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.

Pseudocode

```
FUNCTION GenerateSolution(population, withLS, avgTime):
    startTime = current time

    WHILE current time - startTime <= avgTime:
        parent1 = random index from population
        parent2 = random index from population
        WHILE parent1 == parent2:
            parent2 = random index from population

        child = Recombination(parent1, parent2)

        IF withLS:
            child = LocalSearch.GenerateSolution(child, Steepest, EXCHANGE_EDGES)

        childObj = CalculateDistance(child)

        IF childObj < worst objective in population AND child not in population:
            REMOVE worst from population
            ADD child to population

    bestSolution = empty list
    WHILE population not empty:
        bestSolution = REMOVE best from population

    RETURN bestSolution
```

```
FUNCTION Recombination(parent1, parent2):
    child = empty list
    FOR i FROM 0 TO size of parent1:
        index = position of parent1[i] in parent2
        prevIndex = index - 1
        nextIndex = index + 1
        IF prevIndex == -1: prevIndex = size of parent2 - 1
        IF nextIndex == size of parent2: nextIndex = 0

        prevI = i - 1
        nextI = i + 1
        IF prevI == -1: prevI = size of parent1 - 1
        IF nextI == size of parent1: nextI = 0

        IF index != -1 AND (
            parent1[prevI] == parent2[prevIndex] OR
            parent1[nextI] == parent2[nextIndex] OR
            parent1[prevI] == parent2[nextIndex] OR
            parent1[nextI] == parent2[prevIndex]
        ):
            ADD parent1[i] TO child

    child = Repair(child)
    RETURN child
```

Results

Method	Instance	Min Distance	Max Distance	Average Distance	Execution Time (ms)	Average Time (ms)	Average iterations
MSLS	TSPA	71615	78519	73886.605	2717553	16500.5	20
	TSPB	45111	51523	48358.664	1658511	18915.8	20
ILS	TSPA	69400	71285	70198.045	2004059	--	2890.785
	TSPB	43649	46946	44650.46	2003919	--	2964.39
LOCAL-SEARCH-STEEPEST	TSPA	71756	78077	74003.145	86739	--	--
	TSPB	46019	51211	48491.05	96498	--	--
GREEDY 2-REGRET WEIGHTED	TSPA	71,108	73,395	72,130.045	--	--	--
	TSPB	47,144	55,700	50,919.565	--	--	--
LNS	TSPA	69322	70801	70087.6	--	--	2459.15

Method	Instance	Min Distance	Max Distance	Average Distance	Execution Time (ms)	Average Time (ms)	Average iterations
	TSPB	43593	45139	44391.7	--	--	2598.25
LNS with LS	TSPA	69378	70572	69917.15	--	--	1487.3
	TSPB	43584	45047	44362.7	--	--	1539.4545
HAE with LS	TSPA	69372	70718	69912.54	--	--	
	TSPB	43715	44649	44212.03	--	--	
HAE without LS	TSPA	69412	70913	70017.54	--	--	
	TSPB	43745	44819	44323.1	--	--	

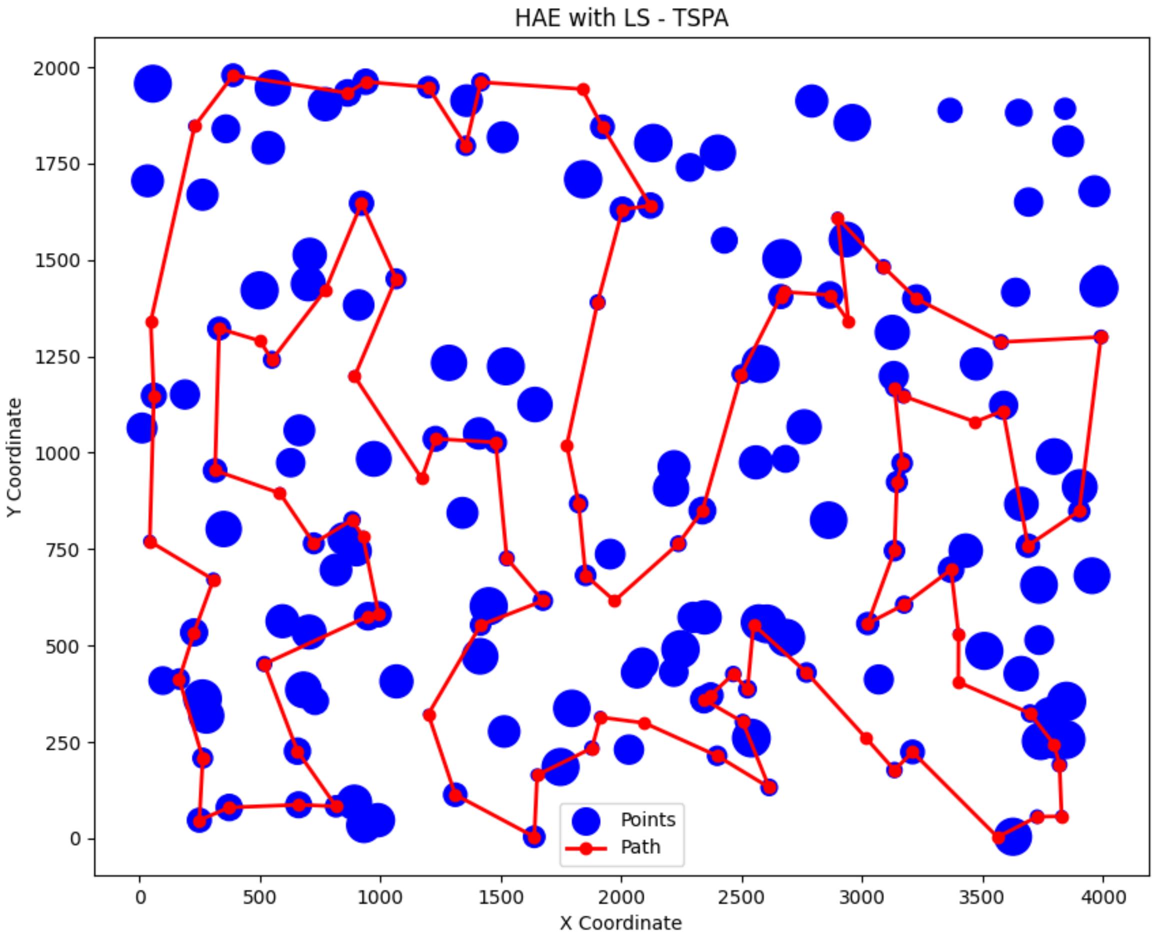
Summary

The Hybrid Evolutionary Algorithm shows competitive performance with methods like ILS, LNS, and Greedy 2-Weighted Regret. Specifically, the minimum and maximum distances in HAE are generally comparable to or slightly better than those in other methods, such as MSLS or LNS.

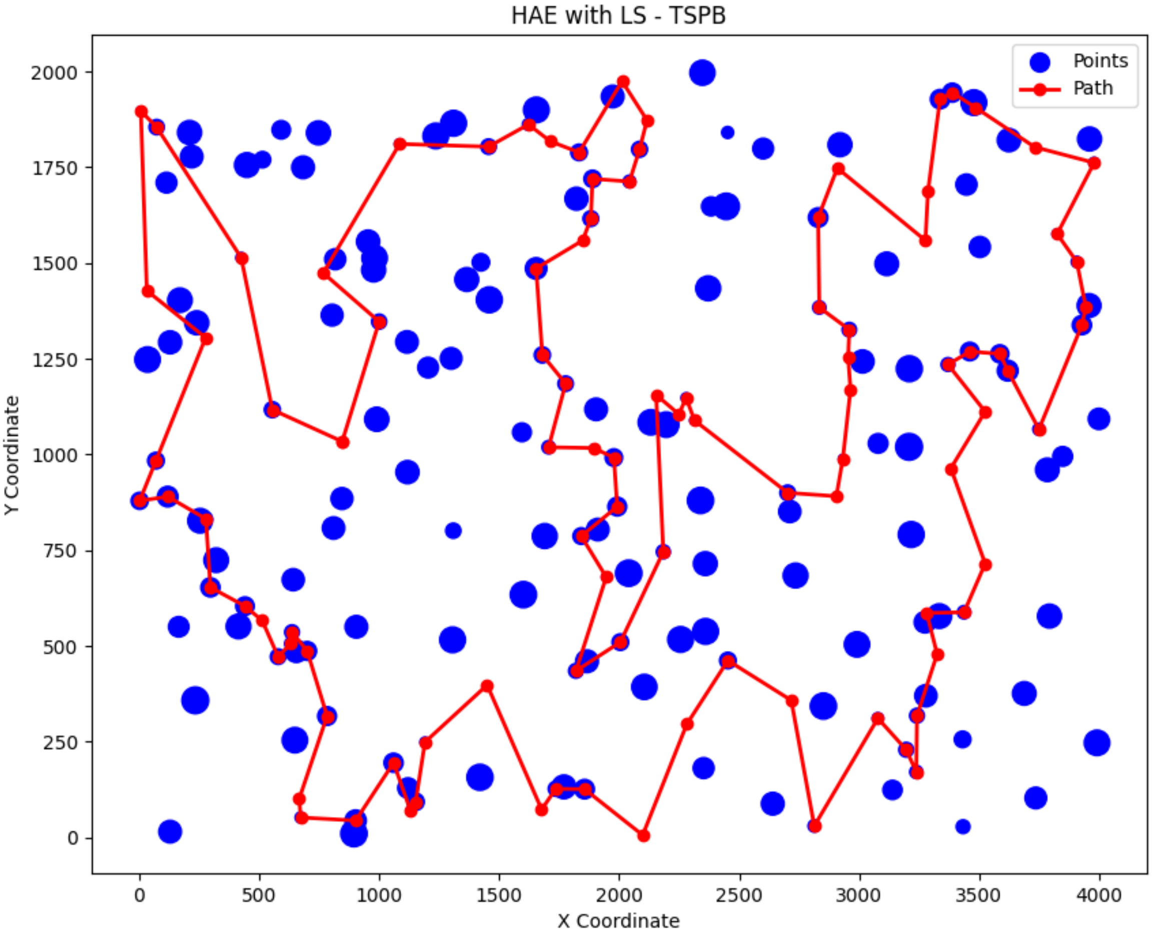
Graphs

HAE with LS

TSPA

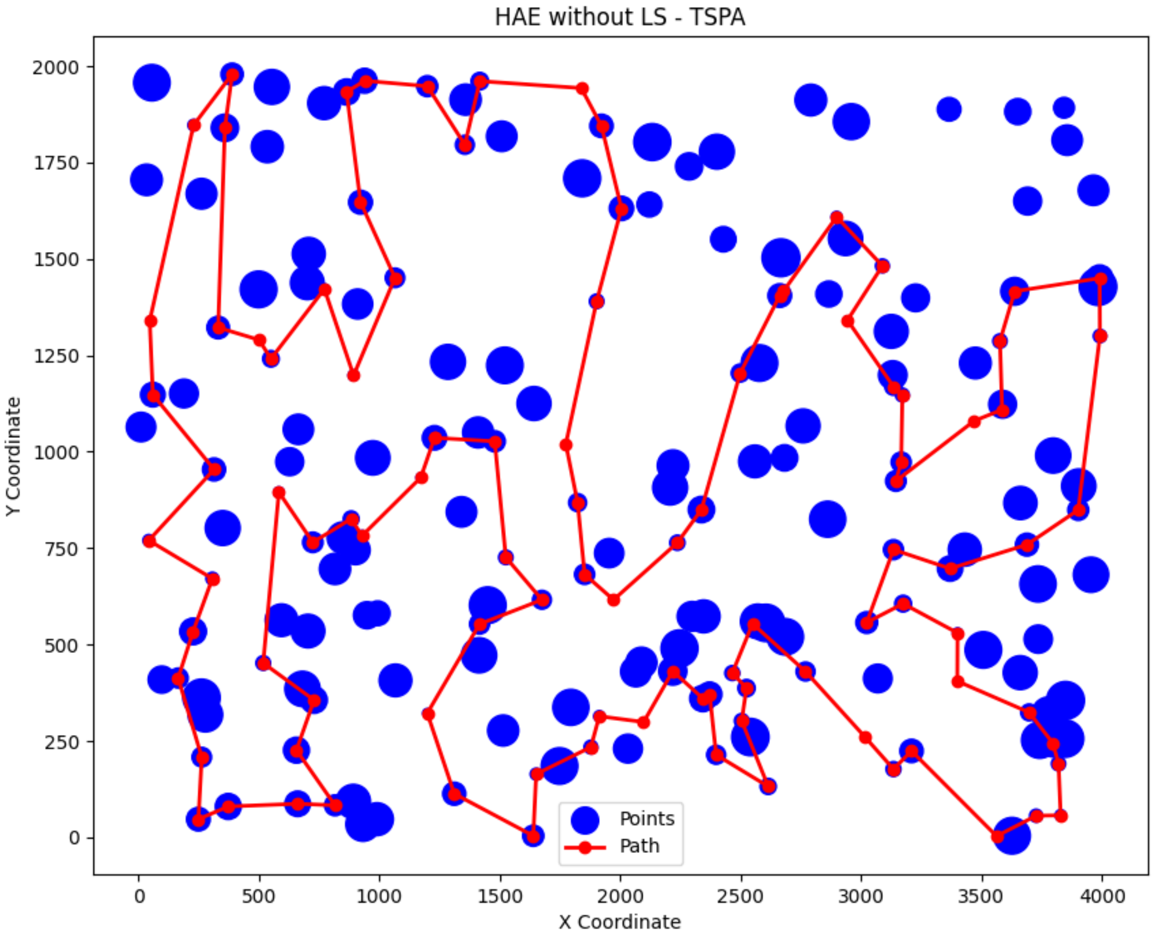

HAE with LS - TSPA
Y Coordinate
X Coordinate
0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000
2500
3000
3500
4000
Points
Path

TSPB

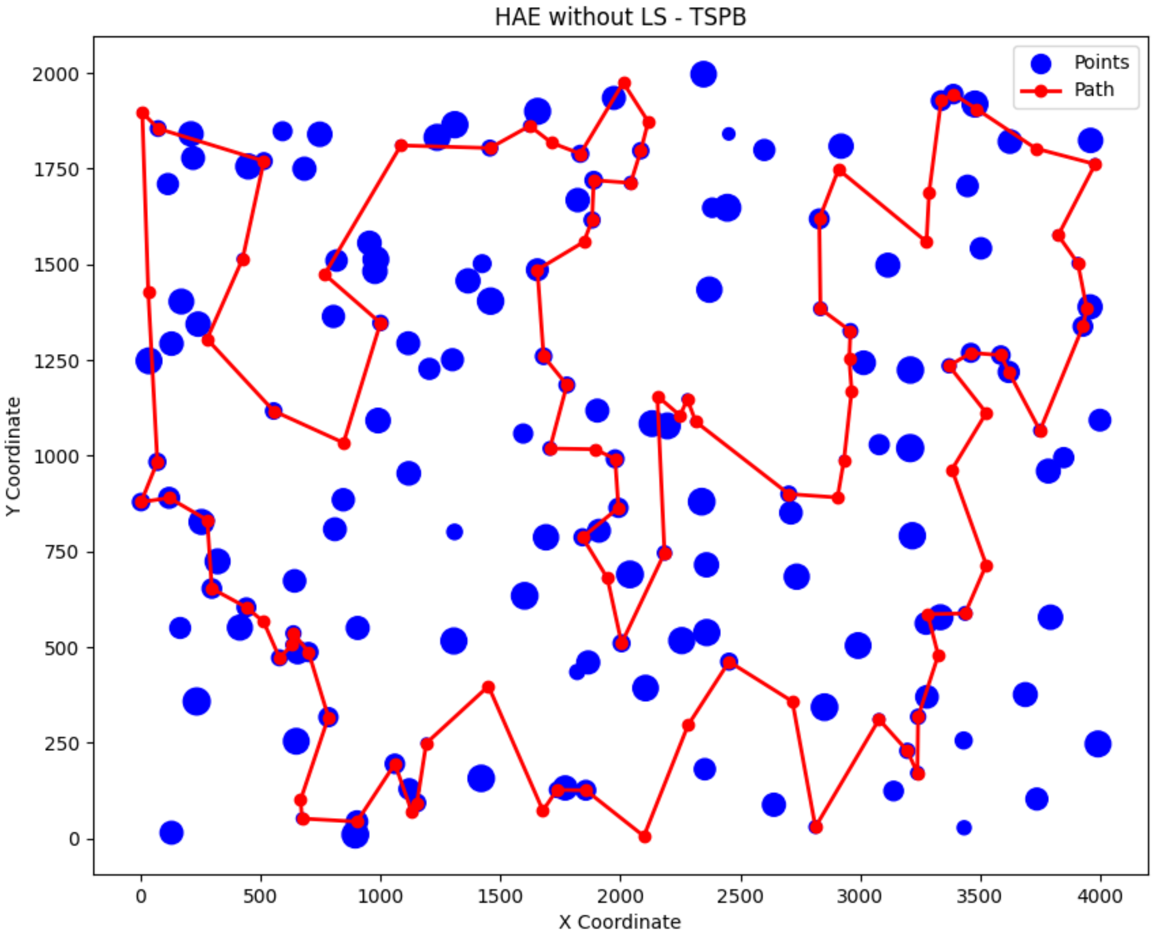


HAE without LS

TSPA


HAE without LS - TSPA
Y Coordinate
X Coordinate
2000
1750
1500
1250
1000
750
500
250
0
0
500
1000
1500
2000
2500
3000
3500
4000
● Points
●— Path

TSPB


HAE without LS - 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