Assignment 9: Hybrid evolutionary algorithm

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

Description of the problem

The goal is to implement a hybrid evolutionary algorithm and compare it with the MSLS, ILS, and LNS methods implemented in the previous assignments.

- Proposed algorithm parameters:
- Elite population of 20.
- Steady state algorithm.
- Parents selected from the population with the uniform probability.

There must be no copies of the same solution in the population (you can compare the entire solution or the value of the objective function).

Proposed recombination operators:

- Operator 1. We locate in the offspring all common nodes and edges and fill the rest of the solution at random.
- Operator 2. We choose one of the parents as the starting solution. We remove from this solution all edges and nodes that are not present in the other parent. The solution is repaired using the heuristic method in the same way as in the LNS method. We also test the version of the algorithm without local search after recombination (we still use local search for the initial population).

If the algorithm described above would cause premature convergence, it can be modified, e.g. additional diversification preservation mechanisms.

Additionally, another custom recombination operator can be proposed.

Experiment with parameters same as ILS/LNS.

Hybrid Evolutionary Algorithm

Pseudocode:

```
1. Generate initial population using LS
2. While within timeout:
    2.1. Select 2 random parent solutions
    2.2. Generate child: a. Find common nodes between parents b. Find common edges between parents (any direction) c. Create stripped solution from parent1, keeping only common nodes d. Fill stripped solution to required node count:
     - Random mode: Insert random non-cycle node at random position, avoiding common edge breaks
     - Heuristic mode: Insert nodes using nn heuristic, avoiding common edge breaks e. Optionally apply LS f. Return completed child
    2.3. If child is unique and better than worst solution in population:
  Replace worst solution with child
3. Return best solution from population
```

Function performance

Method	Dataset A	Dataset B
Multiple Start Local Search	72010.48(70553-77610)	46477.23(45212-47381)

Iterated Local Search	70797.655(69875-72440)	45949.965(44070-47548)
Large-Scale Neighborhood Search - no LS	69935(69230-71274)	44984(44437-46112)
Large-Scale Neighborhood Search - with LS	69774(69230-70258)	44373(43550-45506)
Evolutionary Algorithm heuristic, without Local Search	70664.3(70082-71423)	44494.85(43772-45563)
Evolutionary Algorithm heuristic, with Local Search	70151.75(69326-70688)	44034.7(43472-44892)
Evolutionary Algorithm random, without Local Search	73171.3(72330-74375)	47593.1(46332-48515)
Evolutionary Algorithm random, with Local Search	70986.8(70049-71642)	44976.7(44142-45610)

Average running time

Method	Dataset A	Dataset B
Multiple Start Local Search	46.43 s	46.24 s
Iterated Local Search	46.42 s	46.23 s
Large-Scale Neighborhood Search - no LS	47.29 s	47.28 s
Large-Scale Neighborhood Search - with LS	47.26 s	47.39 s
Evolutionary Algorithm heuristic, without Local Search	52.03 s	174 s
Evolutionary Algorithm heuristic, with Local Search	51.76 s	52.19 s
Evolutionary Algorithm random, without Local Search	100.23 s	65.84 s
Evolutionary Algorithm random, with Local Search	51.95 s	52.3 s

Number of local search runs

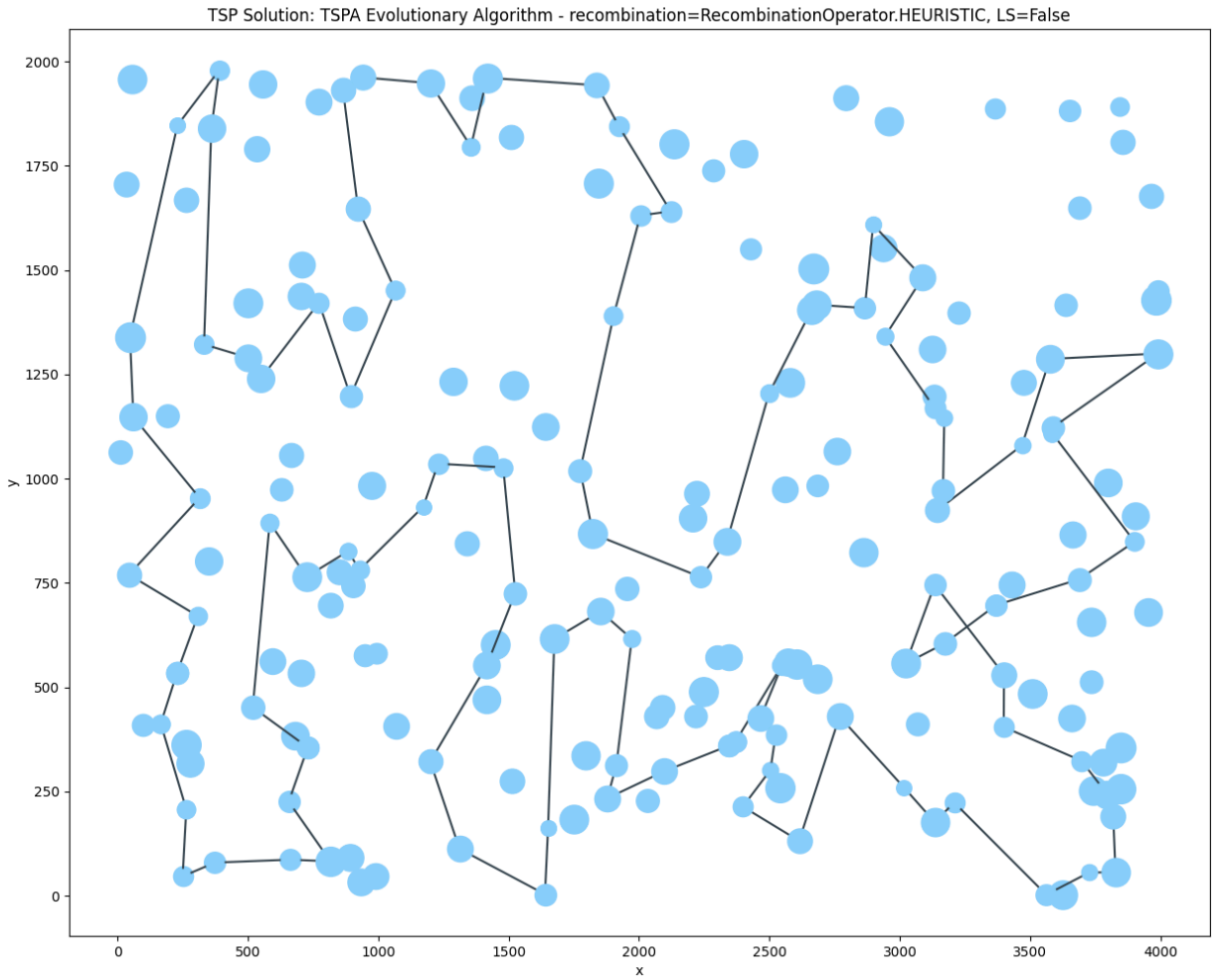
Method	Dataset A	Dataset B
Multiple Start Local Search	200(200-200)	200(200-200)
Iterated Local Search	642.54(624-658)	645.46(593-666)
Large-Scale Neighborhood Search - no LS	678(494-785)	720(682-756)
Large-Scale Neighborhood Search - with LS	613(592-622)	616(593-630)

Evolutionary Algorithm heuristic, without Local Search	2439.9(2090-2836)	1582.5(35-1836)
Evolutionary Algorithm heuristic, with Local Search	1093.25(1024-1136)	883.35(738-995)
Evolutionary Algorithm random, without Local Search	195676.35(158797-216642)	190644.65(99079-210599)
Evolutionary Algorithm random, with Local Search	844.85(753-931)	666.2(622-746)

Evolutionary Algorithm heuristic, without Local Search

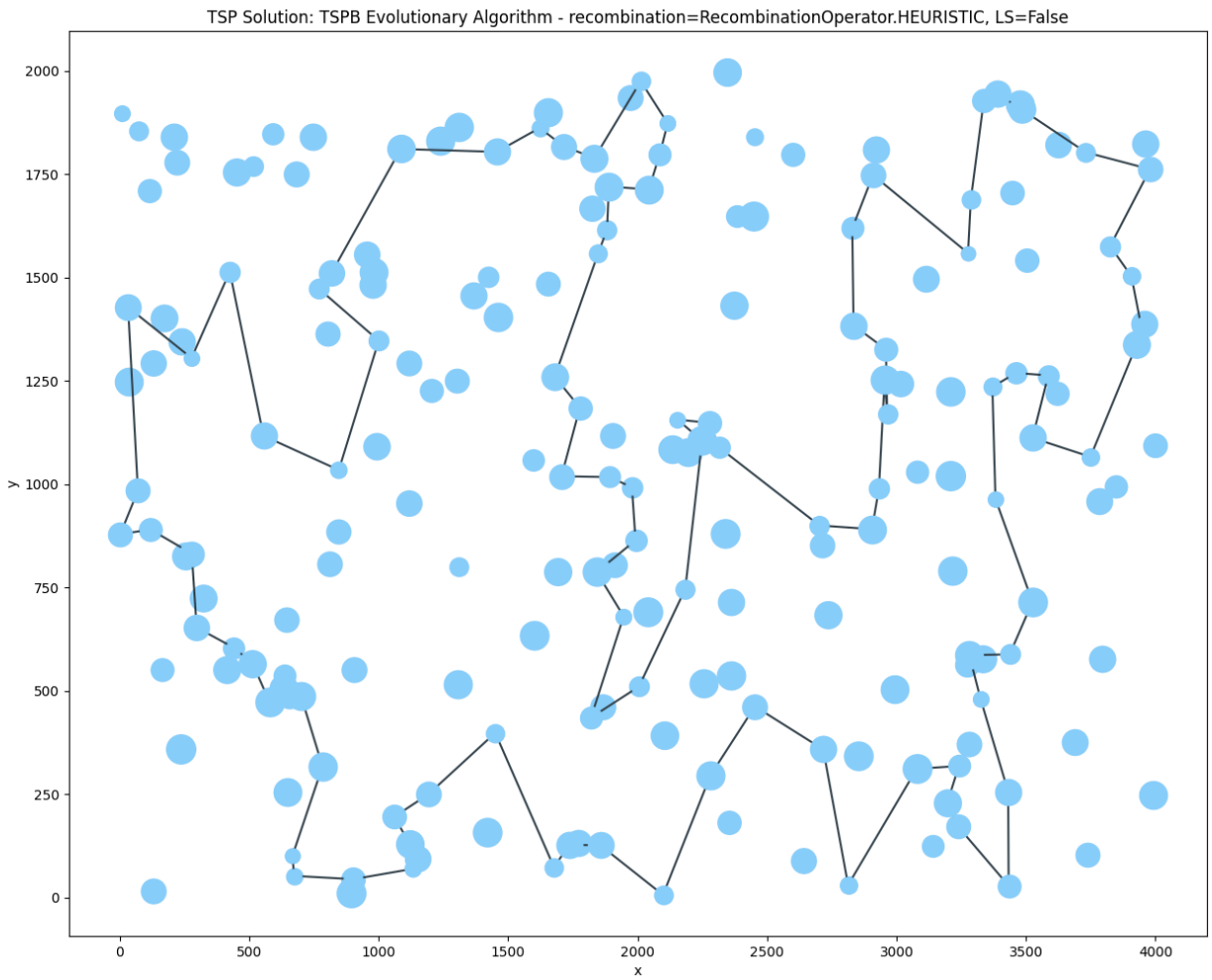
Dataset A:

Best solution: [119, 40, 81, 196, 179, 92, 129, 57, 145, 78, 31, 56, 113, 175, 171, 16, 25, 44, 120, 2, 75, 86, 101, 1, 97, 152, 26, 100, 53, 154, 180, 63, 79, 133, 135, 70, 127, 123, 162, 151, 51, 118, 59, 65, 116, 43, 42, 184, 35, 84, 112, 4, 190, 10, 177, 54, 48, 160, 34, 181, 146, 22, 18, 108, 69, 159, 193, 41, 139, 115, 46, 68, 140, 93, 117, 0, 143, 183, 89, 186, 23, 137, 176, 80, 94, 124, 148, 9, 62, 102, 144, 14, 49, 178, 106, 52, 55, 185, 165, 90]



Dataset B:

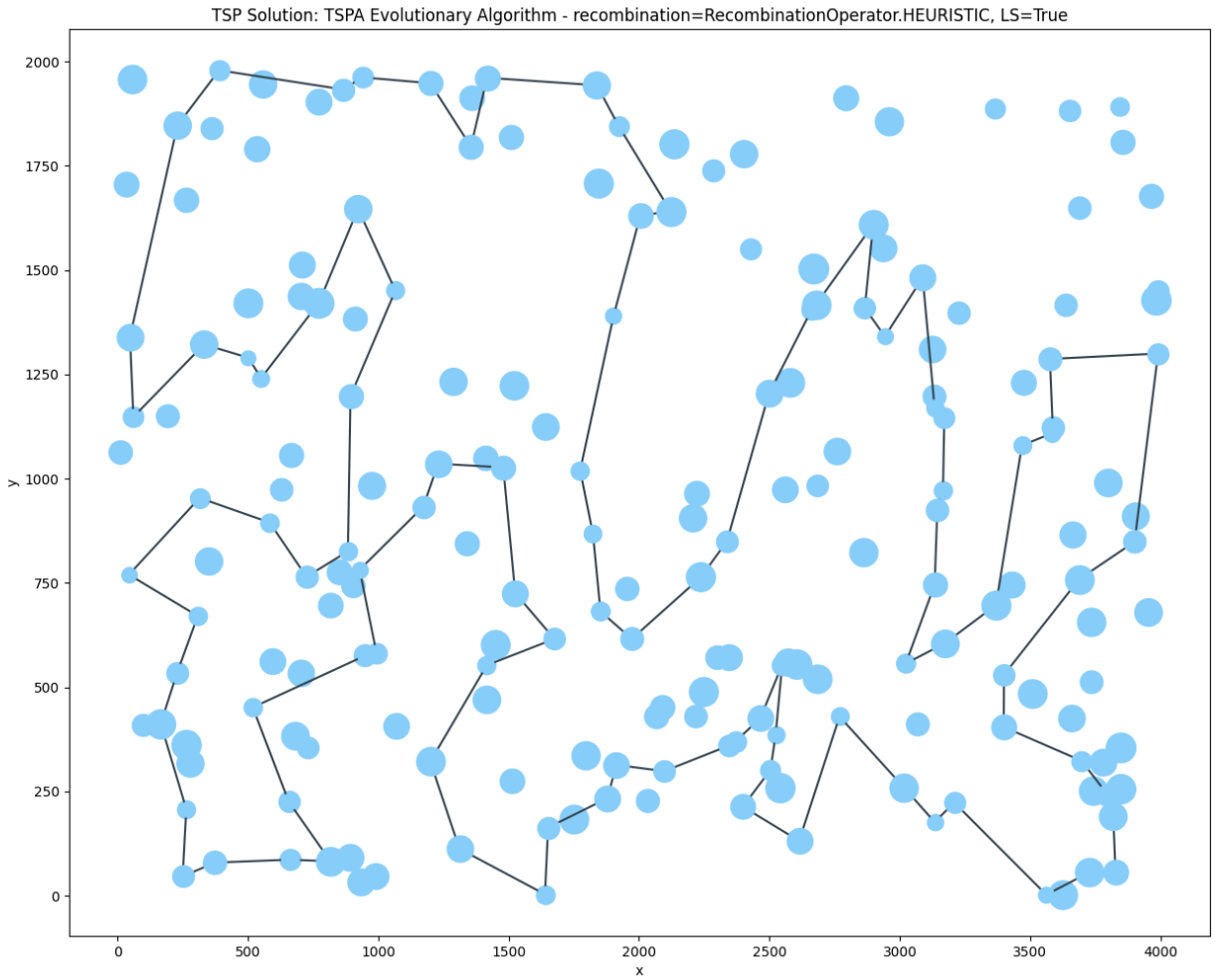
Best solution: [160, 144, 104, 8, 21, 82, 111, 0, 29, 109, 35, 143, 106, 124, 18, 62, 55, 34, 170, 152, 183, 140, 4, 149, 28, 20, 60, 148, 47, 94, 66, 179, 185, 99, 130, 95, 86, 166, 194, 176, 113, 114, 137, 127, 89, 103, 163, 187, 153, 81, 77, 141, 91, 61, 36, 177, 5, 45, 142, 78, 175, 162, 80, 190, 136, 73, 54, 31, 193, 117, 198, 156, 1, 16, 27, 38, 63, 135, 122, 131, 121, 51, 90, 191, 147, 6, 188, 169, 132, 70, 3, 15, 145, 13, 195, 168, 139, 11, 138, 33]



Evolutionary Algorithm heuristic, with Local Search

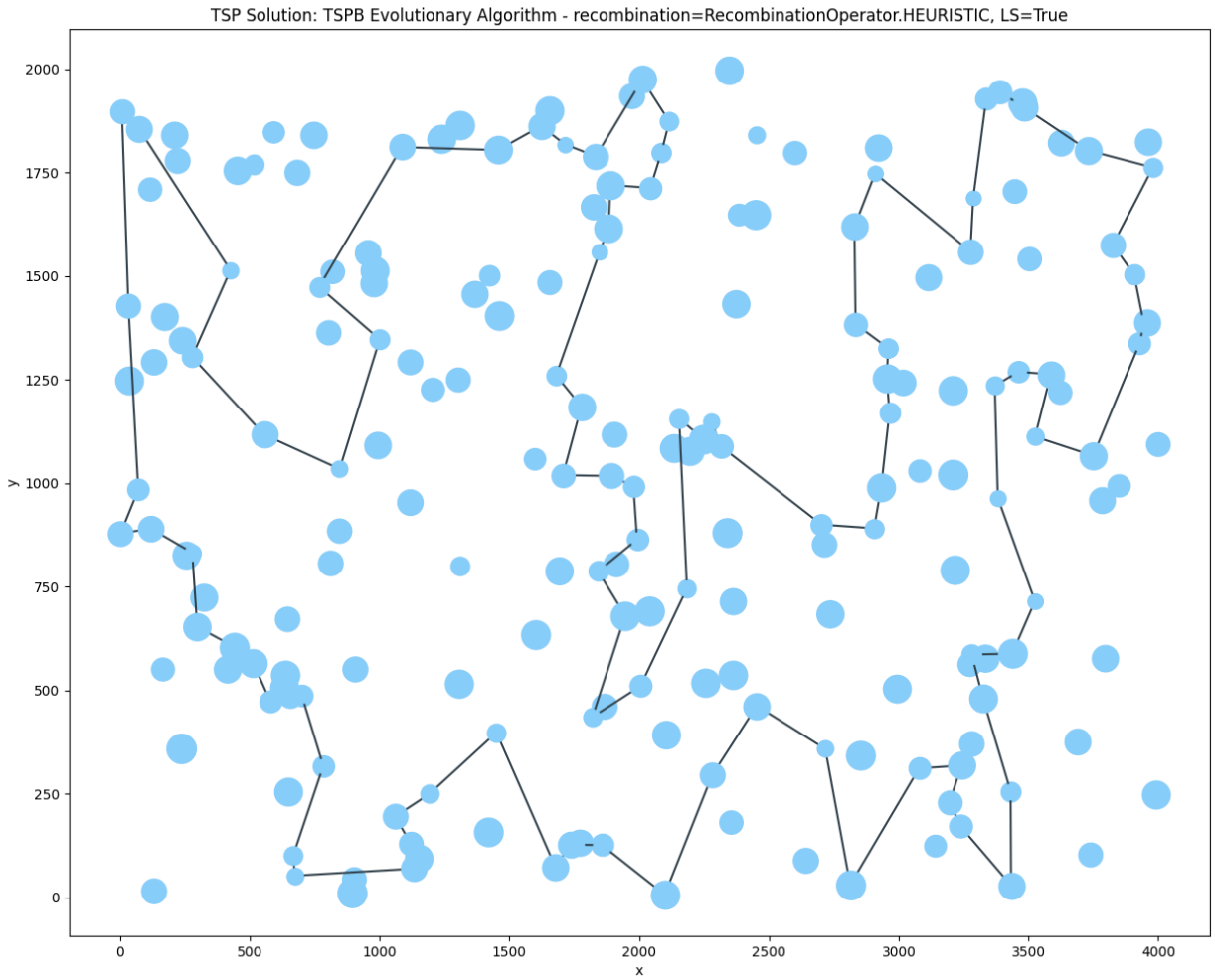
Dataset A:

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



Dataset B:

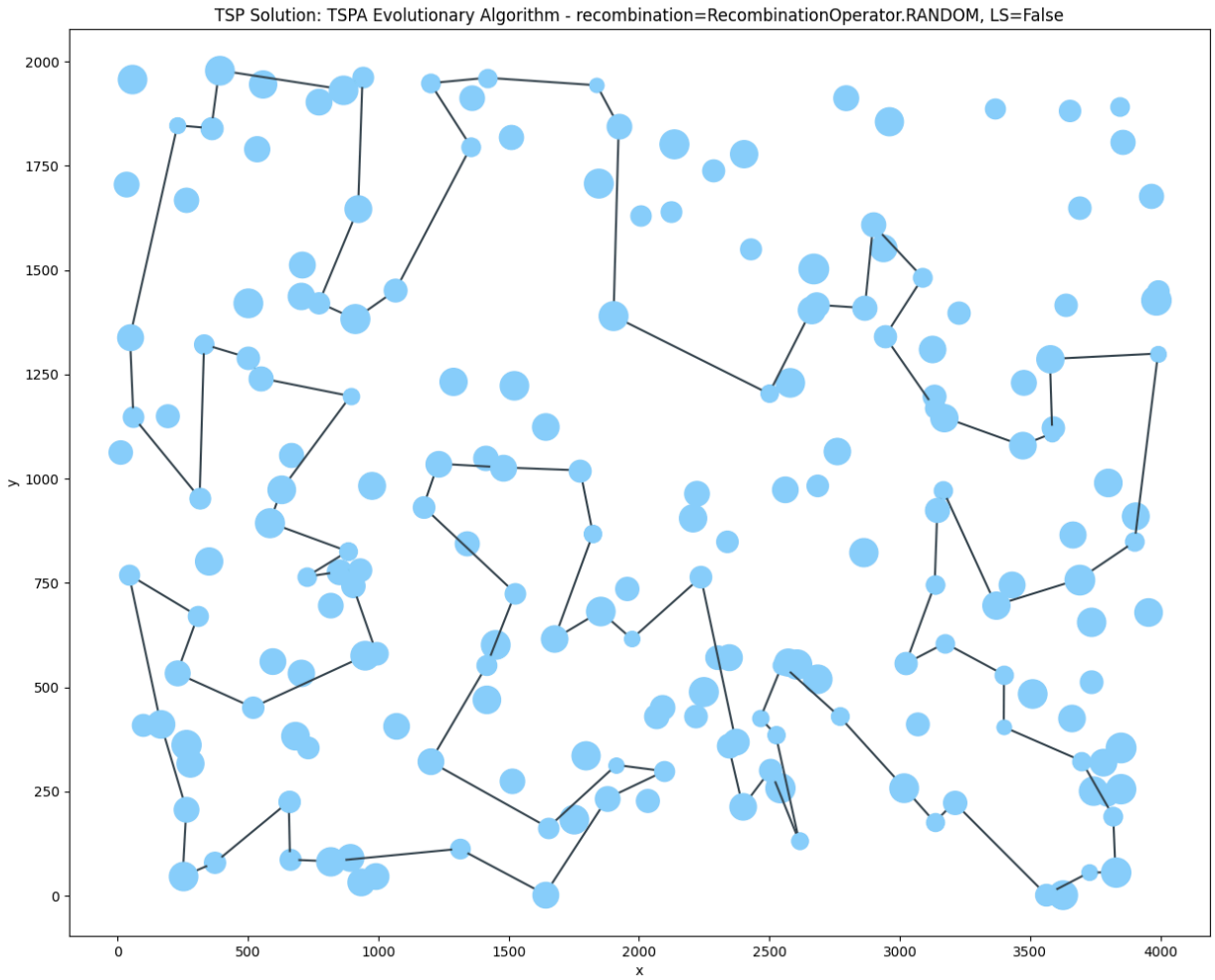
Best solution: [144, 104, 8, 21, 82, 111, 29, 0, 109, 35, 143, 106, 124, 62, 18, 55, 34, 170, 152, 183, 140, 4, 149, 28, 20, 60, 148, 47, 94, 66, 179, 185, 99, 130, 95, 86, 166, 194, 176, 113, 114, 137, 127, 89, 103, 163, 187, 153, 81, 77, 141, 91, 61, 36, 177, 5, 45, 142, 78, 175, 80, 190, 136, 73, 54, 31, 193, 117, 198, 156, 1, 16, 27, 38, 63, 40, 107, 122, 135, 131, 121, 51, 90, 147, 6, 188, 169, 132, 70, 3, 15, 145, 13, 195, 168, 139, 11, 138, 33, 160]



Evolutionary Algorithm random, without Local Search

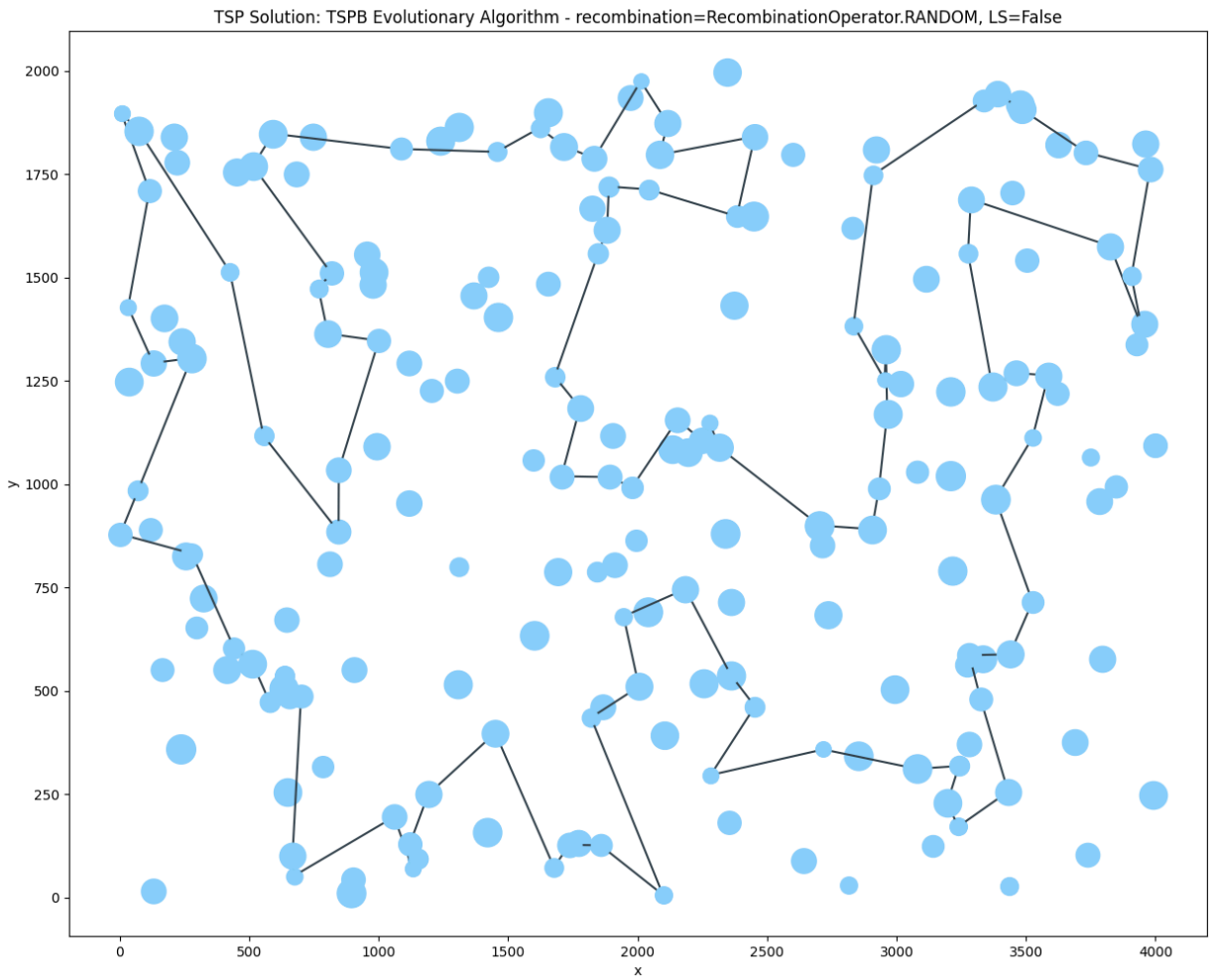
Dataset A:

Best solution: [116, 43, 105, 65, 47, 149, 131, 184, 48, 160, 34, 54, 177, 10, 190, 84, 4, 112, 127, 70, 154, 53, 180, 135, 123, 162, 151, 59, 118, 51, 176, 80, 133, 79, 63, 94, 26, 100, 86, 101, 75, 1, 97, 152, 2, 120, 44, 25, 16, 171, 175, 113, 31, 78, 145, 92, 129, 57, 55, 52, 179, 196, 81, 90, 165, 40, 185, 106, 178, 49, 14, 144, 102, 62, 9, 148, 137, 89, 183, 143, 117, 0, 46, 198, 139, 68, 93, 140, 108, 69, 18, 22, 146, 181, 159, 193, 41, 115, 5, 42]



Dataset B:

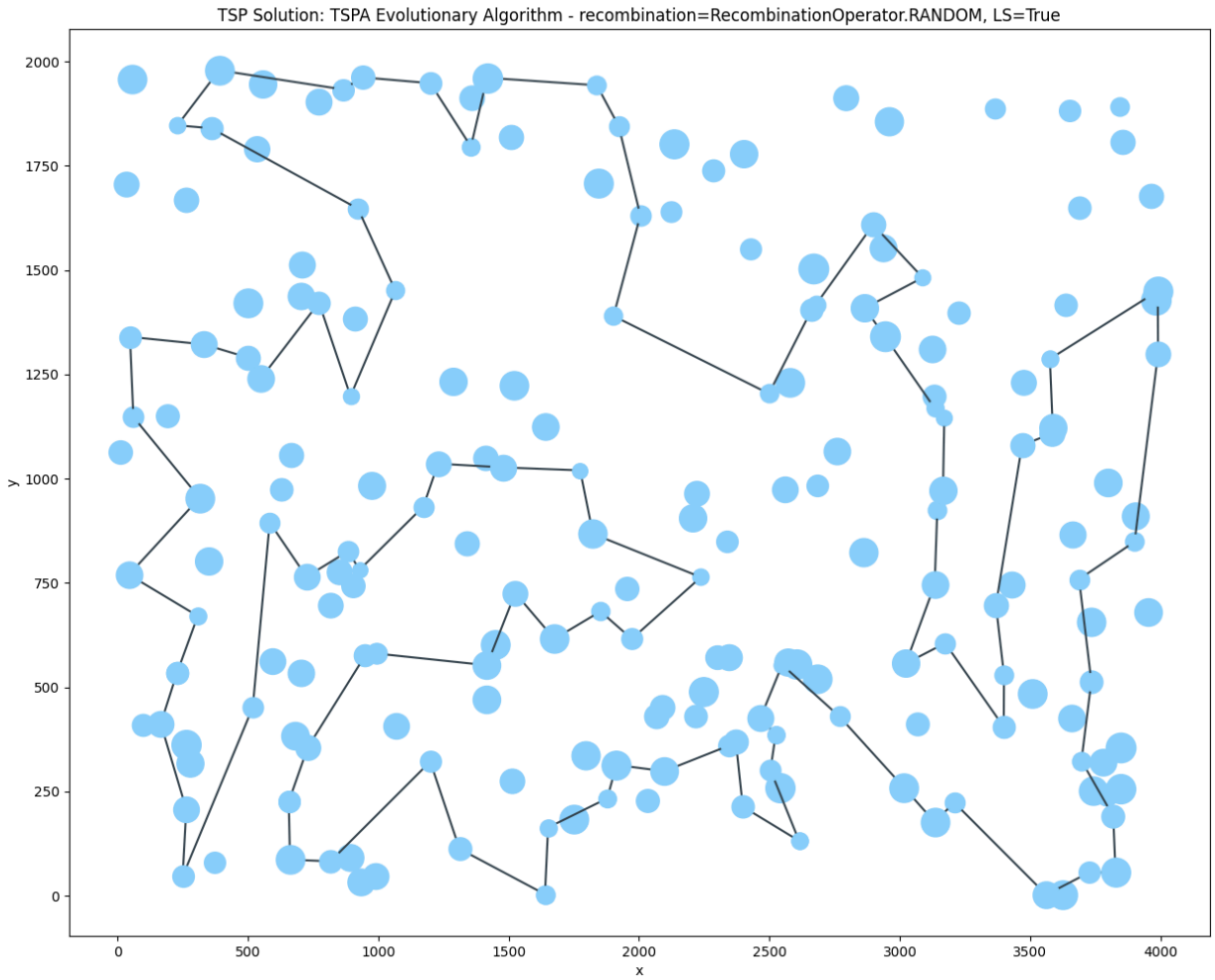
Best solution: [131, 122, 107, 40, 100, 63, 102, 135, 38, 27, 1, 198, 117, 193, 31, 54, 73, 190, 80, 45, 175, 78, 142, 5, 177, 36, 61, 91, 141, 21, 82, 8, 111, 14, 81, 77, 153, 163, 103, 89, 127, 114, 113, 176, 194, 166, 86, 185, 99, 130, 95, 183, 140, 148, 94, 47, 60, 20, 28, 149, 4, 152, 34, 18, 55, 62, 124, 106, 143, 35, 109, 0, 29, 160, 33, 138, 11, 139, 168, 195, 13, 145, 189, 155, 15, 3, 70, 132, 169, 188, 6, 147, 10, 133, 191, 90, 125, 51, 121, 112]



Evolutionary Algorithm heuristic, with Local Search

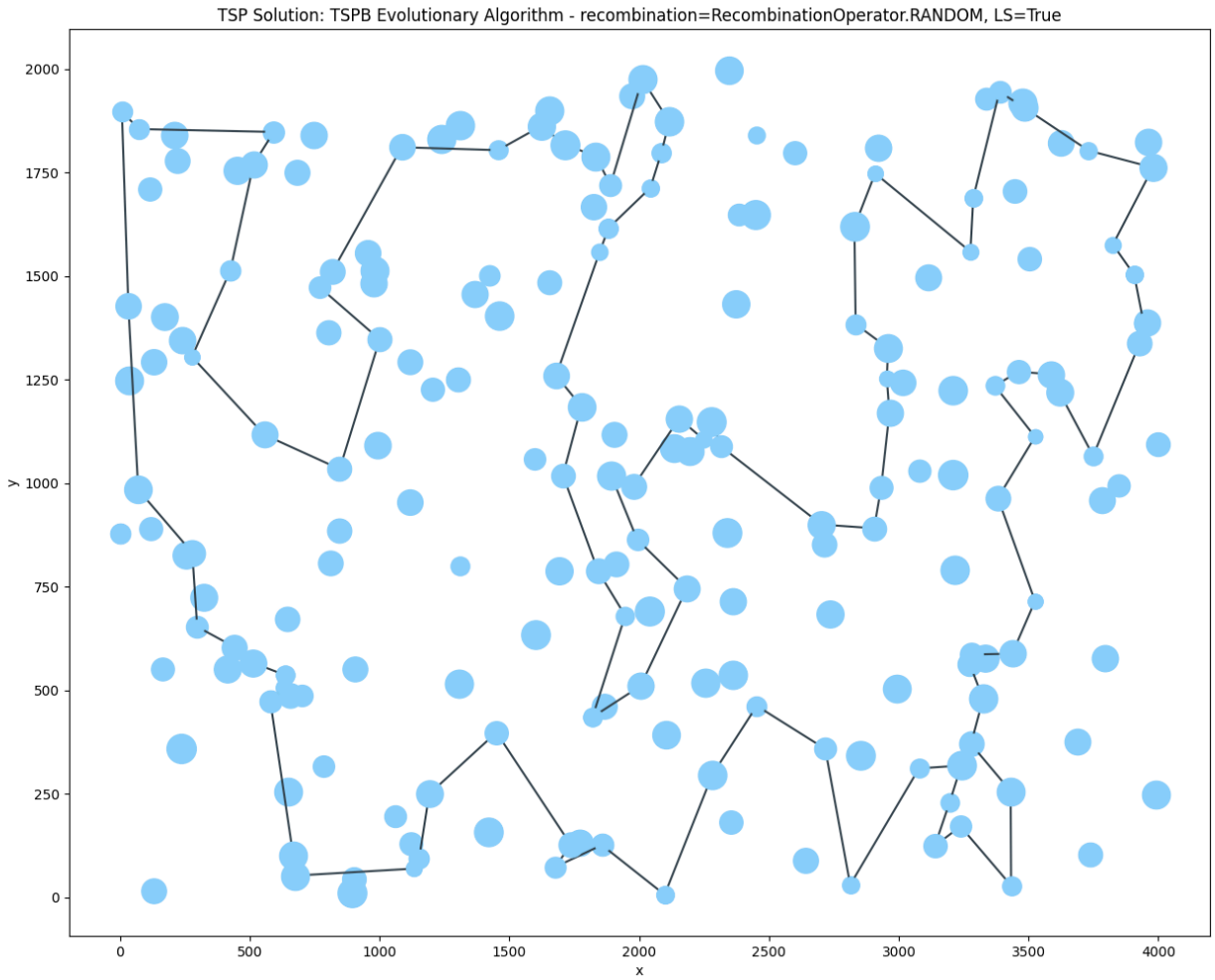
Dataset A:

Best solution: [154, 135, 70, 127, 123, 112, 4, 84, 35, 131, 149, 162, 151, 133, 79, 63, 94, 80, 176, 51, 118, 59, 65, 116, 43, 42, 184, 10, 177, 54, 48, 160, 34, 181, 146, 22, 159, 193, 41, 139, 115, 46, 68, 69, 18, 108, 140, 93, 117, 0, 143, 183, 89, 23, 137, 148, 9, 62, 144, 14, 102, 49, 178, 106, 52, 55, 57, 129, 92, 78, 145, 179, 185, 40, 119, 165, 27, 90, 81, 196, 157, 31, 113, 175, 171, 16, 25, 44, 120, 2, 152, 97, 1, 101, 75, 86, 26, 100, 53, 180]



Dataset B:

Best solution: [91, 141, 77, 81, 153, 187, 163, 103, 89, 165, 127, 137, 114, 26, 113, 180, 176, 194, 166, 86, 185, 95, 130, 99, 22, 179, 66, 94, 47, 148, 60, 20, 28, 149, 140, 183, 152, 170, 34, 55, 18, 62, 124, 106, 143, 35, 109, 0, 29, 160, 33, 144, 111, 82, 21, 8, 104, 138, 11, 139, 168, 195, 145, 15, 3, 70, 13, 132, 169, 188, 6, 147, 191, 90, 51, 121, 131, 135, 122, 133, 10, 107, 40, 63, 38, 1, 156, 198, 117, 54, 31, 193, 190, 80, 175, 78, 5, 177, 61, 36]



Conclusion:

- EA with heuristics and LS achieves best results among EA variants, producing high-quality solutions despite fewer iterations
- Standard EA (random, no LS) performs worst overall
- ILS remains the superior method
- EA effectiveness varies significantly with configuration, potentially matching ILS with optimal settings