Assignment 8 - Global Convexity Tests for Selective TSP

Authors

- Mateusz Idziejczak 155842
- Mateusz Stawicki 155900

Github

> https://github.com/Luncenok/EvolutionaryComputing

Problem Description

This is the same variant of the Traveling Salesman Problem as in previous assignments:

- Select exactly 50% of nodes (rounded up if odd)
- Form a Hamiltonian cycle through selected nodes
- Minimize: total path length + sum of selected node costs
- Distances are Euclidean distances rounded to integers

Instances:

- **TSPA, TSPB** with 200 nodes, selecting 100 nodes.

Goal

Analyze the **global convexity** (fitness-similarity relationships) of the solution space by:

1. Generating **1000 random local optima** for each instance
2. Calculating **similarity measures** between solutions (common edges, common nodes)
3. Analyzing how **similarity varies with solution quality** (min, max, avg similarity values)
4. Creating **12 visualization charts** (2 instances × 3 similarity versions × 2 similarity measures)

Similarity Measures

Two types of similarity measures are used to compare solutions:

1. **Number of Common Edges**: Count of edges that appear in both solutions
2. **Number of Common Selected Nodes**: Count of nodes selected in both solutions

Similarity Versions

For each similarity measure, three versions of analysis are performed:

1. **Average Similarity to All**: For each solution, calculate its average similarity to all other 999 local optima
2. **Similarity to Best of 1000**: Calculate similarity to the best local optimum found among the 1000 generated

3. **Similarity to Best Method (ILS)**: Calculate similarity to the best solution found by ILS (the best method from previous assignments)

Note: When calculating similarity to a single good solution, that solution itself is excluded to avoid an outlier with 100% similarity to itself.

Algorithm Pseudocode

Global Convexity Analysis

```
analyzeGlobalConvexity(instanceName, n, selectCount, distance, costs,
bestILSSolution):
    NUM_LOCAL_OPTIMA = 1000
    localOptima = []
    objectives = []

    # Generate 1000 random local optima
    for i in range(NUM_LOCAL_OPTIMA):
        initial = generateRandomSolution(n, selectCount)
        localOpt = localSearchGreedyEdges(initial, distance, costs)
        localOptima.append(localOpt)
        objectives.append(calculateObjective(localOpt, distance, costs))

    # Find best local optimum
    bestIdx = argmin(objectives)
    bestLocalOpt = localOptima[bestIdx]

    # Calculate statistics
    minObj = min(objectives)
    maxObj = max(objectives)
    avgObj = mean(objectives)

    # For each similarity measure (edges, nodes)
    for measureType in [EDGES, NODES]:

        # Version 1: Average similarity to all other local optima
        similarities_avg = []
        for i in range(NUM_LOCAL_OPTIMA):
            avgSim = 0
            for j in range(NUM_LOCAL_OPTIMA):
                if i != j:
                    avgSim += calculateSimilarity(localOptima[i], localOptima[j],
measureType)
            avgSim /= (NUM_LOCAL_OPTIMA - 1)
            similarities_avg.append(avgSim)

        corr_avg = pearsonCorrelation(objectives, similarities_avg)
        exportToCSV(objectives, similarities_avg, corr_avg)

        # Version 2: Similarity to best of 1000 local optima
        similarities_best1000 = []
        for i in range(NUM_LOCAL_OPTIMA):
```

```
            if i == bestIdx:
                continue  # Skip the best solution itself
            sim = calculateSimilarity(localOptima[i], bestLocalOpt, measureType)
            similarities_best1000.append(sim)

        corr_best1000 = pearsonCorrelation(objectives[excluding bestIdx],
similarities_best1000)
        exportToCSV(objectives[excluding bestIdx], similarities_best1000,
corr_best1000)

        # Version 3: Similarity to best ILS solution
        similarities_ILS = []
        for i in range(NUM_LOCAL_OPTIMA):
            sim = calculateSimilarity(localOptima[i], bestILSSolution,
measureType)
            similarities_ILS.append(sim)

        corr_ILS = pearsonCorrelation(objectives, similarities_ILS)
        exportToCSV(objectives, similarities_ILS, corr_ILS)

    return results
```

Similarity Calculation Functions

```
calculateCommonEdges(solution1, solution2):
    # Build edge set for solution1
    edges1 = set()
    for i in range(len(solution1)):
        u = solution1[i]
        v = solution1[(i + 1) % len(solution1)]
        edges1.add((min(u, v), max(u, v)))  # Normalize edge representation

    # Count common edges with solution2
    commonCount = 0
    for i in range(len(solution2)):
        u = solution2[i]
        v = solution2[(i + 1) % len(solution2)]
        edge = (min(u, v), max(u, v))
        if edge in edges1:
            commonCount += 1

    return commonCount

calculateCommonNodes(solution1, solution2):
    # Convert to sets and count intersection
    nodes1 = set(solution1)
    nodes2 = set(solution2)
    return len(nodes1 & nodes2)
```

Pearson Correlation Coefficient

```
pearsonCorrelation(x, y):
    # Calculate means
    meanX = mean(x)
    meanY = mean(y)

    # Calculate covariance and standard deviations
    cov = sum((x[i] - meanX) * (y[i] - meanY) for i in range(len(x)))
    stdX = sqrt(sum((x[i] - meanX)**2 for i in range(len(x))))
    stdY = sqrt(sum((y[i] - meanY)**2 for i in range(len(y))))

    # Return correlation coefficient
    if stdX == 0 or stdY == 0:
        return 0
    return cov / (stdX * stdY)
```

Correlation interpretation:

- **r close to -1**: Strong negative correlation - better solutions (lower objective) tend to have higher similarity to the reference
- **r close to 0**: No correlation - similarity is independent of solution quality
- **r close to +1**: Strong positive correlation - worse solutions tend to have higher similarity to the reference

For global convexity, we expect **negative correlations**, indicating that good solutions cluster together in the search space (funnel-shaped landscape).

Experimental Setup

- **Instances**: TSPA, TSPB (200 nodes, 100 selected)
- **Objective**: Minimize path length + sum of selected node costs
- **Local search**: Greedy local search with edge exchange (from Assignment 3)
- **Number of local optima**: 1000 per instance
- **Initial solutions**: Random solutions with different starting nodes
- **Reference solutions**:
  - Best of 1000 local optima
  - Best ILS solution from Assignment 6
- **Output**: 12 CSV files (2 instances × 3 versions × 2 measures) containing objective values, similarity counts, and correlation coefficients
- **Visualizations**: 12 scatter plots with regression lines showing fitness-similarity relationships

Key Results

Objective Function Values - All Methods Comparison

Method	TSPA	TSPB
Random	264501 (235453 – 288189)	212513 (189071 – 238254)
Nearest Neighbor (end)	85108 (83182 – 89433)	54390 (52319 – 59030)

Method	TSPA	TSPB
Nearest Neighbor (any)	73178 (71179 – 75450)	45870 (44417 – 53438)
Greedy Cycle	72646 (71488 – 74410)	51400 (49001 – 57324)
Greedy 2-Regret	115474 (105852 – 123428)	72454 (66505 – 77072)
Greedy Weighted	72129 (71108 – 73395)	50950 (47144 – 55700)
NN Any 2-Regret	116659 (106373 – 126570)	73646 (67121 – 79013)
NN Any Weighted	72401 (70010 – 75452)	47653 (44891 – 55247)
LS Random+Steepest+Nodes	88323 (80903 – 97156)	63219 (56207 – 70573)
LS Random+Greedy+Nodes	92779 (86293 – 102205)	65643 (58888 – 73163)
LS Random+Greedy+Edges	81269 (75576 – 86423)	54272 (50610 – 59193)
LS Greedy+Steepest+Nodes	71614 (70626 – 72950)	45414 (43826 – 50876)
LS Greedy+Steepest+Edges	71460 (70510 – 72614)	44979 (43921 – 50629)
LS Greedy+Greedy+Nodes	71913 (71093 – 73048)	45561 (43917 – 51144)
LS Greedy+Greedy+Edges	71817 (70977 – 72844)	45371 (43845 – 51072)
LS Random+Steepest+Edges	73945 (70937 – 78033)	48313 (45799 – 51543)
LM Random+Steepest+Edges	74973 (71993 – 80945)	49391 (46324 – 53526)
Candidates k=5	84660 (78119 – 91398)	49996 (46328 – 53421)
Candidates k=10	77494 (73550 – 83200)	48461 (45358 – 53439)
Candidates k=15	75268 (71917 – 80679)	48201 (45251 – 51868)
Candidates k=20	74451 (71417 – 79637)	48294 (45356 – 51272)
LM Candidates k=10	74829 (72274 – 79625)	49201 (46111 – 53213)
LM Candidates k=20	74962 (71993 – 80945)	49391 (46324 – 53526)
MSLS (200 iterations)	71344 (70813 – 71786)	45758 (45040 – 46209)
ILS	**69359 (69107 – 69776)**	**43785 (43465 – 44209)**
LNS with LS	69751 (69255 – 70147)	44255 (43747 – 44651)
LNS without LS	69851 (69291 – 70389)	44294 (43671 – 45669)
Global Convexity (1000 LO)	**81055 (75733 – 88526)**	**54207 (49379 – 61043)**

Running Time Comparison (ms)

Method	TSPA	TSPB
Random	0.0002 (0.0001 – 0.0021)	0.0001 (0.0000 – 0.0006)

Method	TSPA	TSPB
Nearest Neighbor (end)	0.0387 (0.0344 – 0.0603)	0.0398 (0.0354 – 0.0542)
Nearest Neighbor (any)	1.4470 (1.4212 – 1.5059)	1.4215 (1.3943 – 1.5409)
Greedy Cycle	2.5977 (2.5191 – 2.6936)	2.5575 (2.4907 – 2.6884)
Greedy 2-Regret	2.8215 (2.5644 – 5.3887)	2.5969 (2.5337 – 2.7132)
Greedy Weighted	2.6242 (2.5992 – 2.6960)	2.5782 (2.5349 – 2.6755)
NN Any 2-Regret	1.5038 (1.4172 – 1.6111)	1.4758 (1.4031 – 1.5906)
NN Any Weighted	1.5728 (1.4408 – 1.9339)	1.5160 (1.4143 – 1.6346)
LS Random+Steepest+Nodes	24.2276 (19.0661 – 31.4049)	24.3617 (19.1283 – 31.8331)
LS Random+Greedy+Nodes	3.4547 (1.8853 – 6.7367)	3.1663 (1.7297 – 5.2907)
LS Random+Greedy+Edges	2.5430 (1.5906 – 4.0691)	2.5049 (1.7306 – 4.3680)
LS Greedy+Steepest+Nodes	3.5489 (2.8667 – 4.5868)	2.3749 (1.8601 – 5.1730)
LS Greedy+Steepest+Edges	3.5445 (3.0082 – 4.3915)	2.3720 (1.8860 – 5.0376)
LS Greedy+Greedy+Nodes	3.7871 (3.3134 – 5.0790)	2.6515 (2.1049 – 4.3128)
LS Greedy+Greedy+Edges	3.8511 (3.3632 – 5.2897)	2.5698 (2.1430 – 3.7055)
LS Random+Steepest+Edges	16.3947 (14.4390 – 21.8291)	16.4927 (13.9633 – 40.8522)
LM Random+Steepest+Edges	5.5767 (4.1653 – 8.5713)	5.2454 (4.3264 – 6.3337)
Candidates k=5	4.5188 (3.7939 – 5.9315)	4.5828 (4.0974 – 5.1938)
Candidates k=10	6.1366 (5.3462 – 7.4807)	6.5578 (5.6982 – 7.4067)
Candidates k=15	8.0357 (7.2033 – 9.0747)	8.6749 (7.6928 – 9.6709)
Candidates k=20	9.8870 (8.8077 – 11.1121)	10.5835 (9.3987 – 12.0207)
LM Candidates k=10	7.5203 (6.2955 – 8.8053)	7.2025 (5.9445 – 8.3050)
LM Candidates k=20	21.6993 (19.0448 – 31.7943)	22.4066 (19.5947 – 26.2397)
MSLS (200 iterations)	3259.67 (3206.98 – 3425.49)	3263.80 (3182.64 – 3492.31)
ILS	3260.06 (3259.68 – 3260.58)	3264.38 (3263.81 – 3265.93)
LNS with LS	3260.24 (3259.69 – 3260.96)	3264.46 (3264.02 – 3265.15)
LNS without LS	3260.16 (3259.68 – 3260.74)	3264.30 (3263.82 – 3265.21)
Global Convexity (1000 LO)	**29391**	**30079**

Global Convexity Analysis Results

TSPA - Similarity Analysis (1000 Random Local Optima)

Local Optima Statistics:

- **Objective**: Min=75733, Max=88526, Avg=81055
- **Time**: 29391 ms (~29.4 seconds)

Raw Similarity Counts:

Measure	Similarity Version	Min	Max	Avg
Common Edges (out of 100)	Avg to All	17.3	29.1	22.7
Common Edges (out of 100)	Best of 1000	16	44	29.1
Common Edges (out of 100)	Best ILS	22	52	34.2
Common Nodes (out of 100)	Avg to All	83.5	90.6	88.0
Common Nodes (out of 100)	Best of 1000	85	96	90.0
Common Nodes (out of 100)	Best ILS	82	95	90.0

TSPB - Similarity Analysis (1000 Random Local Optima)

Local Optima Statistics:

- **Objective**: Min=49379, Max=61043, Avg=54207
- **Time**: 30079 ms (~30.1 seconds)

Raw Similarity Counts:

Measure	Similarity Version	Min	Max	Avg
Common Edges (out of 100)	Avg to All	16.7	27.1	22.3
Common Edges (out of 100)	Best of 1000	9	39	25.8
Common Edges (out of 100)	Best ILS	18	49	32.5
Common Nodes (out of 100)	Avg to All	77.7	86.0	82.7
Common Nodes (out of 100)	Best of 1000	75	93	85.1
Common Nodes (out of 100)	Best ILS	74	93	85.1

Summary - Raw Similarity Values

Instance	Measure	Avg to All	Best of 1000	Best ILS
TSPA	Common Edges	22.7	29.1	34.2
TSPA	Common Nodes	88.0	90.0	90.0
TSPB	Common Edges	22.3	25.8	32.5
TSPB	Common Nodes	82.7	85.1	85.1

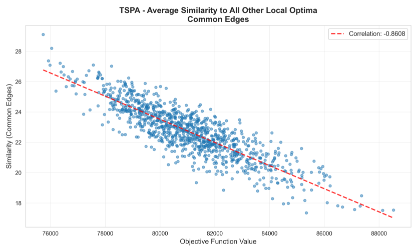
Bonus: Correlation Coefficients (r)

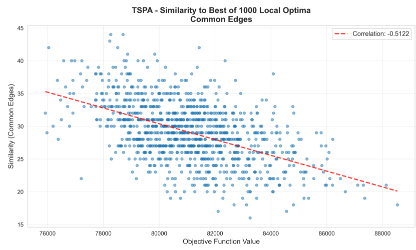
Instance	Measure	Avg to All	Best of 1000	Best ILS
TSPA	Common Edges	-0.861	-0.512	-0.643
TSPA	Common Nodes	-0.256	-0.196	-0.304
TSPB	Common Edges	-0.810	-0.407	-0.569
TSPB	Common Nodes	-0.345	-0.263	-0.375

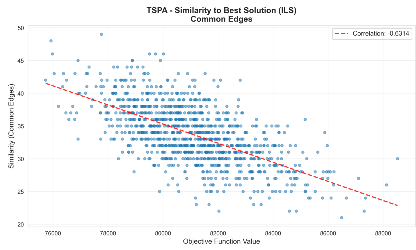
Visualizations

TSPA - Common Edges

Average Similarity to All	Similarity to Best of 1000	Similarity to Best ILS

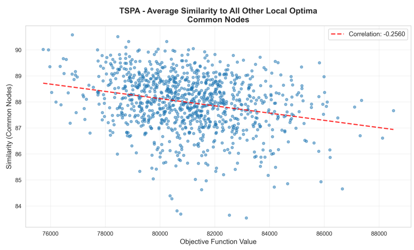


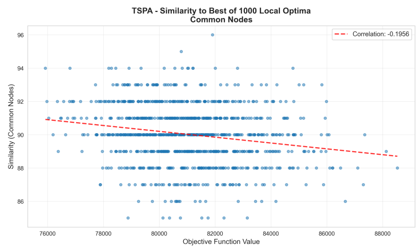


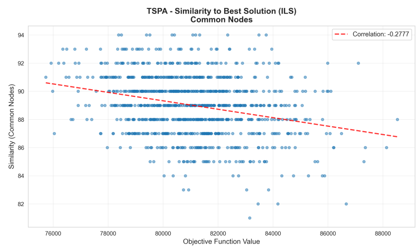


TSPA - Common Nodes

Average Similarity to All	Similarity to Best of 1000	Similarity to Best ILS

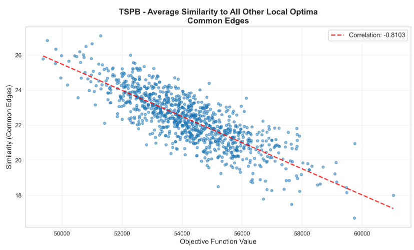


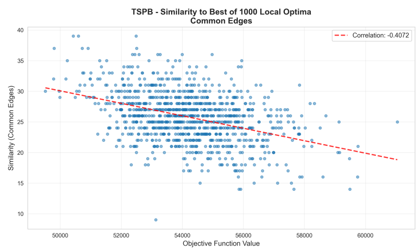


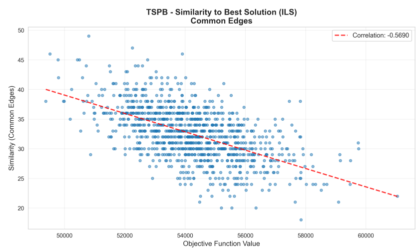


TSPB - Common Edges

Average Similarity to All	Similarity to Best of 1000	Similarity to Best ILS

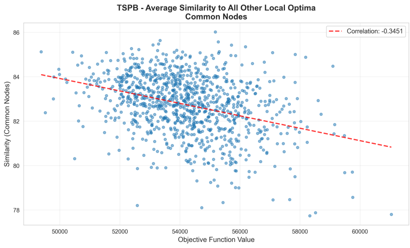


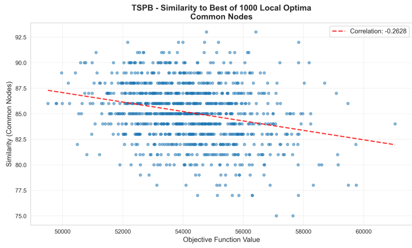


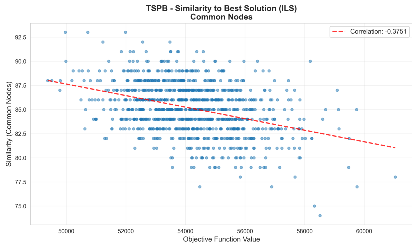


TSPB - Common Nodes

Average Similarity to All	Similarity to Best of 1000	Similarity to Best ILS







Conclusions

Similarity Measures Comparison

Measure	TSPA Avg	TSPB Avg	Interpretation
Common Edges	22.7 / 100	22.3 / 100	Low overlap (~23%) - edges vary significantly
Common Nodes	88.0 / 100	82.7 / 100	High overlap (~85%) - nodes are constrained

Edge diversity is high because many valid tour arrangements exist for similar node sets. **Node similarity** is high because cost structure constrains which nodes are "good" to select.

Global Convexity Evidence

Metric	TSPA	TSPB
Avg similarity to all (edges)	22.7	22.3
Similarity to best ILS (edges)	34.2	32.5
Increase toward best	+52%	+46%
Correlation (edges, avg to all)	-0.861	-0.810

Strong negative correlations (-0.81 to -0.86 for edges) confirm global convexity: better solutions are more similar to each other, indicating a funnel-shaped landscape where good solutions cluster together.

Key Findings

1. **Low edge similarity** (~23%) among random local optima indicates high structural diversity
2. **High node similarity** (~85%) shows node selection is constrained by cost structure
3. **Better solutions share more structure**: 50% more common edges with ILS than average
4. **9-14% quality gap** between best local optima (75733/49379) and ILS (69107/43465) demonstrates the value of sophisticated metaheuristics over simple multi-start approaches
5. **Edge-based optimization** is the primary challenge - node selection is relatively fixed