Assignment 10: Own Method

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

Own method

The goal of this assignment is to improve the algorithms that were implemented earlier. Either a modification of previously implemented algorithms or a completely new method should be proposed to improve the average score and the best result obtained in previous experiments (on all instances) at the same time. The new algorithm needs to be described in detail. For example, mechanisms such as:

- generating better starting solutions,
- further improvements of the efficiency of local searches.
- global memory of moves evaluations,
- new mechanisms of candidate moves;
- other operators of neighborhood, perturbation, recombination,
- simultaneous use of different operators,
- use of machine learning mechanisms,
- ...

The proposed method should be precisely described and its algorithm should be provided in pseudocode. Report – analogous as before.

Simulated Annealing

Hyper parameters:

- Initial Temperature: higher values – more exploration, slower convergence; lower values – less exploration, faster convergence.
- Cooling Rate: fast cooling (0.8) – quick convergence but might get stuck; slow cooling (0.99) – better solutions but longer run time.
- Iterations per temperature: more iterations – better exploration at each temperature but longer run time; fewer iterations – less exploration but faster run time.

Initial temp = 10,000
Cooling rate = 0.99
Iterations per temperature = 1,000
Min temperature = 1
Max time = 120s

SA solution update criteria:

```
If current solution is better OR accept with probability
e^log((current cost - new cost) / temperature)
```

Pseudo code:

```
Generate initial solution randomly
Mark as the best solution

Run while temperature hasn't cooled off (above the min limit):
      Run for a set number of iterations per temperature:
            Generate a new solution
            Calculate delta (new cost - current cost)

            If delta < 0 OR random(0,1 ) < exp(-delta/temperature):
                  Update the current solution

                  If current solution is better than the best solution:
                        Update the best solution

      Update the temperature

 Return the best solution
```

Hybrid Ant Colony Optimization + Local Search

ACO hyper parameters:

- Pheromone importance (alpha): higher – more exploitation of good parts; lower – more exploration.
- Heuristic information importance (beta): higher – more greedy behavior (shorter distances); more random exploration.
- Pheromone evaporation rate (rho): higher – faster forgetting poor solutions; lower – longer memory of all solutions.
- Exploitation vs exploration probability (q0): higher – more exploitation; lower – more exploration.
- Number of ants: higher – more exploration, longer run time; lower – less exploration, faster run time.

```
n_ants=45,
   alpha=1.3,
   beta=2.8,
    rho=0.12,
    q0=0.82,
    max_time=450,
```

Pseudo code:

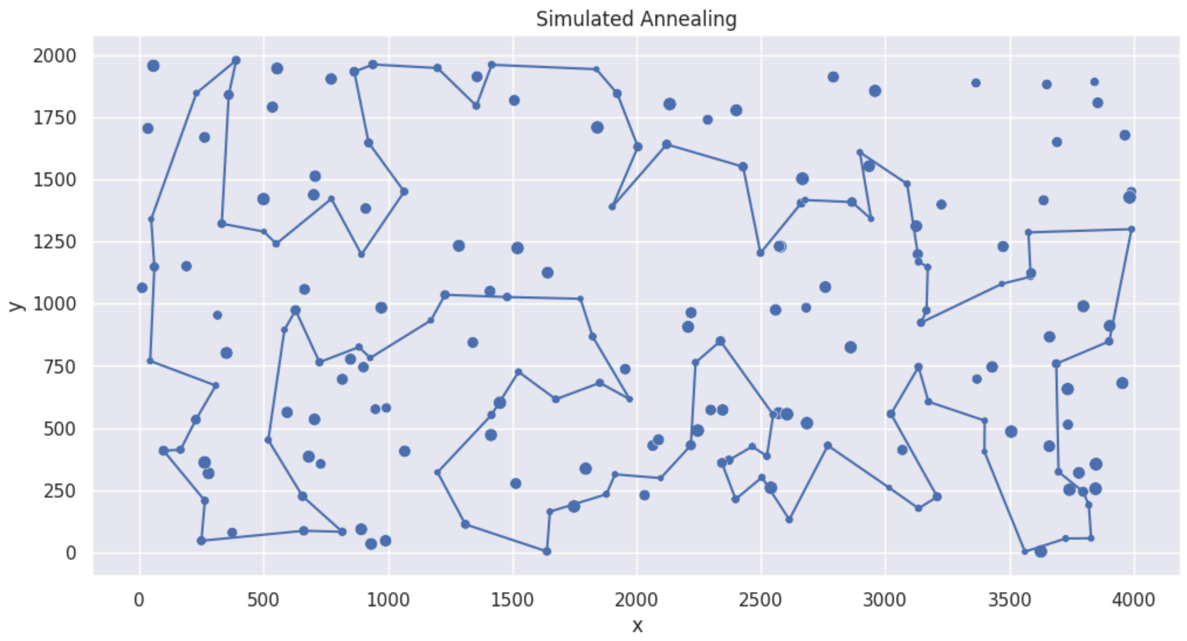
```
Function ConstructSolution():
      Initialize an empty tour
      While tour is not complete:
            If random(0, 1) < q0:
                  Set next as city with max(pheromone^alpha * eta^beta)
            Else:
                  probabilistic selection based on pheromone and eta

Main algorithm:
      Set best tour as empty
      Run until time limit is off:
            For each ant:
                  Build a tour using ConstructSolution()
                  Improve the tour using LocalSearch()

                  If best solution is found:
                        Update best solution

            Update pheromones
```

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,	70664.3(70082-71423)	44494.85(43772-45563)

without Local Search		
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)
Simulated Annealing	71299.44(72463-70176)	45529.82(47077-43998)
Ant Colony Optimization with Local Search	79006.2(81021-76995)	49238.6(49564-48544)

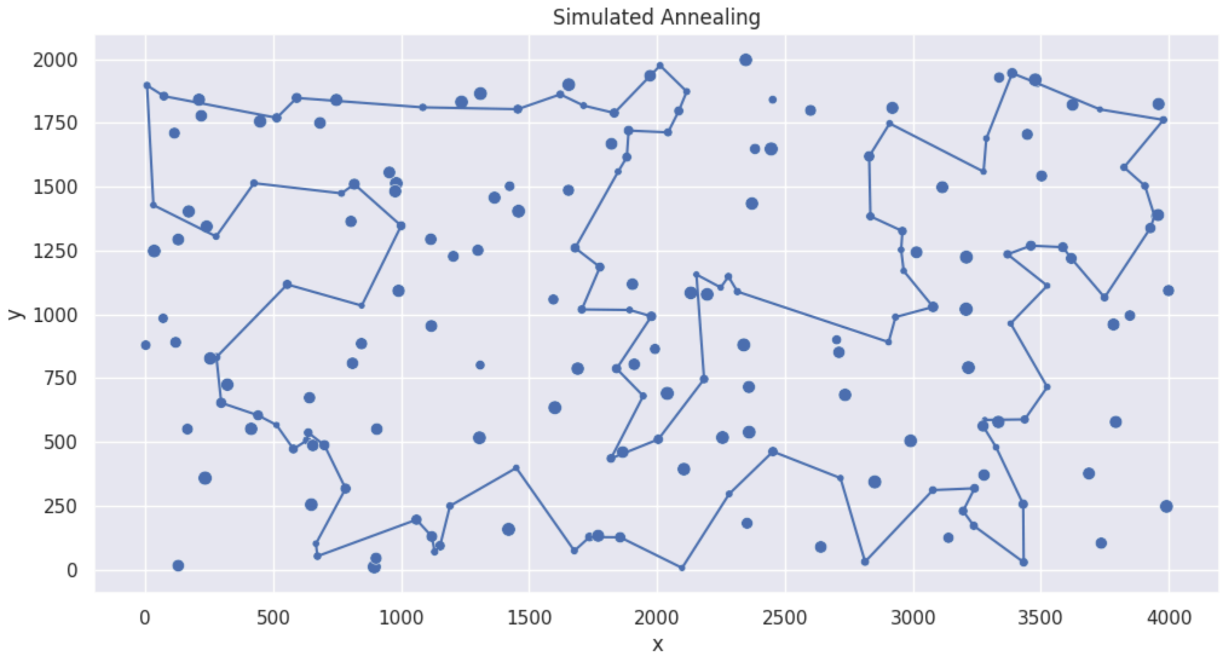
Best solution Simulated Annealing (Dataset A):

```
[178, 106,  52,  55, 185,  40, 165,  90,  81, 196,  31,  56, 113,
        175, 171,  16,  78, 145,  92,  57, 129,  25,  44, 120,   2,  75,
        101,  86, 100,  26,  97,   1, 152, 124,  94, 121,  53, 180, 154,
        135,  70, 127, 123, 162, 151, 133,  79,  63,  80, 176,  51, 118,
         59,  65, 116,  43,   5,  42, 184,  84, 112,   4,  10, 177,  30,
         54,  48, 160,  34, 146,  22,  18, 108,  69, 159, 193,  41, 139,
        115,  46,  68, 140,  93, 117,   0, 143, 183,  89,  23, 137, 186,
         15, 148,   9,  62, 102,  49, 144,  14,   3]
```



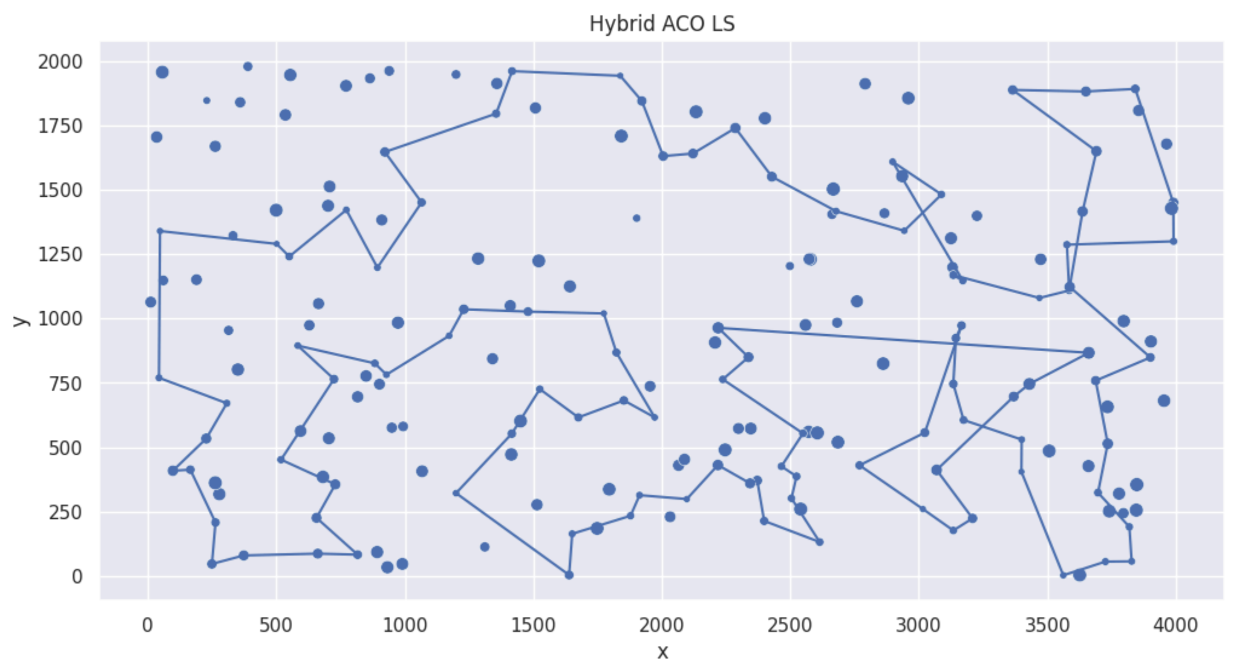
Best solution Simulated Annealing (Dataset B):

```
[  0, 109,  35, 106, 124, 128,  62,  18,  55,  34, 170, 152, 183,
   140, 149,  28,  20,  60, 148,  47,  94,  66, 179,  22,  99, 130,
    95, 185,  86, 166, 194, 176, 180, 113, 114, 137, 127,  89, 103,
   163, 187, 153,  81,  77, 141,  91,  61,  36, 177,   5,  78, 175,
   142,  45,  80, 190, 136,  73,  54,  31, 193, 117, 198, 156,   1,
   131, 121,  51, 191,  90, 122, 135,  63,  40, 107, 133,  10, 147,
     6, 188, 169, 132,  70,   3,  15, 145,  13, 195, 168, 139,  11,
   138,  33, 160, 104,   8,  21,  82, 111,  29]
```



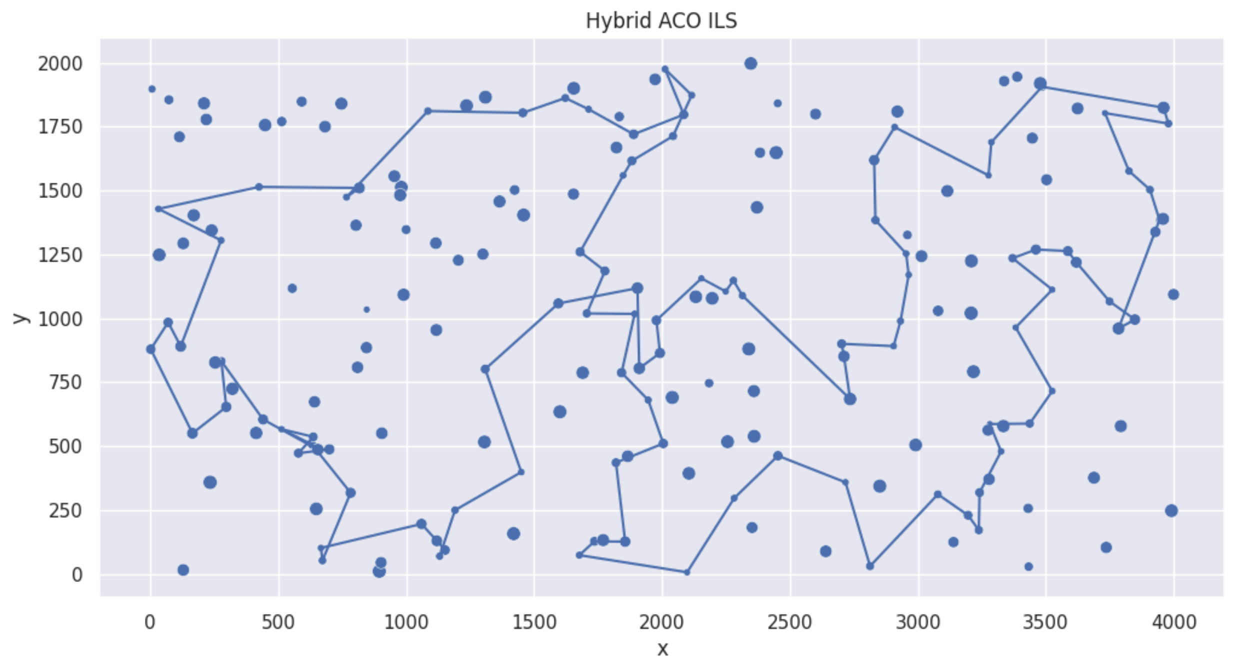
Best solution ACO + LS (Dataset A):

[12,124,94,152,97,1,101,75,86,26,100,121,53,180,154,135,70,123,162,151,133,79,63,80,176,51,118,59,65,116,42,43,184,35,84,112,4,190,10,177,54,30,48,160,34,22,193,41,139,115,46,68,0,143,183,89,23,186,114,15,62,49,14,144,106,178,185,40,165,90,27,164,7,21,95,39,119,81,196,157,31,113,175,171,16,78,145,92,57,55,52,129,2,120,44,25,82,179,91,169]



Best solution ACO + LS (Dataset B):

[191,90,147,6,188,169,13,15,70,3,145,195,168,139,11,138,33,104,8,82,21,91,61,36,141,77,81,
153,187,163,89,127,103,113,176,194,166,86,185,95,130,99,22,179,172,52,94,47,148,20,60,23,
28,140,183,152,170,34,18,62,124,106,143,159,119,35,109,0,29,160,144,56,49,182,25,177,5,17
5,78,142,45,190,80,136,31,117,54,193,73,164,198,1,156,42,27,38,16,135,63,122]


Hybrid ACO ILS
y
x