSPB, RANDOM:



Nodes, TSPA, REGRET:

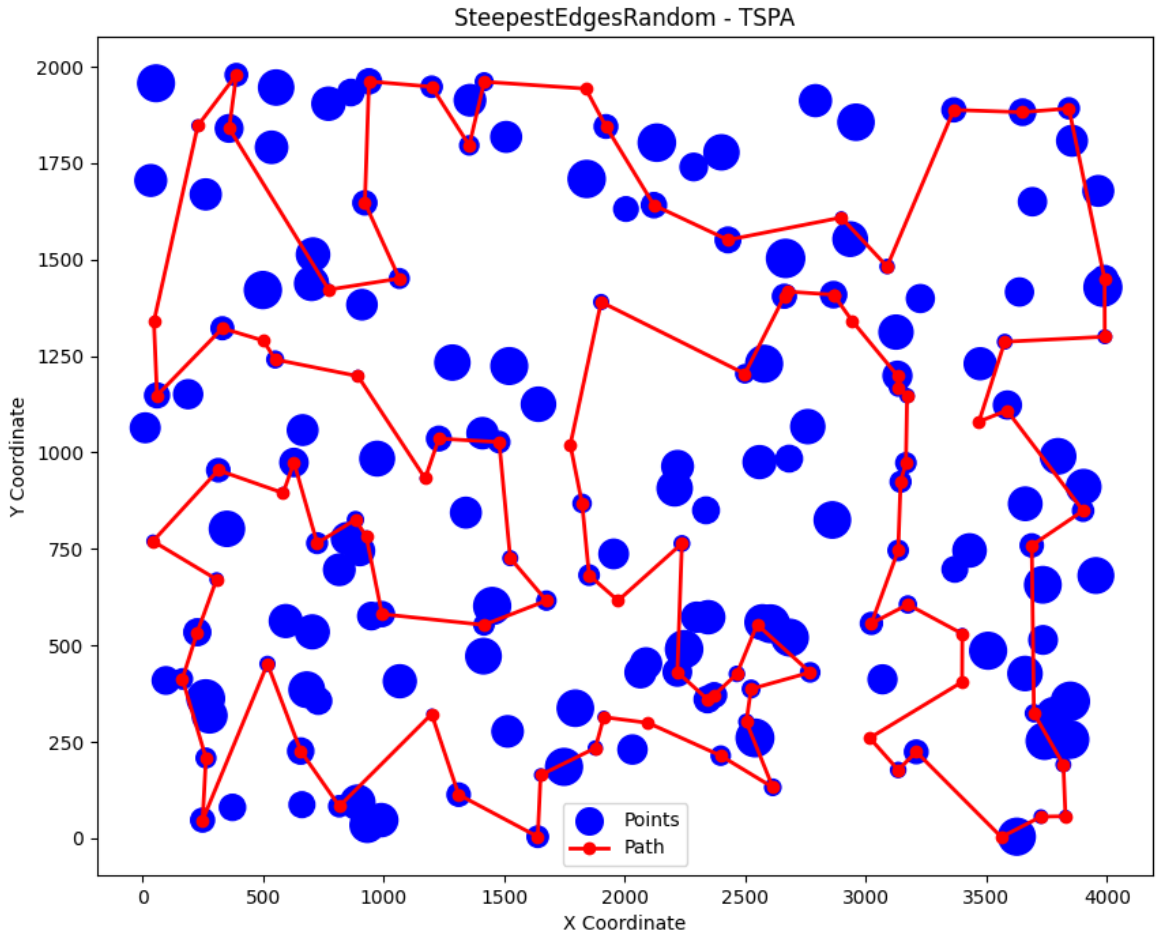


Nodes, TSPB, REGRET:

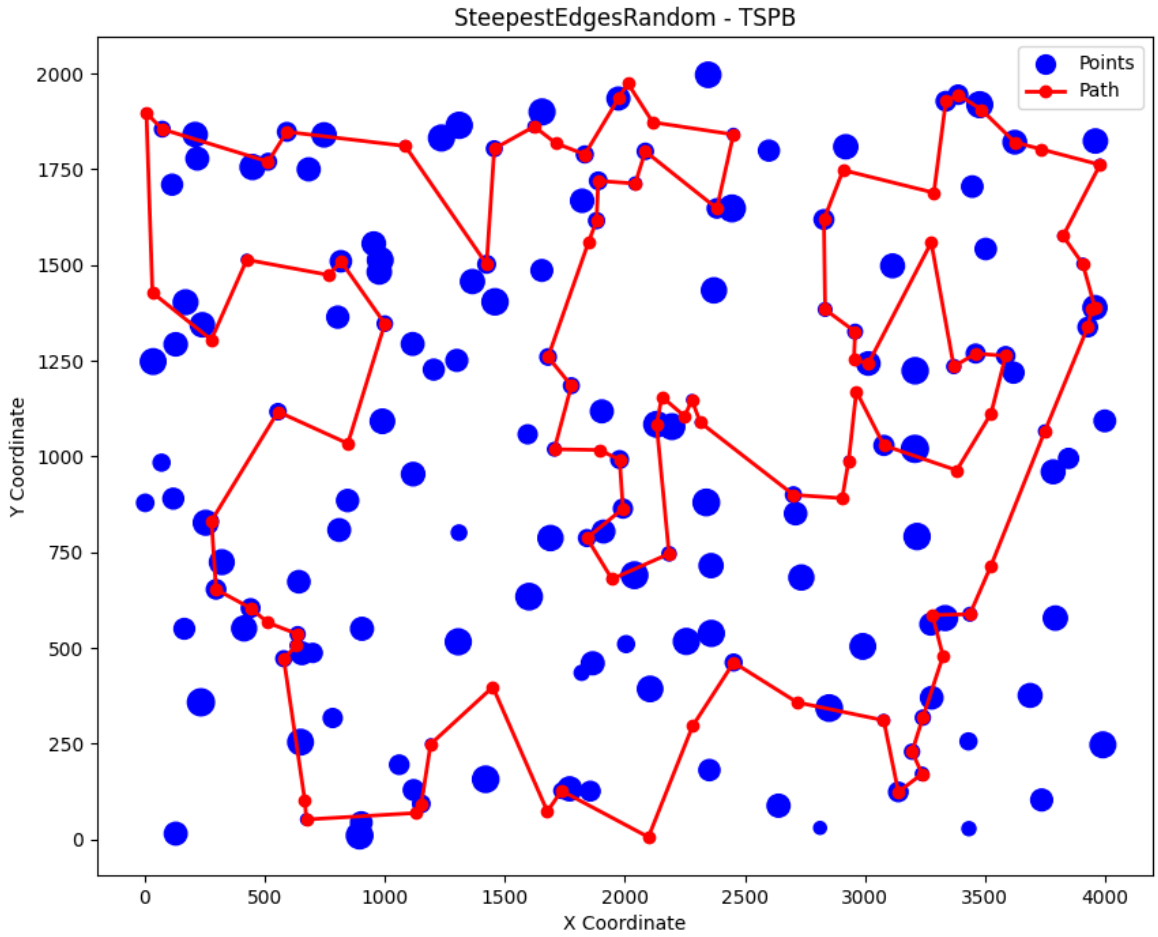


STEEPEST:

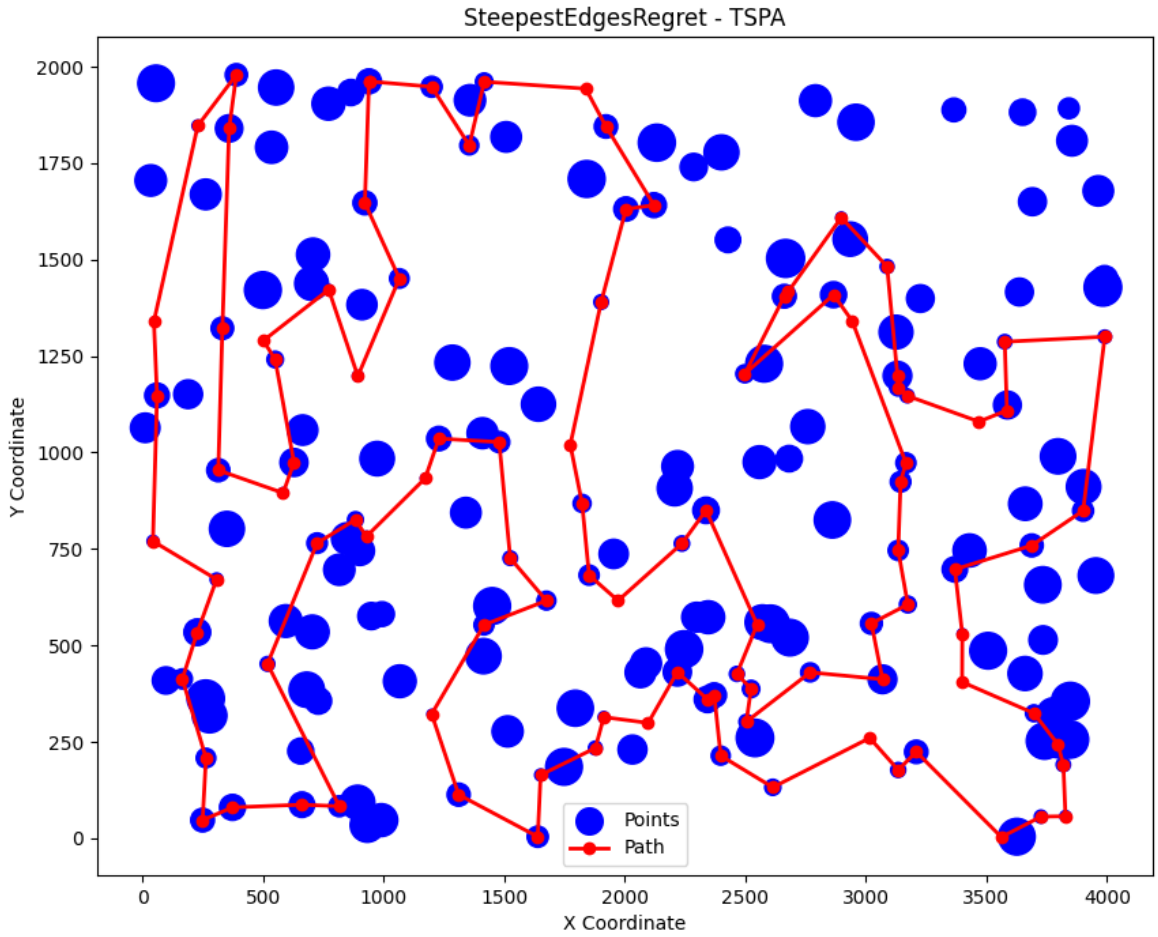
Exchange edges, TSPA, RANDOM:



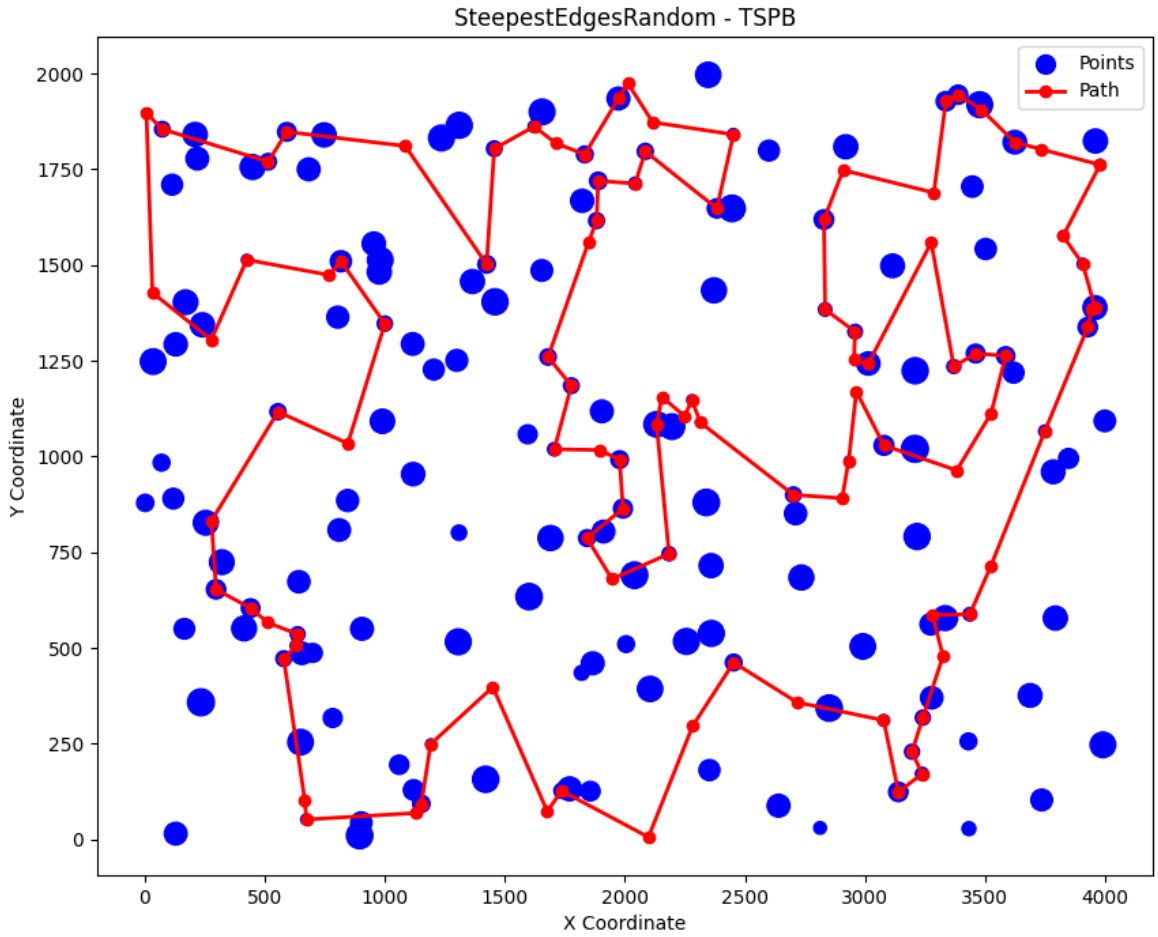
Exchange edges, TSPB, RANDOM:



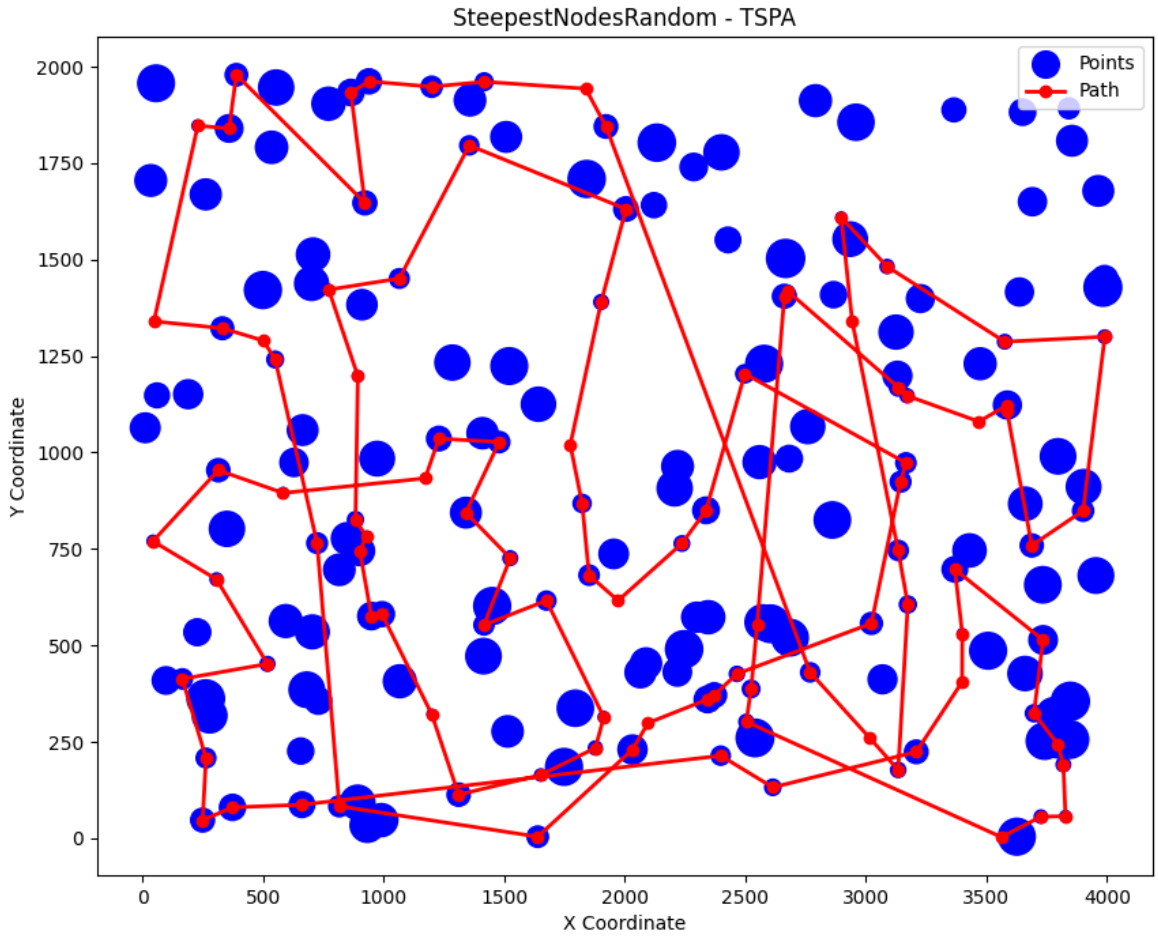
Exchange edges, TSPA, REGRET:



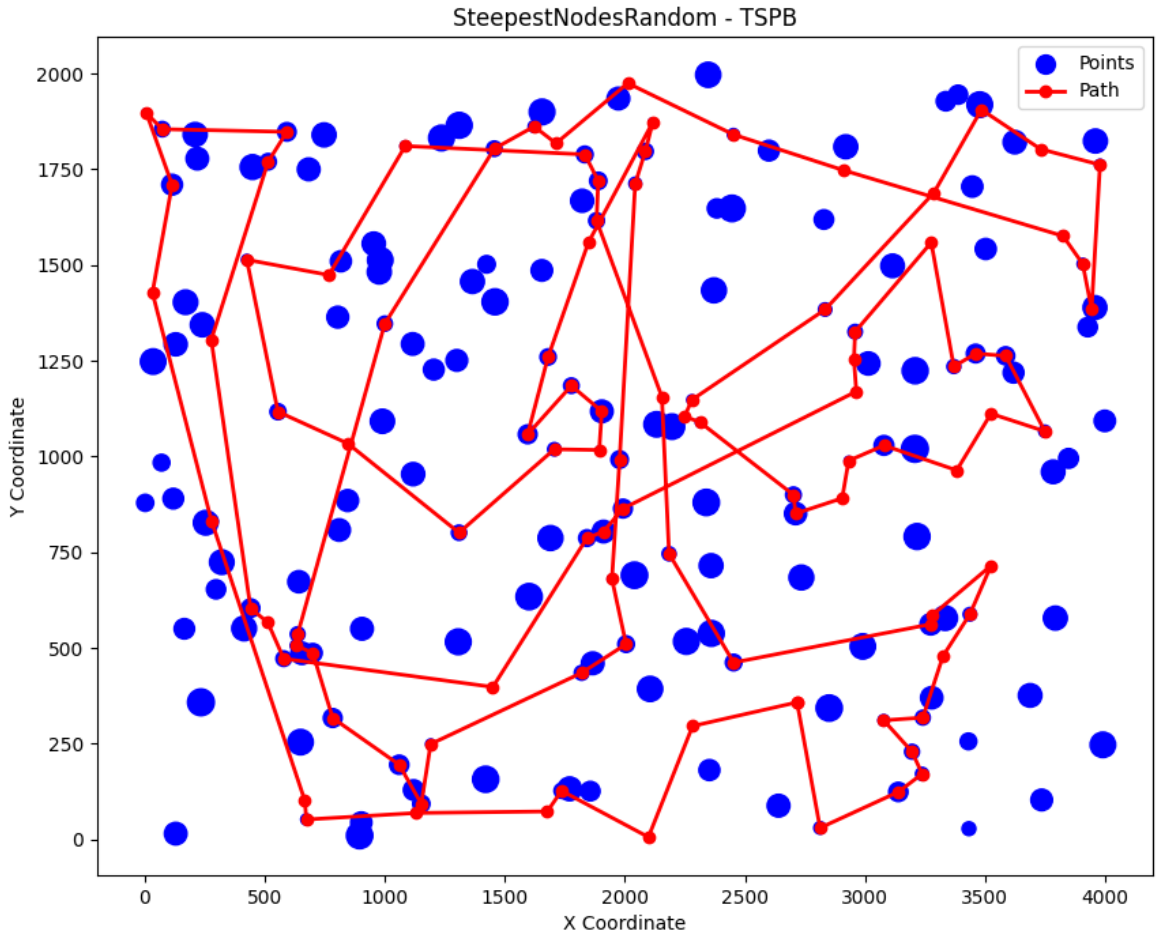
Exchange edges, TSPB, REGRET:



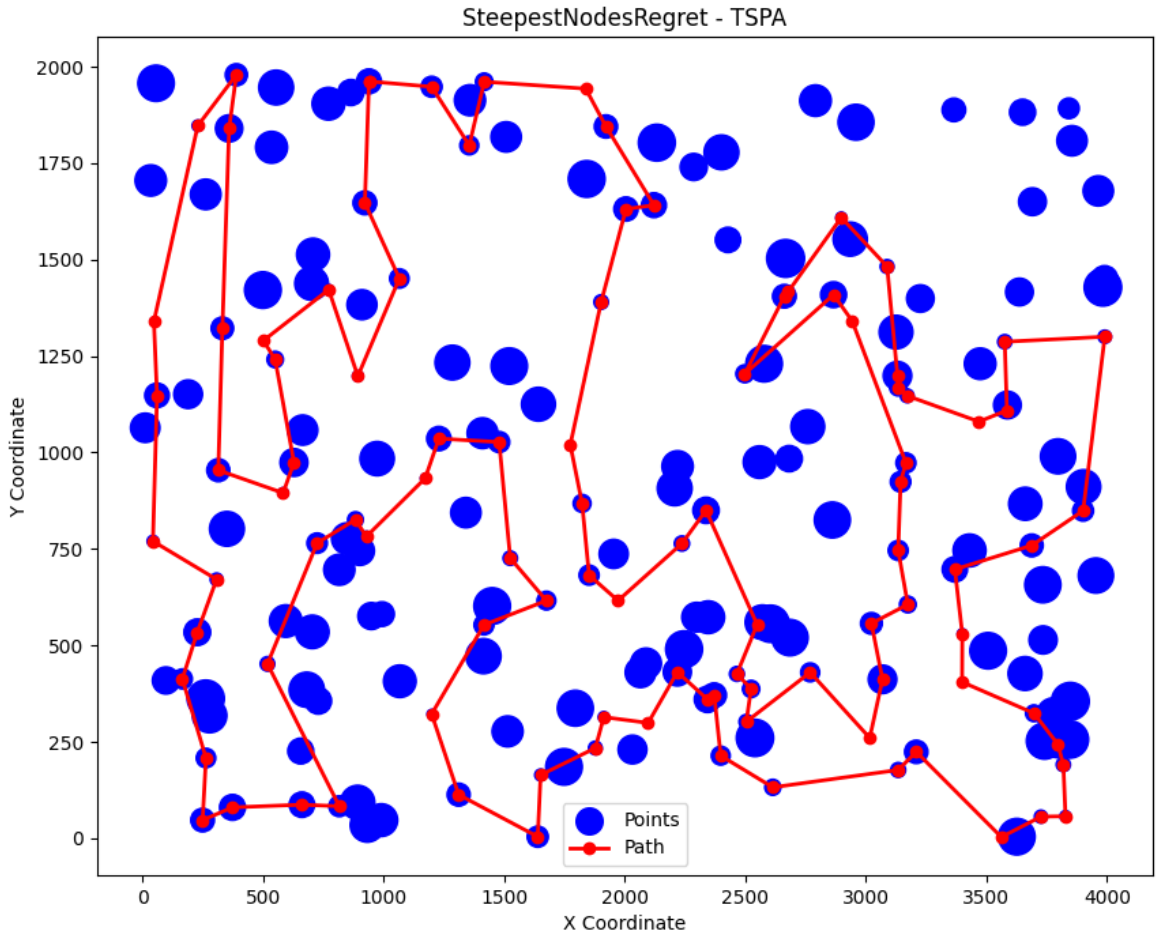
Nodes, TSPA, RANDOM:



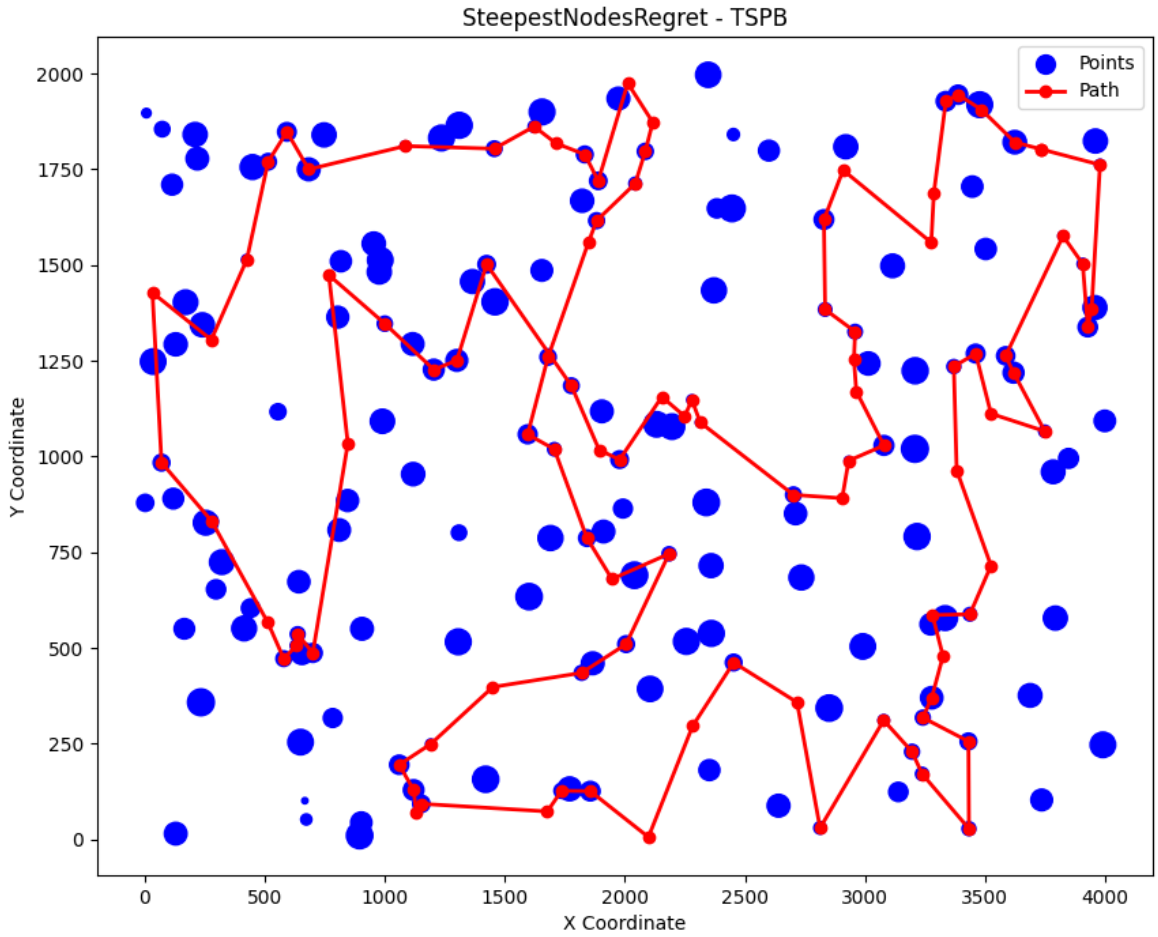
Nodes, TSPB, RANDOM:



Nodes, TSPA, REGRET:



Nodes, TSPB, REGRET:


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