Report : Max-Cut Problem Analysis

Khalid Hasan Tuhin - 2105002

May 7, 2025

1 High-Level Algorithm Descriptions

The following algorithms were implemented to solve the Max-Cut problem:

- **Randomized**: Assigns vertices to partitions X or Y randomly with equal probability, averaging results over multiple iterations.

- **Greedy**: Starts with the edge of maximum weight, greedily assigning remaining vertices to maximize the current cut.

- **Semi-Greedy**: Uses a restricted candidate list based on a greedy function and random selection, balancing greediness and randomness.

- **Local Search**: Improves an initial solution by iteratively moving vertices to the partition that maximizes the cut until no further improvement is possible.

- **GRASP**: Combines semi-greedy construction with local search, iterating multiple times to find the best solution.

2 Comparison of Algorithms

Based on the results, the performance of algorithms varies across graphs:

- For graph G1:

    - Randomized: 9585
    - Greedy: 10956
    - Semi-Greedy: 11203
    - Local Search: 11366 (after 5 iterations)
    - GRASP: 11432 (after 5 iterations)
    - Known Best: 12078

- For graph G2:

    - Randomized: 9596
    - Greedy: 11030
    - Semi-Greedy: 11116
    - Local Search: 11389 (after 5 iterations)
    - GRASP: 11408 (after 5 iterations)
    - Known Best: 12084

- For graph G3:

- Randomized: 9586
- Greedy: 11012
- Semi-Greedy: 11172
- Local Search: 11350 (after 5 iterations)
- GRASP: 11409 (after 5 iterations)
- Known Best: 12077

- For graph G4:
  - Randomized: 9583
  - Greedy: 11031
  - Semi-Greedy: 11230
  - Local Search: 11382 (after 5 iterations)
  - GRASP: 11441 (after 5 iterations)
  - Known Best: N/A

- For graph G5:
  - Randomized: 9588
  - Greedy: 10990
  - Semi-Greedy: 11235
  - Local Search: 11380 (after 5 iterations)
  - GRASP: 11446 (after 5 iterations)
  - Known Best: N/A

- For graph G6:
  - Randomized: 75
  - Greedy: 1515
  - Semi-Greedy: 1624
  - Local Search: 1915 (after 5 iterations)
  - GRASP: 1960 (after 5 iterations)
  - Known Best: N/A

- For graph G7:
  - Randomized: -82
  - Greedy: 1362
  - Semi-Greedy: 1524
  - Local Search: 1779 (after 5 iterations)
  - GRASP: 1829 (after 5 iterations)
  - Known Best: N/A

- For graph G8:
  - Randomized: -84
  - Greedy: 1331
  - Semi-Greedy: 1478
  - Local Search: 1776 (after 5 iterations)

- – GRASP: 1830 (after 5 iterations)
- – Known Best: N/A

- For graph G9:

  - – Randomized: -27
  - – Greedy: 1507
  - – Semi-Greedy: 1557
  - – Local Search: 1779 (after 5 iterations)
  - – GRASP: 1804 (after 5 iterations)
  - – Known Best: N/A

- For graph G10:

  - – Randomized: -78
  - – Greedy: 1350
  - – Semi-Greedy: 1531
  - – Local Search: 1745 (after 5 iterations)
  - – GRASP: 1783 (after 5 iterations)
  - – Known Best: N/A

GRASP tends to perform closest to the known best solutions across graphs, indicating its effectiveness. Local Search values are notably lower, suggesting potential issues with local optima or initial partitions. Randomized consistently underperforms due to its lack of optimization.
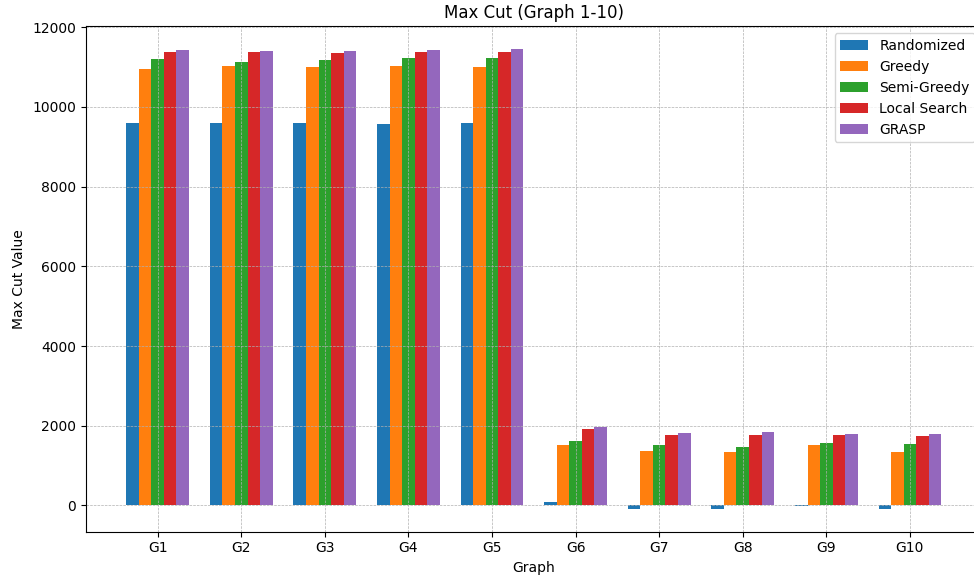
3 Visualization

Figure 1: Max Cut Values for Graphs G1-G10