Local Search Methods for Hamiltonian Cycle Problem

Evolutionary Computation Laboratory: Assignment 3

151927 Samuel Janas, samuel.janas@student.put.poznan.pl
151955 Bruno Urbaniak, bruno.urbaniak@student.put.poznan.pl
151960 Patryk Maciejewski, patryk.maciejewski.1@student.put.poznan.pl

1 Introduction

In this report, we explore the implementation and evaluation of local search methods applied to the Hamiltonian Cycle Problem (HCP). Building upon the previous studies where various greedy heuristics were analyzed, this report introduces two primary local search strategies: the **Steepest Descent** and the **Greedy Local Search**. These methods aim to iteratively improve upon an initial solution by exploring the neighborhood of solutions through specific move operations.

The primary objective is to minimize the objective function, which comprises the total cost of selected nodes and the total distance of the cycle formed. By leveraging local search techniques, we aim to enhance the quality of solutions obtained from initial heuristics, thereby achieving better optimization results.

2 Problem Description and Implementation

The Hamiltonian Cycle Problem (HCP) involves finding a closed loop that visits a subset of nodes exactly once. In this study, we focus on selecting 50% of the available nodes to form such a cycle while minimizing the combined objective function of total node costs and cycle distances.

2.1 Local Search Overview

Local search algorithms iteratively explore the neighborhood of the current solution to find better solutions. The two types of local search implemented in this study are:

- **Steepest Descent**: This method evaluates all possible moves within the neighborhood and selects the move that offers the most significant improvement in the objective function.
- **Greedy Local Search**: This approach examines moves in a randomized order and selects the first move that results in an improvement, without necessarily evaluating the entire neighborhood.

2.2 Neighborhood Structure

The neighborhood of a solution is defined by specific move operations, categorized into intra-route and inter-route moves:

- **Intra-route Moves**:
  - **Two-Nodes Exchange**: Swapping the positions of two nodes within the same route.
  - **Two-Edges Exchange**: Reversing the order of nodes between two edges within the same route.
- **Inter-route Moves**:
  - **Node Exchange**: Exchanging a selected node with an unselected node.

The combination of these moves defines the entire neighborhood for the local search algorithms.

2.3 Implementation Details

The local search methods are implemented in **Go**, building upon the existing codebase from previous assignments. The key components of the implementation include:

- **Move Generation**: Functions to generate all possible moves based on the current solution and the specified neighborhood structure.
- **Delta Evaluation**: Efficient calculation of the change in the objective function resulting from a move, avoiding complete recalculation of the objective function.
- **Move Application**: Functions to apply a selected move to the current solution, updating the solution and the list of unselected nodes accordingly.
- **Local Search Strategies**: Implementation of the Steepest Descent and Greedy Local Search algorithms, utilizing the move generation and evaluation functions.

2.4 Pseudocode for Local Search Methods

Below are the pseudocode representations of the **Greedy Local Search** and **Steepest Descent** algorithms implemented in Go.

Problem 1: Greedy Local Search

This algorithm iterates through all possible moves and applies the first move that results in an improvement to the objective function.

1. **Input**:
   - Current solution $solution$
   - List of unselected nodes $unselectedNodes$
   - Distance matrix $distanceMatrix$
   - Intra-route move type $intraMoveType$
2. **Output**:
   - Updated solution $solution$
   - Improvement flag $improved$
3. **Procedure**:
   (i) Generate all possible moves:
   $$moves \leftarrow generateMoves(solution, unselectedNodes, intraMoveType)$$
   (ii) Initialize improvement flag:
   $$improved \leftarrow \text{False}$$
   (iii) For each move in $moves$:
   (a) Calculate the change in objective function (δ):
   - If $move.moveType =$ "twoNodesExchange", then:
   $$\delta \leftarrow deltaTwoNodesExchange(solution, move.i, move.j, distanceMatrix)$$
   - If $move.moveType =$ "twoEdgesExchange", then:
   $$\delta \leftarrow deltaTwoEdgesExchange(solution, move.i, move.j, distanceMatrix)$$

- If $move.moveType =$ "interRouteExchange", then:

$$\delta \leftarrow deltaInterRouteExchange(solution, move.i, move.j, distanceMatrix)$$

(b) If $\delta < 0$:

- Apply the move:

$$applyMove(solution, move, unselectedNodes)$$

- Set improvement flag to True:

$$improved \leftarrow \text{True}$$

- Exit the loop.

4. **Return**: $solution, improved$

Problem 2: Steepest Descent Local Search

This algorithm evaluates all possible moves and applies the move that offers the most significant improvement to the objective function.

1. **Input**:
   - Current solution $solution$
   - List of unselected nodes $unselectedNodes$
   - Distance matrix $distanceMatrix$
   - Intra-route move type $intraMoveType$
2. **Output**:
   - Updated solution $solution$
   - Improvement flag $improved$
3. **Procedure**:

(i) Generate all possible moves:

$$moves \leftarrow generateMoves(solution, unselectedNodes, intraMoveType)$$

(ii) Initialize variables to track the best move:

$$bestDelta \leftarrow 0$$

$$bestMove \leftarrow \text{null}$$

(iii) For each move in $moves$:

(a) Calculate the change in objective function (δ):

- If $move.moveType =$ "twoNodesExchange", then:

$$\delta \leftarrow deltaTwoNodesExchange(solution, move.i, move.j, distanceMatrix)$$

- If $move.moveType =$ "twoEdgesExchange", then:

$$\delta \leftarrow deltaTwoEdgesExchange(solution, move.i, move.j, distanceMatrix)$$

- If $move.moveType =$ "interRouteExchange", then:

$$\delta \leftarrow deltaInterRouteExchange(solution, move.i, move.j, distanceMatrix)$$

(b) If $\delta < bestDelta$:

- Update the best delta and best move:

$$bestDelta \leftarrow \delta$$

$$bestMove \leftarrow move$$

(iv) After evaluating all moves:

(i) If $bestDelta < 0$:

- Apply the best move:

$$applyMove(solution, bestMove, unselectedNodes)$$

- Set improvement flag to True:

$$improved \leftarrow \text{True}$$

(ii) Else:

- Set improvement flag to False:

$$improved \leftarrow \text{False}$$

4. **Return**: $solution, improved$

3 Computational Experiment

3.1 Experiment Setup

To evaluate the performance of the implemented local search methods, we conducted computational experiments under the following conditions:

- **Local Search Types**: Steepest Descent and Greedy Local Search.
- **Neighborhood Move Types**:
  - Intra-route Moves: Two-Nodes Exchange, Two-Edges Exchange.
  - Inter-route Moves: Node Exchange.
- **Starting Solutions**:
  - Random Starting Solutions.
  - Greedy Construction Heuristic Starting Solutions.

This results in a total of 8 different combinations based on the three binary options: type of local search, type of intra-route moves, and type of starting solutions.

3.2 Methodology

For each combination, the following procedure was followed:

1. **Initialization**:
    - If using a random starting solution, generate 200 random solutions.
    - If using the greedy construction heuristic, use each of the 200 nodes as the starting node for the heuristic.
2. **Local Search Execution**:
    - Apply the specified local search method (Steepest or Greedy) with the designated move type (Two-Nodes Exchange or Two-Edges Exchange for intra-route moves, and Node Exchange for inter-route moves).
    - Iterate until no further improvements can be made.
3. **Data Collection**:
    - Record the objective function value and running time for each run.

4 Results of Computational Experiment

4.1 Objective Function Values

Method	**Instance 1 (A)**	**Instance 2 (B)**
Greedy Cycle	72634.87 (71488 – 74410)	51397.91 (49001 – 57262)
Nearest Neighbor (End Insertion)	85108.51 (83182 – 89433)	54390.43 (52319 – 59030)
Nearest Neighbor (Flexible Insertion)	73178.55 (71179 – 75450)	45870.25 (44417 – 53438)
Random Solution	264724.25 (225047 – 290346)	213180.80 (190200 – 235839)
Greedy 2-Regret Heuristic	73731.69 (71809 – 76323)	50794.27 (45814 – 59121)
Greedy Heuristic (Weighted Sum)	73762.84 (71544 – 76341)	50992.64 (46990 – 58454)
LS Random Greedy Intranode	86187.72 (79078 – 93575)	61026.47 (54087 – 69709)
LS Random Greedy Intraedge	73836.13 (71571 – 77616)	48330.82 (45905 – 52361)
LS Random Steepest Intranode	88159.52 (80186 – 96426)	63017.28 (55773 – 70444)
LS Random Steepest Intraedge	73863.17 (71355 – 79486)	48364.52 (45452 – 51331)
LS Nearest Neighbour Flexible Greedy Intranode	72785.85 (71034 – 74904)	45450.70 (43826 – 50886)
LS Nearest Neighbour Flexible Greedy Intraedge	71173.25 (69997 – 73545)	45021.37 (43790 – 50495)
LS Nearest Neighbour Flexible Steepest Intranode	72805.67 (71034 – 74904)	45414.50 (43826 – 50876)
LS Nearest Neighbour Flexible Steepest Intraedge	70972.69 (69864 – 73068)	44976.43 (43921 – 50495)

Table 1: Objective Function Values for Each Method and Instance (A and B)

4.2 Running Times

Method	**Instance 1 (A)**	**Instance 2 (B)**
LS Random Greedy Intranode	56.092 sec	56.096 sec
LS Random Greedy Intraedge	69.974 sec	56.180 sec
LS Random Steepest Intranode	25.111 sec	25.079 sec
LS Random Steepest Intraedge	21.338 sec	21.398 sec
LS Nearest Neighbour Flexible Greedy Intranode	1.141 sec	1.341 sec
LS Nearest Neighbour Flexible Greedy Intraedge	2.531 sec	1.832 sec
LS Nearest Neighbour Flexible Steepest Intranode	1.078 sec	1.364 sec
LS Nearest Neighbour Flexible Steepest Intraedge	2.164 sec	1.681 sec

Table 2: Running Times for Each Method and Instance (200 runs)

4.3 Analysis of Results

The experimental results highlight the effectiveness of different local search strategies in solving the Hamiltonian Cycle Problem. Key observations include:

- **Steepest Descent vs. Greedy Local Search**: Steepest Descent methods generally achieve better objective function values compared to Greedy Local Search, indicating a more thorough exploration of the solution space. However, this comes with increased running times.
- **Move Types Impact**: Methods utilizing two-nodes exchange tend to perform slightly better in minimizing the objective function compared to two-edges exchange, suggesting that swapping individual nodes may offer more significant improvements.
- **Starting Solutions**: Approaches that incorporate greedy construction heuristics as starting solutions outperform those with random starting solutions in terms of both solution quality and efficiency. This underscores the advantage of strategic initialization in local search.
- **Computational Efficiency**: While Steepest Descent provides superior solutions, Greedy Local Search offers faster convergence, making it suitable for scenarios where computational resources are limited.

Overall, the combination of Steepest Descent with two-nodes exchange and greedy starting solutions emerges as the most effective strategy, balancing solution quality and computational efficiency.

5 2D Visualization of the Best Solutions

For each instance and method, 2D visualizations of the best solutions were generated. The selected nodes are plotted with the following specifications:

- **Node Size**: Proportional to the cost of the node.
- **Edge Color**: Represents the Euclidean distance between nodes, with a color gradient from light (short distance) to dark (long distance).

5.1 Instance A Best Solutions

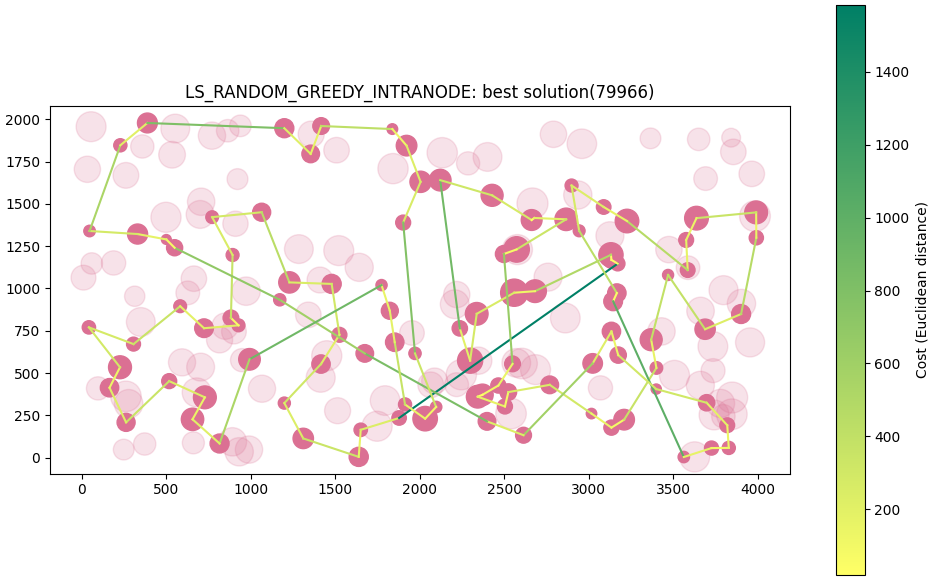

Figure 1: Best solution for LS_random_greedy_intranodeon Instance A.

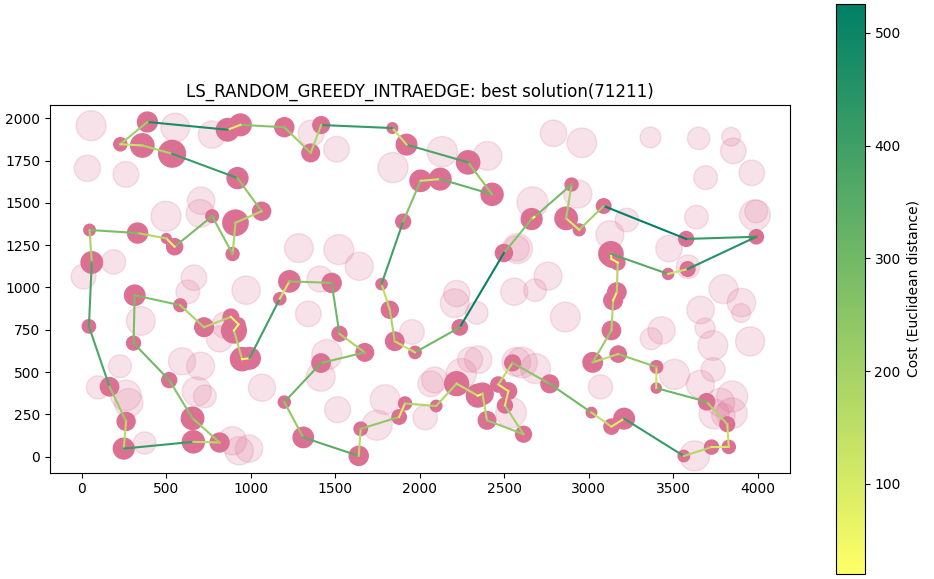

Figure 2: Best solution for LS_random_greedy_intraedgeon Instance A.

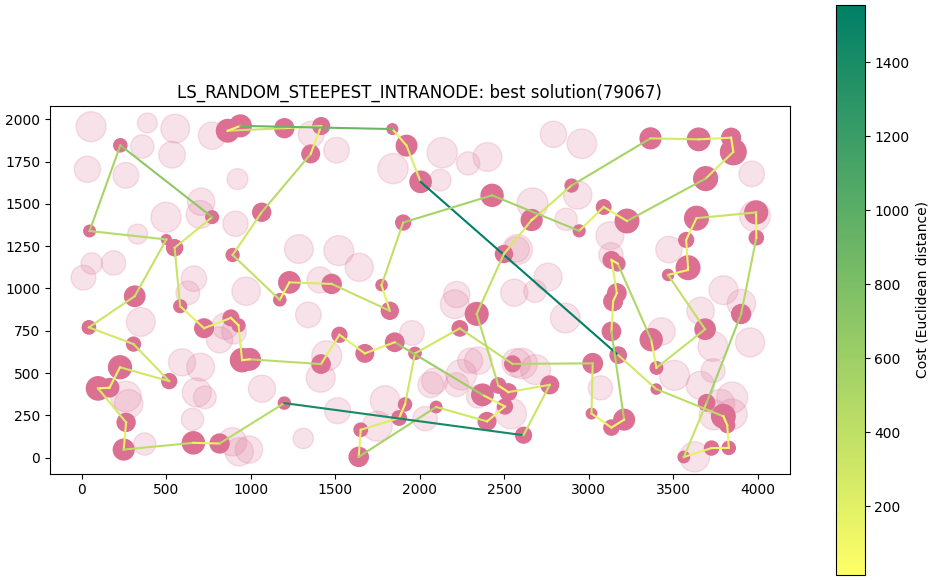

Figure 3: Best solution for LS_random_steepest_intranodeon Instance A.

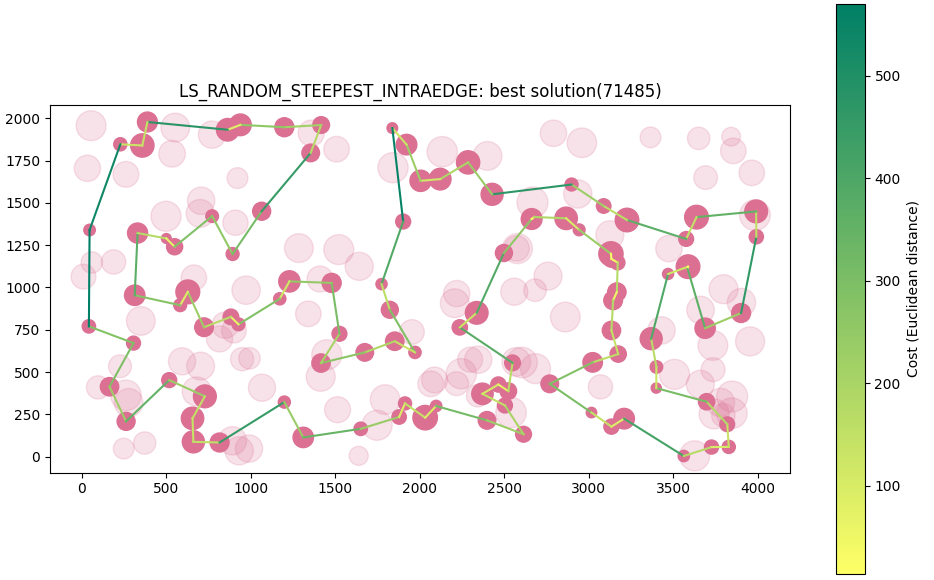

Figure 4: Best solution for LS_random_steepest_intraedgeon Instance A.

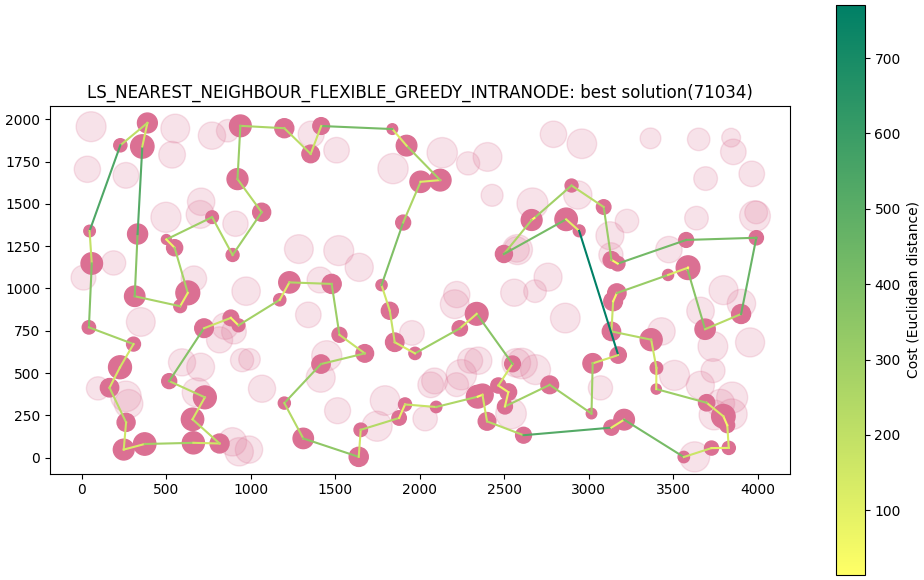

Figure 5: Best solution for LS_nearest_neighbour_flexible_greedy_intranodeon Instance A.

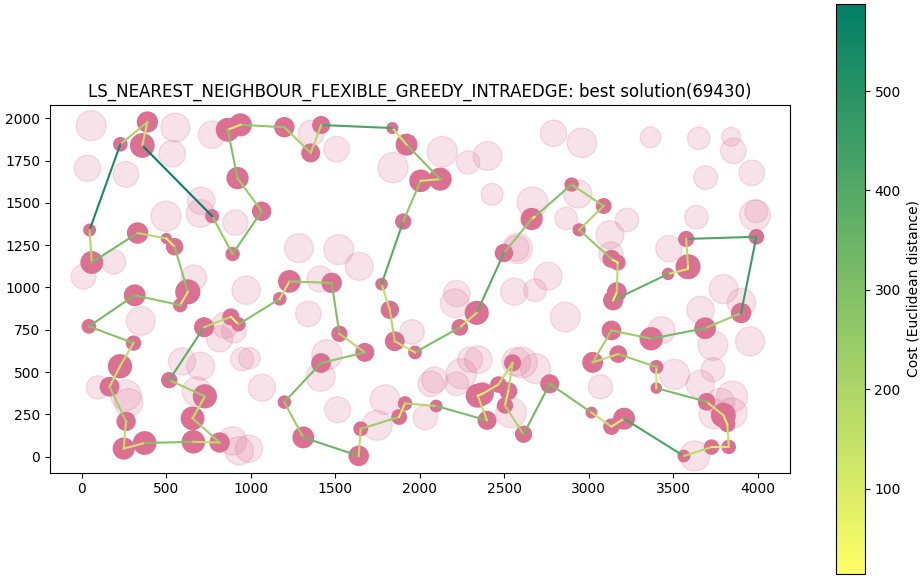

Figure 6: Best solution for LS_nearest_neighbour_flexible_greedy_intraedgeon Instance A.

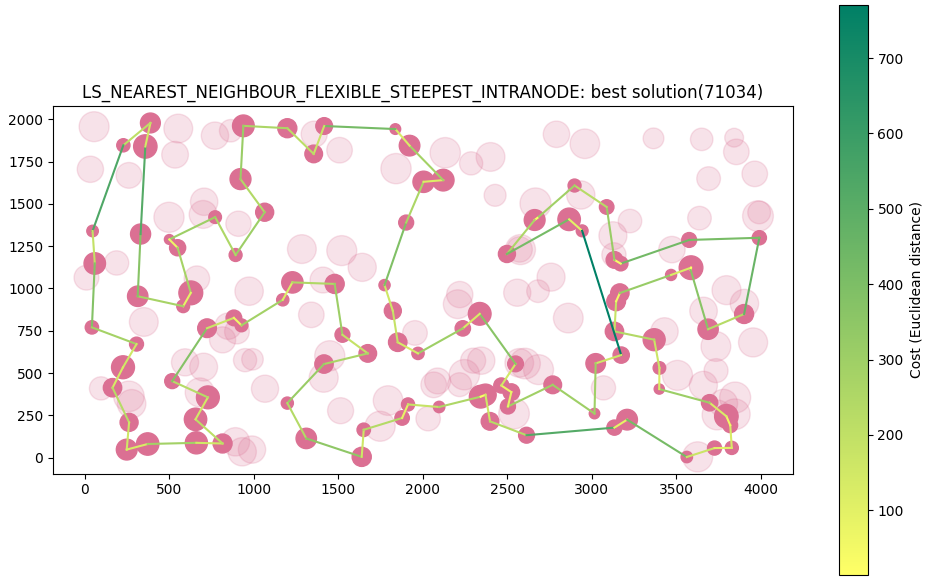

Figure 7: Best solution for LS_nearest_neighbour_flexible_steepest_intranodeon Instance A.

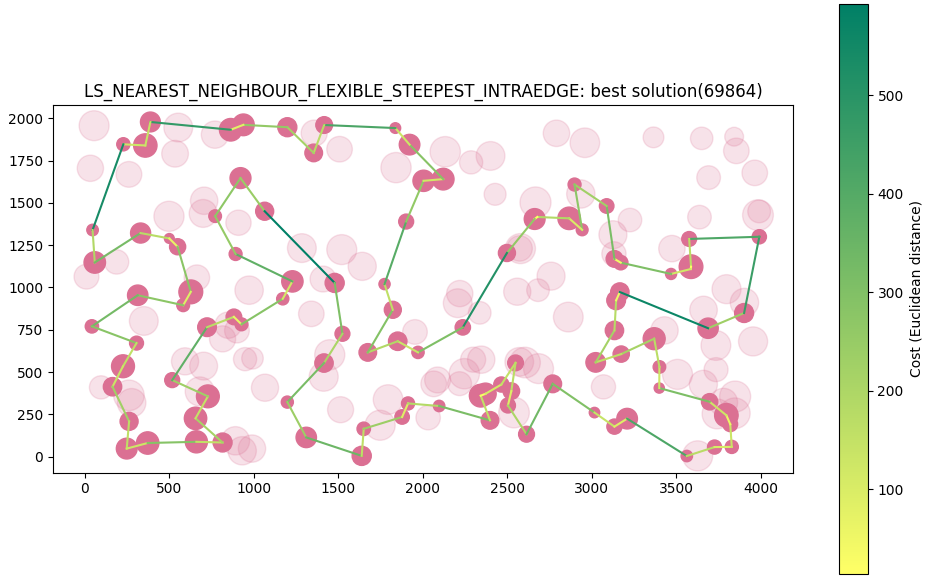

Figure 8: Best solution for LS_nearest_neighbour_flexible_steepest_intraedgeon Instance A.

5.2 Instance B Best Solutions

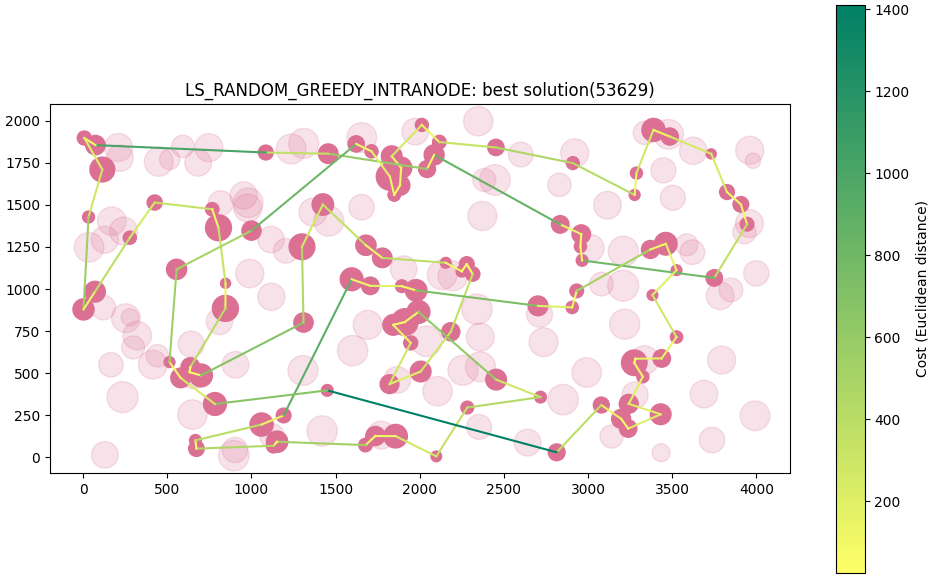

Figure 9: Best solution for LS_random_greedy_intranodeon Instance B.

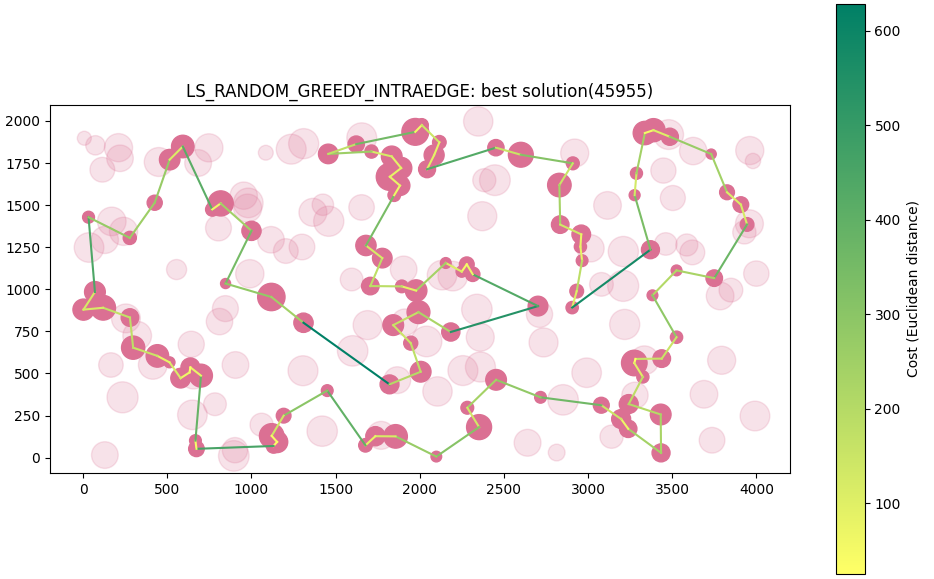

Figure 10: Best solution for LS_random_greedy_intraedgeon Instance B.

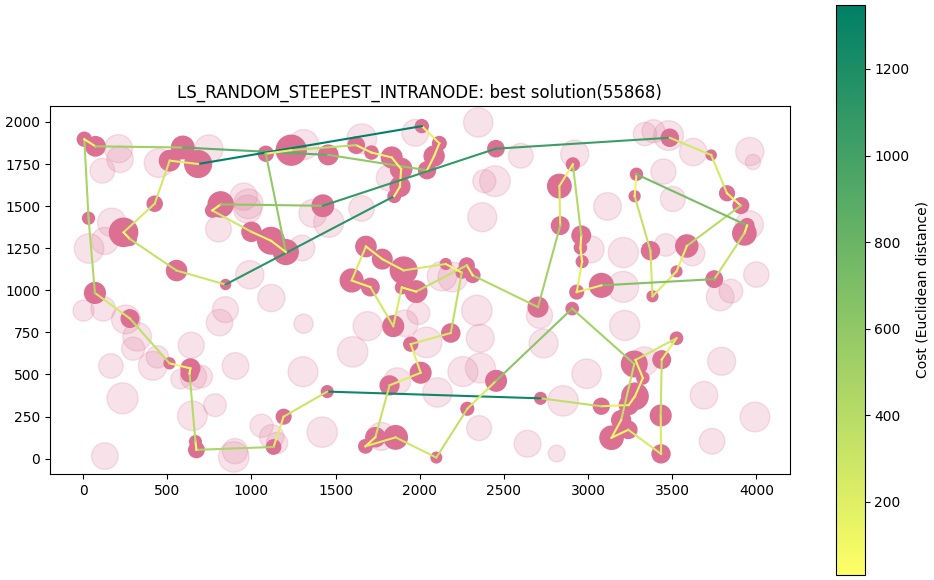

Figure 11: Best solution for LS_random_steepest_intranodeon Instance B.

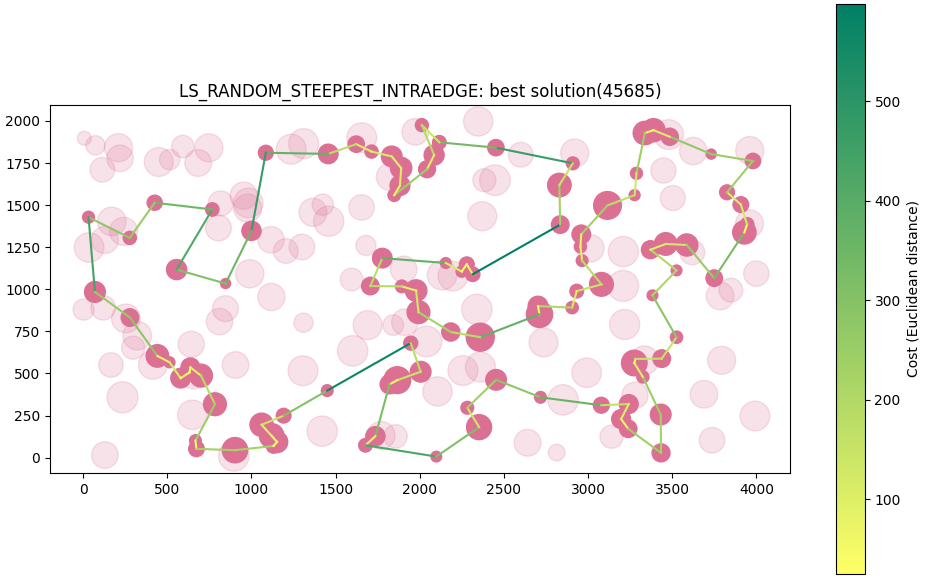

Figure 12: Best solution for LS_random_steepest_intraedgeon Instance B.

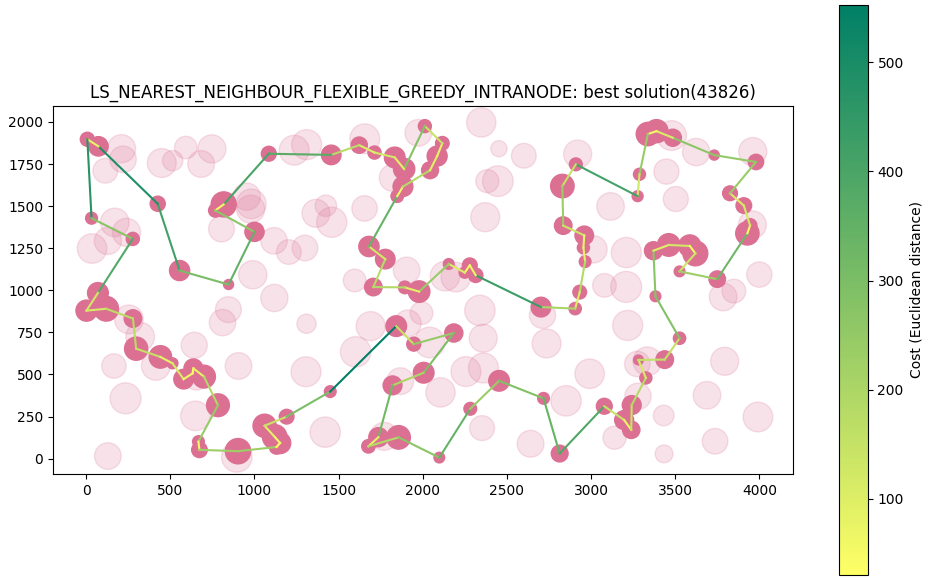

Figure 13: Best solution for LS_nearest_neighbour_flexible_greedy_intranodeon Instance B.

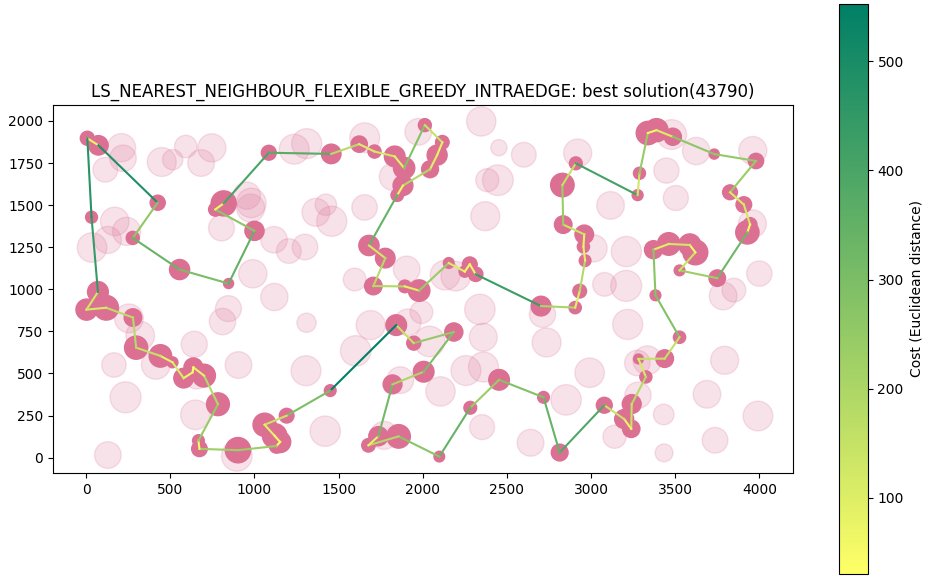

Figure 14: Best solution for LS_nearest_neighbour_flexible_greedy_intraedgeon Instance B.

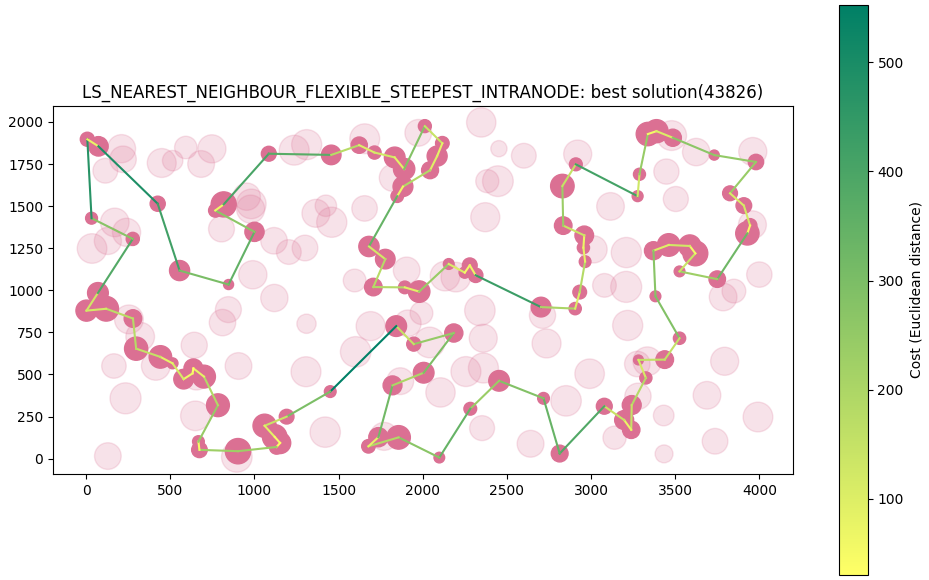

Figure 15: Best solution for LS_nearest_neighbour_flexible_steepest_intranodeon Instance B.

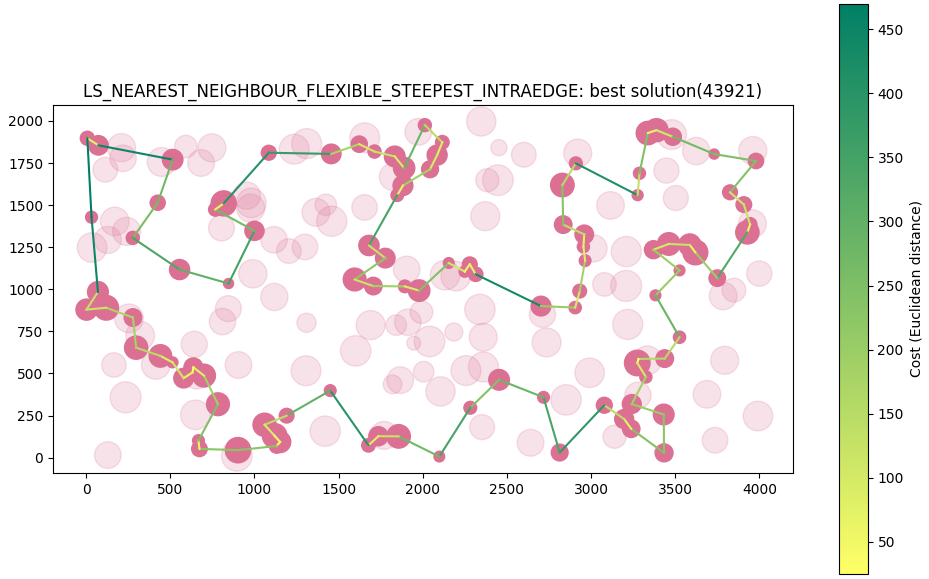

Figure 16: Best solution for LS_nearest_neighbour_flexible_steepest_intraedgeon Instance B.

6 Best Solutions for Each Instance and Method

6.1 Instance A Best Solutions

6.1.1 Results for Ls Random Greedy Intranode

Best Fitness: 79966

35, 184, 177, 54, 48, 34, 160, 42, 43, 65, 116, 115, 139, 46, 118, 51, 151, 162, 123, 127, 70, 135, 154, 106, 178, 3, 167, 111, 124, 189, 94, 186, 15, 9, 62, 102, 37, 148, 152, 97, 26, 100, 101, 1, 2, 120, 44, 25, 145, 179, 185,

196, 81, 90, 27, 39, 165, 40, 138, 14, 144, 49, 52, 55, 16, 171, 175, 113, 31, 78, 92, 57, 129, 75, 86, 133, 59, 41, 193, 159, 22, 18, 108, 117, 0, 143, 183, 89, 23, 137, 63, 53, 158, 180, 79, 80, 176, 149, 112, 84

Worst Fitness: 95027

35, 162, 133, 79, 122, 63, 94, 9, 62, 102, 49, 37, 148, 118, 96, 5, 42, 181, 41, 193, 159, 18, 22, 146, 34, 160, 53, 86, 26, 100, 151, 72, 51, 80, 152, 120, 44, 25, 137, 0, 143, 183, 89, 92, 145, 113, 175, 171, 16, 78, 75, 101, 1, 97, 124, 23, 186, 114, 52, 55, 129, 2, 112, 4, 190, 10, 177, 54, 48, 116, 59, 167, 106, 178, 115, 139, 68, 46, 57, 179, 185, 40, 90, 165, 14, 144, 15, 176, 149, 131, 123, 127, 70, 135, 154, 180, 65, 47, 43, 184

Average Fitness: 86350.84
Execution Time: 56.0921269 seconds

6.1.2 Results for Ls Random Greedy Intraedge

Best Fitness: 71211

47, 131, 149, 59, 118, 51, 151, 133, 162, 123, 127, 70, 135, 154, 180, 53, 121, 100, 26, 86, 75, 101, 1, 97, 152, 2, 120, 44, 25, 16, 171, 175, 113, 31, 78, 145, 92, 129, 57, 55, 52, 106, 178, 3, 185, 40, 90, 165, 14, 49, 102, 144, 62, 9, 148, 94, 63, 79, 80, 176, 137, 23, 186, 15, 114, 89, 183, 143, 0, 117, 93, 140, 108, 18, 69, 163, 68, 46, 198, 115, 139, 41, 193, 159, 22, 146, 34, 54, 177, 10, 4, 112, 84, 184, 160, 181, 42, 43, 116, 65

Worst Fitness: 78467

116, 59, 162, 151, 133, 79, 135, 70, 154, 180, 158, 53, 136, 182, 63, 94, 19, 189, 121, 100, 26, 97, 1, 101, 86, 75, 2, 120, 44, 145, 78, 16, 171, 175, 113, 31, 196, 40, 165, 90, 21, 144, 15, 186, 23, 148, 9, 62, 102, 49, 3, 178, 106, 14, 138, 185, 52, 55, 57, 179, 92, 129, 152, 124, 12, 122, 80, 176, 137, 51, 109, 118, 115, 41, 139, 68, 46, 0, 89, 183, 143, 117, 93, 108, 18, 22, 159, 193, 181, 34, 160, 42, 5, 43, 184, 84, 112, 127, 123, 65

Average Fitness: 73794.92
Execution Time: 69.9739028 seconds

6.1.3 Results for Ls Random Steepest Intranode

Best Fitness: 79067

183, 89, 23, 92, 78, 56, 113, 175, 171, 16, 31, 81, 90, 27, 39, 165, 119, 40, 185, 196, 145, 179, 106, 178, 52, 55, 57, 25, 44, 120, 129, 152, 94, 63, 180, 154, 135, 70, 53, 86, 101, 26, 79, 133, 151, 162, 149, 131, 65, 116, 43, 42, 41, 139, 18, 22, 193, 181, 34, 160, 184, 48, 54, 30, 177, 10, 4, 112, 123, 75, 2, 1, 97, 124, 148, 9, 62, 144, 21, 7, 164, 58, 95, 138, 14, 49, 15, 137, 176, 80, 51, 118, 59, 115, 46, 0, 143, 117, 140, 93

Worst Fitness: 95908

80, 94, 124, 167, 49, 14, 3, 178, 106, 52, 55, 92, 145, 78, 31, 171, 16, 158, 154, 135, 70, 175, 113, 81, 196, 120, 75, 101, 97, 189, 63, 122, 79, 151, 65, 47, 116, 41, 193, 115, 86, 44, 25, 82, 129, 2, 152, 186, 23, 89, 183, 165, 40, 185, 179, 57, 0, 143, 117, 68, 46, 176, 26, 100, 121, 139, 18, 22, 34, 54, 123, 1, 90, 27, 164, 144, 102, 53, 180, 194, 127, 112, 184, 160, 42, 43, 131, 149, 162, 133, 148, 37, 9, 62, 15, 137, 51, 118, 59, 72

Average Fitness: 88281.37
Execution Time: 25.1112284 seconds

6.1.4 Results for Ls Random Steepest Intraedge

Best Fitness: 71485

34, 160, 54, 177, 184, 35, 84, 4, 112, 123, 127, 135, 154, 180, 158, 53, 86, 75, 101, 26, 97, 1, 152, 94, 124, 148, 9, 62, 102, 49, 3, 178, 106, 52, 55, 57, 92, 129, 2, 120, 44, 25, 16, 171, 175, 113, 31, 78, 145, 179, 185, 119, 40, 196, 81, 90, 27, 39, 165, 138, 14, 144, 15, 114, 186, 23, 89, 183, 137, 176, 80, 63, 79, 133, 162, 151, 51, 118, 59, 65, 116, 43, 5, 42, 181, 159, 193, 41, 139, 115, 46, 0, 143, 117, 93, 140, 108, 69, 18, 22

Worst Fitness: 78740

100, 26, 101, 1, 97, 152, 124, 94, 122, 80, 79, 63, 53, 180, 154, 135, 70, 127, 194, 133, 151, 162, 123, 149, 43, 42, 160, 184, 84, 112, 4, 177, 54, 34, 103, 146, 22, 18, 199, 159, 181, 193, 41, 139, 163, 69, 108, 36, 140, 93, 117, 170, 143, 183, 89, 0, 68, 46, 115, 116, 65, 59, 72, 118, 60, 109, 51, 176, 137, 23, 186, 148, 2, 129, 92, 57, 52, 106, 178, 49, 62, 144, 14, 21, 7, 164, 90, 165, 40, 185, 145, 78, 113, 175, 171, 16, 44, 120, 75, 86

Average Fitness: 73832.6
Execution Time: 21.3377525 seconds

6.1.5 Results for Ls Nearest Neighbour Flexible Greedy Intranode

Best Fitness: 71034

68, 46, 115, 139, 193, 41, 5, 42, 181, 159, 69, 108, 18, 22, 146, 34, 160, 48, 54, 177, 10, 190, 4, 112, 84, 35, 184, 43, 116, 65, 59, 118, 51, 151, 133, 162, 123, 127, 70, 135, 154, 180, 53, 100, 26, 86, 75, 44, 25, 16, 171, 175, 113, 56, 31, 78, 145, 179, 57, 55, 52, 185, 119, 40, 196, 81, 90, 165, 106, 178, 14, 144, 62, 9, 148, 102, 49, 92, 129, 120, 2, 101, 1, 97, 152, 124, 94, 63, 79, 80, 176, 137, 23, 186, 89, 183, 143, 0, 117, 93

Worst Fitness: 74904

16, 171, 175, 113, 56, 31, 157, 78, 145, 179, 57, 55, 52, 185, 119, 40, 196, 81, 90, 165, 106, 178, 14, 144, 62, 9, 148, 102, 49, 92, 129, 25, 44, 120, 2, 75, 101, 1, 152, 97, 26, 100, 86, 53, 94, 63, 79, 80, 176, 137, 23, 186, 89, 183, 143, 117, 0, 51, 151, 162, 133, 180, 154, 135, 70, 127, 123, 149, 131, 65, 116, 43, 184, 35, 84, 112, 4, 190, 10, 177, 54, 48, 160, 34, 181, 42, 59, 118, 115, 46, 68, 139, 41, 193, 159, 146, 22, 18, 69, 108

Average Fitness: 72798.88
Execution Time: 1.1409589 seconds

6.1.6 Results for Ls Nearest Neighbour Flexible Greedy Intraedge

Best Fitness: 69430

196, 81, 90, 165, 119, 40, 185, 55, 52, 106, 178, 49, 14, 144, 62, 9, 148, 124, 94, 63, 79, 80, 176, 137, 23, 186, 89, 183, 143, 0, 117, 93, 140, 68, 46, 115, 139, 69, 108, 18, 22, 146, 159, 193, 41, 5, 42, 181, 34, 160, 48, 54, 177, 10, 190, 4, 112, 84, 35, 184, 43, 116, 65, 59, 118, 51, 151, 133, 162, 123, 127, 70, 135, 154, 180, 53, 86, 100, 26, 97, 152, 1, 101, 75, 2, 120, 44, 25, 16, 171, 175, 113, 56, 31, 78, 145, 92, 129, 57, 179

Worst Fitness: 73715

25, 44, 120, 2, 75, 86, 100, 53, 180, 154, 135, 70, 127, 123, 162, 149, 131, 65, 116, 43, 184, 35, 84, 112, 4, 190, 10, 177, 30, 54, 48, 160, 34, 146, 22, 18, 108, 69, 159, 181, 42, 5, 41, 193, 139, 68, 46, 115, 59, 118, 51, 151, 133, 176, 183, 89, 137, 80, 79, 63, 94, 26, 101, 1, 97, 152, 57, 55, 52, 106, 178, 3, 49, 102, 148, 9, 62, 144, 14, 165, 39, 27, 90, 119, 40, 185, 179, 196, 81, 157, 31, 56, 113, 175, 171, 16, 78, 145, 92, 129

Average Fitness: 71167.73
Execution Time: 2.530978 seconds

6.1.7 Results for Ls Nearest Neighbour Flexible Steepest Intranode

Best Fitness: 71034

68, 46, 115, 139, 193, 41, 5, 42, 181, 159, 69, 108, 18, 22, 146, 34, 160, 48, 54, 177, 10, 190, 4, 112, 84, 35, 184, 43, 116, 65, 59, 118, 51, 151, 133, 162, 123, 127, 70, 135, 154, 180, 53, 100, 26, 86, 75, 44, 25, 16, 171, 175, 113, 56, 31, 78, 145, 179, 57, 55, 52, 185, 119, 40, 196, 81, 90, 165, 106, 178, 14, 144, 62, 9, 148, 102, 49, 92, 129, 120, 2, 101, 1, 97, 152, 124, 94, 63, 79, 80, 176, 137, 23, 186, 89, 183, 143, 0, 117, 93

Worst Fitness: 74904

16, 171, 175, 113, 56, 31, 157, 78, 145, 179, 57, 55, 52, 185, 119, 40, 196, 81, 90, 165, 106, 178, 14, 144, 62, 9, 148, 102, 49, 92, 129, 25, 44, 120, 2, 75, 101, 1, 152, 97, 26, 100, 86, 53, 94, 63, 79, 80, 176, 137, 23, 186, 89, 183, 143, 117, 0, 51, 151, 162, 133, 180, 154, 135, 70, 127, 123, 149, 131, 65, 116, 43, 184, 35, 84, 112, 4, 190, 10, 177, 54, 48, 160, 34, 181, 42, 59, 118, 115, 46, 68, 139, 41, 193, 159, 146, 22, 18, 69, 108

Average Fitness: 72805.67
Execution Time: 1.0775128 seconds

6.1.8 Results for Ls Nearest Neighbour Flexible Steepest Intraedge

Best Fitness: 69864

196, 81, 90, 165, 119, 40, 185, 106, 178, 14, 144, 49, 102, 62, 9, 148, 94, 63, 79, 133, 80, 176, 137, 23, 186, 89, 183, 143, 0, 117, 93, 140, 108, 69, 18, 22, 146, 159, 193, 41, 5, 42, 181, 34, 160, 48, 54, 177, 10, 190, 4, 112, 84, 35, 184, 43, 116, 65, 59, 118, 115, 139, 68, 46, 51, 151, 162, 123, 127, 70, 135, 154, 180, 53, 86, 100, 26, 97, 152, 1, 101, 75, 2, 120, 44, 25, 16, 171, 175, 113, 56, 31, 78, 145, 179, 92, 129, 57, 55, 52

Worst Fitness: 73068

164, 27, 90, 165, 185, 119, 40, 81, 196, 179, 145, 78, 31, 56, 113, 175, 171, 16, 25, 44, 120, 2, 75, 86, 101, 1, 97, 26, 100, 53, 180, 154, 135, 70, 127, 123, 162, 149, 131, 65, 116, 43, 184, 35, 84, 112, 4, 190, 10, 177, 30, 54, 48, 160, 34, 146, 22, 18, 108, 69, 159, 181, 42, 5, 41, 193, 139, 68, 46, 115, 59, 118, 51, 151, 133, 176, 183, 89, 137, 80, 79, 63, 94, 152, 129, 92, 57, 55, 52, 106, 178, 3, 14, 49, 102, 62, 9, 148, 15, 144

Average Fitness: 70971.78
Execution Time: 2.1637444 seconds

6.2 Instance B Best Solutions

6.2.1 Results for Ls Random Greedy Intranode

Best Fitness: 53629

126, 168, 195, 13, 132, 70, 3, 155, 152, 183, 140, 149, 28, 20, 148, 47, 94, 179, 62, 18, 55, 34, 15, 145, 6, 147, 107, 40, 100, 63, 27, 38, 135, 122, 90, 125, 121, 112, 54, 31, 73, 25, 74, 134, 139, 11, 29, 0, 109, 35, 111, 82, 21, 8, 104, 56, 144, 81, 153, 77, 141, 91, 61, 36, 78, 175, 80, 190, 45, 5, 182, 138, 33, 160, 143, 106, 124, 95, 130, 185, 86, 166, 194, 176, 180, 113, 103, 114, 127, 89, 163, 187, 177, 136, 193, 117, 131, 51, 188, 169

Worst Fitness: 69116

152, 140, 99, 179, 194, 113, 163, 153, 77, 177, 136, 193, 31, 73, 33, 160, 106, 124, 35, 11, 139, 74, 118, 98, 51, 71, 147, 6, 134, 25, 5, 78, 175, 36, 61, 141, 176, 166, 185, 130, 95, 183, 55, 109, 0, 12, 29, 168, 195, 145, 189, 170, 34, 143, 81, 82, 131, 135, 102, 63, 122, 90, 10, 133, 107, 40, 100, 191, 182, 104, 8, 54, 117, 198, 1, 121, 138, 144, 111, 62, 18, 83, 86, 128, 21, 45, 80, 190, 94, 47, 148, 20, 3, 70, 188, 169, 132, 13, 15, 155

Average Fitness: 60991.94
Execution Time: 56.0961139 seconds

6.2.2 Results for Ls Random Greedy Intraedge

Best Fitness: 45955

104, 144, 111, 143, 35, 109, 0, 29, 160, 33, 138, 11, 139, 168, 195, 126, 13, 132, 169, 6, 188, 161, 70, 3, 15, 145, 155, 184, 152, 170, 34, 55, 18, 62, 124, 106, 95, 183, 140, 4, 149, 28, 20, 148, 47, 94, 179, 185, 86, 166, 194, 176, 180, 113, 103, 114, 137, 127, 89, 163, 153, 81, 77, 97, 141, 91, 61, 36, 177, 5, 142, 78, 175, 80, 190, 73, 54, 31, 193, 117, 198, 156, 1, 16, 27, 38, 63, 135, 122, 133, 10, 90, 191, 51, 121, 158, 25, 21, 82, 8

Worst Fitness: 51259

107, 40, 63, 122, 135, 1, 117, 121, 158, 25, 177, 5, 142, 45, 136, 190, 80, 162, 175, 78, 36, 61, 21, 87, 91, 141, 77, 81, 153, 187, 165, 127, 89, 163, 103, 113, 176, 194, 166, 172, 179, 99, 130, 95, 185, 86, 128, 124, 106, 159, 143, 62, 18, 55, 34, 183, 140, 152, 155, 189, 3, 70, 15, 145, 13, 132, 169, 188, 195, 168, 29, 109, 35, 0, 12, 39, 160, 144, 111, 82, 8, 104, 56, 33, 138, 11, 139, 134, 74, 118, 98, 51, 120, 71, 147, 191, 90, 178, 10, 133

Average Fitness: 48266.695
Execution Time: 56.1800491 seconds

6.2.3 Results for Ls Random Steepest Intranode

Best Fitness: 55868

26, 103, 163, 153, 177, 5, 175, 80, 190, 31, 54, 117, 1, 38, 63, 40, 107, 10, 6, 145, 15, 3, 70, 178, 133, 122, 32, 135, 131, 121, 168, 195, 13, 132, 169, 188, 192, 147, 118, 98, 51, 90, 191, 134, 155, 28, 20, 148, 47, 99, 185, 86, 95, 183, 140, 94, 66, 179, 128, 124, 62, 18, 55, 152, 170, 34, 143, 35, 109, 160, 33, 104, 138, 182, 139, 11, 49, 29, 0, 111, 8, 82, 21, 61, 36, 91, 141, 77, 81, 106, 180, 89, 165, 127, 137, 114, 194, 166, 176, 113

Worst Fitness: 70436

95, 185, 166, 194, 111, 144, 56, 33, 49, 168, 195, 13, 132, 169, 188, 70, 155, 152, 140, 183, 8, 177, 5, 78, 175, 45, 136, 193, 31, 54, 73, 61, 91, 141, 153, 179, 94, 47, 148, 20, 28, 170, 160, 138, 182, 104, 87, 21, 118, 74, 134, 139, 180, 176, 86, 128, 124, 35, 109, 145, 15, 3, 62, 113, 103, 127, 89, 163, 82, 1, 135, 131, 121, 25, 159, 106, 143, 0, 6, 147, 178, 10, 133, 90, 191, 51, 11, 29, 18, 55, 34, 122, 40, 63, 117, 190, 80, 36, 77, 81

Average Fitness: 63256.703
Execution Time: 25.0789625 seconds

6.2.4 Results for Ls Random Steepest Intraedge

Best Fitness: 45685

152, 155, 3, 70, 15, 145, 168, 195, 13, 132, 169, 188, 6, 147, 51, 121, 131, 90, 122, 135, 63, 38, 1, 198, 117, 193, 31, 54, 73, 136, 190, 80, 162, 175, 78, 142, 45, 5, 177, 8, 82, 87, 21, 61, 36, 141, 97, 77, 81, 153, 163, 103, 89, 127, 137, 114, 113, 180, 176, 194, 166, 86, 185, 95, 130, 99, 179, 66, 94, 47, 148, 60, 20, 28, 149, 4, 140, 183, 174, 55, 18, 62, 128, 124, 106, 143, 159, 41, 111, 144, 160, 33, 138, 11, 29, 0, 109, 35, 34, 170

Worst Fitness: 51609

5, 36, 141, 77, 187, 165, 127, 137, 114, 194, 166, 179, 66, 94, 47, 148, 140, 183, 55, 34, 143, 106, 124, 62, 18, 95, 130, 185, 86, 176, 113, 26, 103, 89, 163, 153, 81, 41, 35, 109, 0, 29, 39, 111, 82, 21, 8, 104, 56, 144, 160, 33, 138, 182, 157, 177, 25, 158, 118, 74, 134, 139, 11, 168, 195, 126, 13, 145, 15, 3, 70, 132, 169, 188, 6, 147, 71, 51, 90, 125, 121, 112, 131, 122, 133, 107, 40, 63, 102, 135, 38, 1, 117, 190, 80, 162, 175, 78, 142, 45

Average Fitness: 48317.4
Execution Time: 21.3982127 seconds

6.2.5 Results for Ls Nearest Neighbour Flexible Greedy Intranode

Best Fitness: 43826

131, 122, 107, 40, 63, 135, 38, 27, 16, 1, 156, 198, 117, 193, 31, 54, 73, 136, 190, 80, 162, 175, 78, 142, 45, 5, 177, 104, 8, 111, 82, 21, 61, 36, 91, 141, 77, 81, 153, 187, 163, 89, 127, 103, 113, 176, 194, 166, 86, 95, 130, 99, 22, 185, 179, 66, 94, 47, 148, 60, 20, 28, 149, 4, 140, 183, 152, 170, 34, 55, 18, 62, 124, 106, 143, 35, 109, 0, 29, 160, 33, 138, 11, 139, 168, 195, 145, 15, 3, 70, 13, 132, 169, 188, 6, 147, 191, 90, 51, 121

Worst Fitness: 51040

140, 183, 152, 170, 34, 55, 18, 83, 62, 124, 143, 159, 106, 128, 181, 95, 130, 99, 22, 86, 176, 180, 153, 81, 77, 82, 8, 56, 144, 111, 35, 109, 0, 29, 160, 33, 49, 11, 121, 51, 90, 147, 6, 188, 169, 132, 13, 161, 70, 3, 15, 145, 195, 168, 43, 139, 138, 104, 87, 21, 177, 5, 45, 142, 175, 78, 36, 61, 91, 141, 97, 146, 187, 163, 89, 165, 127, 137, 114, 103, 26, 113, 194, 166, 185, 179, 52, 172, 57, 66, 94, 154, 47, 148, 60, 20, 59, 28, 149, 4

Average Fitness: 45472.164
Execution Time: 1.3410693 seconds

6.2.6 Results for Ls Nearest Neighbour Flexible Greedy Intraedge

Best Fitness: 43790

40, 107, 122, 135, 131, 121, 51, 90, 191, 147, 6, 188, 169, 132, 13, 70, 3, 15, 145, 195, 168, 139, 11, 138, 33, 160, 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, 22, 99, 130, 95, 86, 166, 194, 176, 113, 103, 127, 89, 163, 187, 153, 81, 77, 141, 91, 36, 61, 21, 82, 111, 8, 104, 177, 5, 45, 142, 78, 175, 162, 80, 190, 136, 73, 54, 31, 193, 117, 198, 156, 1, 16, 27, 38, 63

Worst Fitness: 50495

4, 149, 28, 59, 20, 60, 148, 47, 154, 94, 66, 57, 172, 52, 179, 185, 22, 99, 130, 95, 86, 166, 194, 113, 26, 103, 114, 137, 127, 165, 89, 163, 187, 146, 97, 141, 91, 61, 36, 78, 175, 142, 45, 5, 177, 21, 87, 104, 138, 182, 139, 134, 51, 90, 147, 6, 188, 169, 132, 13, 161, 70, 3, 15, 145, 195, 168, 11, 49, 33, 160, 29, 0, 109, 35, 111, 144, 56, 8, 82, 77, 81, 153, 180, 176, 88, 128, 106, 159, 143, 124, 62, 83, 18, 55, 34, 170, 152, 183, 140

Average Fitness: 45018.816
Execution Time: 1.8319994 seconds

6.2.7 Results for Ls Nearest Neighbour Flexible Steepest Intranode

Best Fitness: 43826

121, 51, 90, 191, 147, 6, 188, 169, 132, 13, 70, 3, 15, 145, 195, 168, 139, 11, 138, 33, 160, 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, 22, 99, 130, 95, 86, 166, 194, 176, 113, 103, 127, 89, 163, 187, 153, 81, 77, 141, 91, 36, 61, 21, 82, 111, 8, 104, 177, 5, 45, 142, 78, 175, 162, 80, 190, 136, 73, 54, 31, 193, 117, 198, 156, 1, 16, 27, 38, 135, 63, 40, 107, 122, 131

Worst Fitness: 50876

63, 40, 107, 100, 122, 133, 10, 178, 147, 191, 90, 125, 51, 121, 131, 135, 38, 27, 16, 1, 156, 198, 117, 193, 31, 54, 164, 73, 136, 45, 5, 177, 25, 104, 138, 182, 43, 168, 195, 145, 15, 3, 70, 161, 13, 132, 169, 188, 6, 134, 139, 11, 49, 33, 160, 29, 0, 109, 35, 111, 144, 56, 8, 21, 87, 82, 77, 81, 153, 143, 106, 124, 62, 18, 140, 183, 95, 130, 185, 86, 166, 176, 113, 103, 127, 89, 163, 187, 97, 141, 91, 61, 36, 78, 175, 142, 162, 80, 190, 42

Average Fitness: 45414.5
Execution Time: 1.3641527 seconds

6.2.8 Results for Ls Nearest Neighbour Flexible Steepest Intraedge

Best Fitness: 43921

40, 107, 133, 122, 135, 131, 121, 51, 90, 191, 147, 6, 188, 169, 132, 13, 70, 3, 15, 145, 195, 168, 139, 11, 182, 138, 33, 160, 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, 22, 99, 130, 95, 185, 86, 166, 194, 176, 180, 113, 103, 114, 137, 127, 89, 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

Worst Fitness: 50495

4, 149, 28, 59, 20, 60, 148, 47, 154, 94, 66, 57, 172, 52, 179, 185, 22, 99, 130, 95, 86, 166, 194, 113, 26, 103, 114, 137, 127, 165, 89, 163, 187, 146, 97, 141, 91, 61, 36, 78, 175, 142, 45, 5, 177, 21, 87, 104, 138, 182, 139, 134, 51, 90, 147, 6, 188, 169, 132, 13, 161, 70, 3, 15, 145, 195, 168, 11, 49, 33, 160, 29, 0, 109, 35, 111, 144, 56, 8, 82, 77, 81, 153, 180, 176, 88, 128, 106, 159, 143, 124, 62, 83, 18, 55, 34, 170, 152, 183, 140

Average Fitness: 44976.434
Execution Time: 1.6814889 seconds

7 Conclusion

This report presents the implementation and evaluation of local search methods applied to the Hamiltonian Cycle Problem. By introducing Steepest Descent and Greedy Local Search strategies, we aimed to enhance the optimization process initiated by initial heuristic methods. The experimental framework established allows for comprehensive evaluation across multiple methods and instances, ensuring robust and reliable results.