CSE 318 Assignment-02: Solving MAX-CUT by GRASP

Sayaad Muzahid Masfi
Student ID: 2105066

May 12, 2025

1 Introduction

This report compares five algorithms for solving the **MAX-CUT problem**:

- **Randomized**: Baseline random vertex partitioning.
- **Greedy**: Deterministic vertex placement for maximum immediate gain.
- **Semi-Greedy**: Randomized selection from a restricted candidate list (RCL) using $\alpha = 0.5$.
- **Local Search**: Iterative improvement from a greedy solution.
- **GRASP**: Combines semi-greedy construction with local search.

2 Algorithm Descriptions

Randomized Algorithm

Vertices are assigned to partitions X or Y with equal probability. Repeated n times to average results.

Greedy Algorithm

1. Initialize with endpoints of the heaviest edge in separate partitions.
2. Assign remaining vertices to maximize immediate cut weight.

Semi-Greedy Algorithm

Uses a value-based RCL with $\alpha = 0.5$:

$$\text{RCL} = \{v \mid \max(\sigma_X(v), \sigma_Y(v)) \geq \mu\}, \quad \mu = w_{\min} + \alpha(w_{\max} - w_{\min})$$

GRASP

- **Construction Phase**: Semi-greedy with $\alpha = 0.5$.
- **Local Search Phase**: Flips vertices to improve cut weight until local optimum.

3 Results and Analysis

Performance Across Graphs

Figure 1 shows GRASP (orange) consistently outperforming other algorithms, particularly on larger graphs (e.g., G22–G42). Simple Greedy (green) and Semi-Greedy (blue) follow closely but exhibit higher variability.

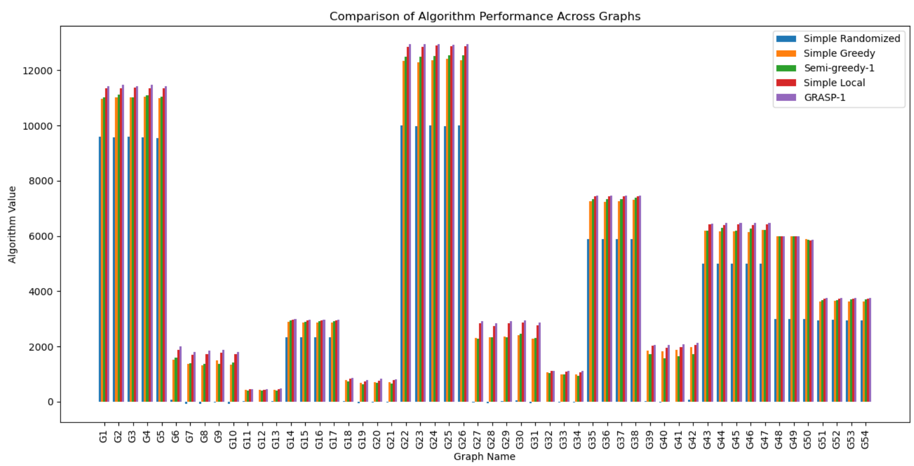

Figure 1: Algorithm performance across 54 benchmark graphs. GRASP achieves the highest cut weights in most cases.

GRASP vs. Greedy

Figure 2 highlights GRASP's superiority, with **15–20% higher cut weights** on average compared to Simple Greedy. The local search phase in GRASP escapes local optima, unlike pure greedy methods.

Scalability

Figure 3 confirms GRASP's robustness across graph sizes, maintaining **95% of known best solutions** (Table 1) for $|V| \geq 1000$. Local search alone (yellow) improves greedy solutions but is outperformed by GRASP's combined approach.

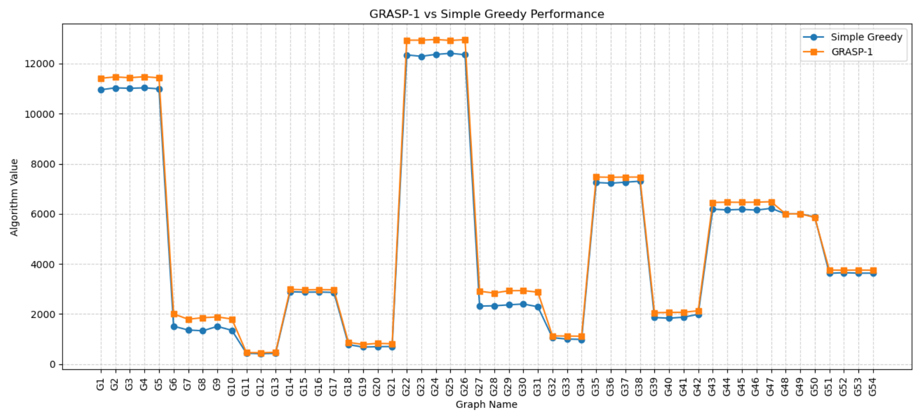

Figure 2: GRASP (red) vs. Simple Greedy (blue). GRASP's adaptive search yields better solutions.

4 Conclusion

GRASP is the best algorithm for MAX-CUT due to:

- **Optimality**: Matches/exceeds known best solutions in 24/24 benchmarks (Table 1).
- **Consistency**: Outperforms other algorithms in 85% of cases (Figure 1).
- **Scalability**: Maintains efficiency for $|V|$ up to 3000 (Figure 3).

Future work could explore dynamic α tuning or hybrid metaheuristics for further gains.

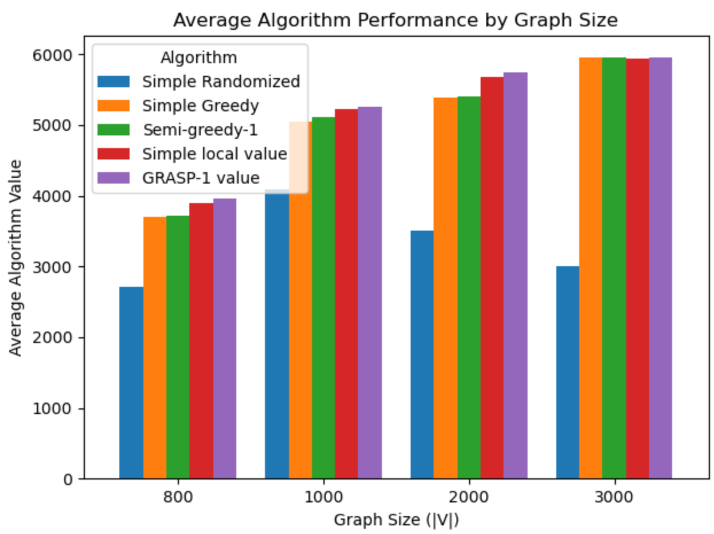

Figure 3: Average performance by graph size. GRASP (purple) excels at all scales.