assignment 3

Link to the source code: GitHub Repository

Authors

- Wojciech Bogacz 156034
- Krzysztof Skodzik 156039

Problem Description

The problem considered in this project involves a set of nodes, each represented by three integer values: two coordinates (x,y), (x,y) that define the node's position in a two dimensional plane, and a cost value associated with the node. The objective is to select exactly 50% of all available nodes (if the total number of nodes is odd, the number of selected nodes is rounded up) and construct a Hamiltonian cycle — that is, a closed path that visits each selected node exactly once and returns to the starting node — with the goal to minimize the sum of two components: 1. The total length of the constructed cycle, and 2. The total cost of all selected nodes. The distances between nodes are calculated using the Euclidean metric, rounded to the nearest integer.

Implemented Algorithms

Local Search Greedy

Local Search Steepest

Results

Comparison of previous methods results

Instance A

Method	Best	Worst	Average
random	215188	291323	264017.17
nn_end	89198	123123	104012.785
nn_anywhere	71498	74410	72635.71
greedy cycle	72639	75639	72639.0
2-Regret Insertion	101812	115628	110710.93
Weighted Sum (a=1.0b. β=1.0b)	71108	73458	72136.85

Instance B

Method	Best	Worst	Average
random	235224	2347798	2142763.25
nn_end	62606	77453	69764.43
nn_anywhere	49691	57324	51408.965
greedy cycle	50243	50243	50243.0
2-Regret Insertion	66265	77972	71454.77
Weighted Sum (a=1.0b. β=1.0b)	47144	55700	50968.82

Local Search results

Instance A

Method	best	worst	average
greedy_intra-edges_start-greedy	70843	72648	71594.850
greedy_intra-edges_start-random	71564	76328	73528.85
greedy_intra-nodes_start-greedy	70887	73027	71739.395
greedy_intra-nodes_start-random	70677	99835	80999.125
steepest_intra-edges_start-greedy	70518	72614	71463.08
steepest_intra-edges_start-random	71246	76153	73699.995
steepest_intra-nodes_start-greedy	70826	73884	71615.95
steepest_intra-nodes_start-random	81322	95342	87939.335

Instance B

Method	best	worst	average
greedy_intra-edges_start-greedy	46178	55471	50171.07
greedy_intra-edges_start-random	45537	53087	48534.865
greedy_intra-nodes_start-greedy	46373	55395	50347.535
greedy_intra-nodes_start-random	53417	71474	61451.295
steepest_intra-edges_start-greedy	45867	54814	49987.205
steepest_intra-edges_start-random	45993	55416	49195.645
steepest_intra-nodes_start-greedy	46371	55395	50201.47
steepest_intra-nodes_start-random	53686	71546	62955.885

Best path for the instance A

Best path for the instance B

Time

Instance A

Method	min	max	average
greedy_intra-edges_start-greedy	0.00	0.006	0.003
greedy_intra-edges_start-random	0.062	0.095	0.080
greedy_intra-nodes_start-greedy	0.000	0.005	0.002
greedy_intra-nodes_start-random	0.066	0.118	0.085
steepest_intra-edges_start-greedy	0.001	0.005	0.002
steepest_intra-edges_start-random	0.039	0.058	0.045
steepest_intra-nodes_start-greedy	0.000	0.005	0.002
steepest_intra-nodes_start-random	0.049	0.142	0.071

Instance B

Method	min	max	average
greedy_intra-edges_start-greedy	0.3	0.01	0.004
greedy_intra-edges_start-random	0.064	0.102	0.083
greedy_intra-nodes_start-greedy	0.3	0.007	0.003
greedy_intra-nodes_start-random	0.058	0.137	0.082
steepest_intra-edges_start-greedy	0.001	0.01	0.004
steepest_intra-edges_start-random	0.038	0.054	0.045
steepest_intra-nodes_start-greedy	0.001	0.007	0.004
steepest_intra-nodes_start-random	0.048	0.086	0.063

Conclusions

The conducted experiments demonstrated that both the greedy and Steepest local search algorithms significantly improved the initial solutions. The problem combines two classic optimization problems, respectivelyusing heuristic heuristics such as medians, nearest neighbor, and insertion methods. Among the tested variants, approaches starting from greedy-based initial solutions consistently yielded better and more stable results compared to random starts, while the steepest variant occasionally achieved marginally better results than the greedy one. Although the steepest variant was slightly more highly efficient, with execution times in the millisecond range, and achieved significant reductions in total cost for both instances A and B, Overall, local search proved to be a robust and efficient optimization strategy, and future work could focus on integrating metaheuristic or hybrid approaches to further enhance solution quality and avoid local minima.