Report on Assignment2 - MAX CUT

2105067 - AHTASHAMUL HAQUE

May 2025

1 Introduction

1.1 The max-cut problem

Given an undirected graph, $G = (V, U)$, where V is the set of vertices and U is the set of edges, and weights w_{uv} associated with each edge $(u, v) \epsilon U$, the maximum cut (MAX-CUT) problem consists in finding a nonempty proper subset of vertices $S \subset V (S \neq \phi)$, such that the weight of the cut (S, S'), given by $w(S, S') = \sum_{u\epsilon S, v\epsilon S'} w_{uv}$, is maximized.

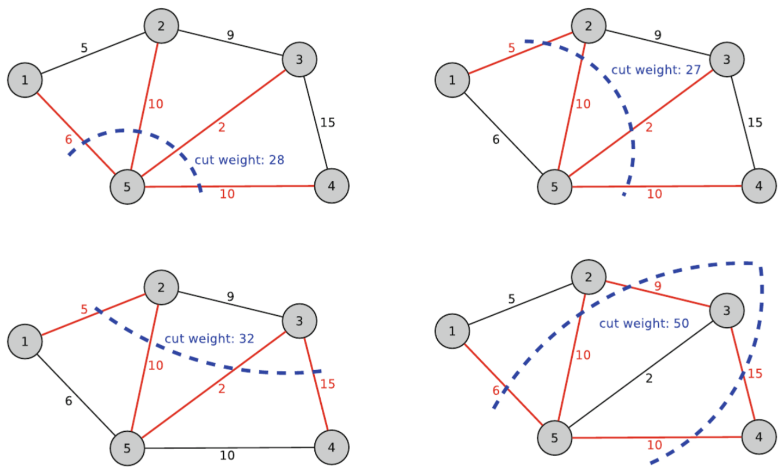


Figure 1: Example of the maximum cut problem on a graph with five vertices and seven edges. Four cuts are shown. The maximum cut is $(S, S') = (\{1, 2, 4\}, \{3, 5\})$ and has a weight $w(S, S') = 50$.

2 Analysis

The implementation details and analysis of the randomized, greedy, and semi-greedy construction phases, along with the key metrics such as average iterations, average local optima, and GRASP cost, would be discussed in this section. This investigation sheds light on their convergence characteristics, adaptability, and efficiency.

2.1 Simple Randomized or Randomized-1

The randomized construction phase involves creating initial solutions using a randomized approach.

Initially, both partitions X and Y are empty. For each vertex $v \in V$, place v in partition X or partition Y with uniform randomness, i.e., with probability $\frac{1}{2}$. The procedure terminates when all vertices are placed either in X or Y.

This means that the solutions are generated by making random decisions without adhering to any strict heuristic rules. The higher average number of iterations indicates that the randomized algorithm might require more iterations to converge or find locally optimal solutions. As the construction phase is random, It has performed worst in almost all the scenerios of given 54 graphs.

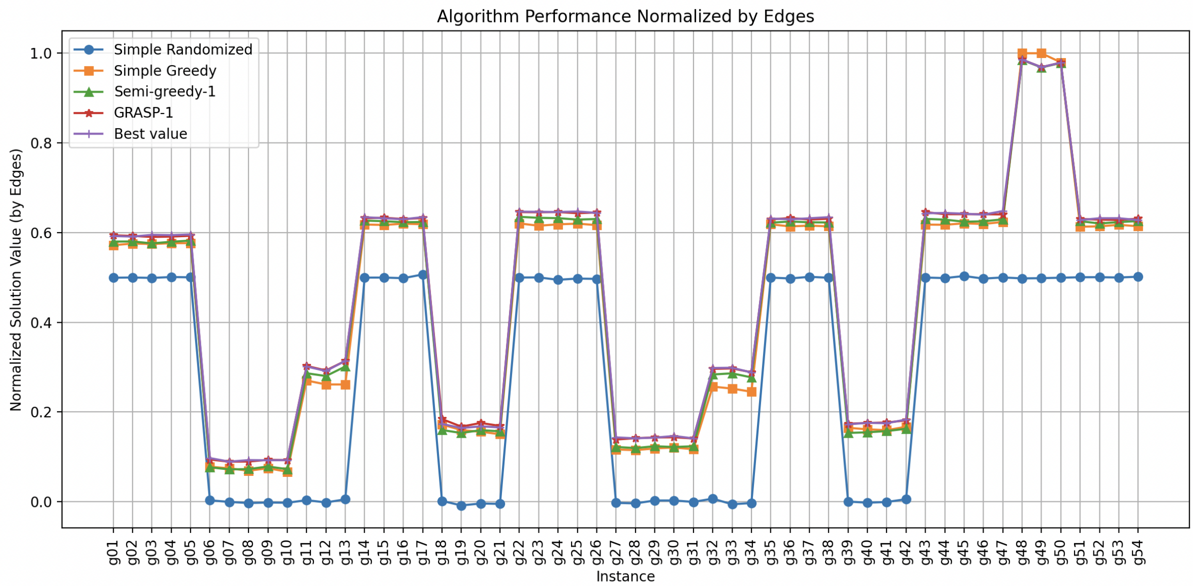


Figure 2: Performance of all the algorithms with normalized EDGES

Here blue line shows Simple Randomized Algorithm performs worst among all the approaches.

2.2 Simple Greedy or Greedy-1

The greedy construction phase relies on a deterministic heuristic that makes locally optimal decisions at each step. Hence, it was run for only one iteration (local search followed by construction). The lower average number of iterations suggests that the greedy algorithm converges faster, indicating that the heuristic choices are effective at reaching solutions quickly.

But there is an issue in this approach. The greedy approach follows a specific path in the solution space, always moving towards the best solution, and may get stuck in local optima. Hence, the average best value is lower (in most cases) compared to the GRASP best value, which uses a semi-greedy construction phase.

2.3 Semi-greedy-1

The semi-greedy construction phase combines aspects of both randomization and greedy heuristics. This could mean that the algorithm balances exploration and exploitation to some extent (here, $\alpha = 0.6$). The average number of iterations and local optima falls between the randomized and greedy approaches, showcasing a trade-off between exploration and convergence speed. The average best value might also fall in between the other two methods since semi-greedy construction aims to strike a balance between the two extremes, potentially resulting in moderate-quality initial solutions.

Among all the three constructive algorithms Semi greedy has performed continuously better in all 54 graphs

2.4 Local Search

Local search is a refinement technique used to improve an initial solution by iteratively exploring its neighborhood—typically defined by small changes such as moving or flipping a vertex from one partition to another. Starting from a solution generated by a heuristic like greedy or semi-greedy, the algorithm evaluates nearby configurations and moves to a better one if it increases the objective function (e.g., the Max-Cut value). This process continues until no further improvement is possible, indicating that a local optimum has been reached. While it doesn't guarantee a global optimum, local search is efficient and often significantly enhances the quality of heuristic solutions.

GRASP (Greedy Randomized Adaptive Search Procedure)

GRASP is an iterative multi-start metaheuristic that combines greedy randomized construction with local search to find high-quality solutions for combinatorial optimization problems of Max-Cut. In each iteration, a solution is built using a semi-greedy heuristic where vertices are added to sets X or Y based on a restricted candidate list (RCL), determined by the cut-off value μ from the given equation. A vertex v is selected randomly from the RCL and assigned to the set that contributes more to the cut weight, based on the greedy function value $\max\{\sigma_X(v), \sigma_Y(v)\}$. After a full assignment is constructed, a local search is applied to improve the solution by moving vertices between X and Y if it increases the cut value. This process is repeated for a fixed number of iterations, and the best solution found is returned.

With increasing number of iterations, the maxcut value from GRASP is tend to increase which is shown in the following plot for some given graphs

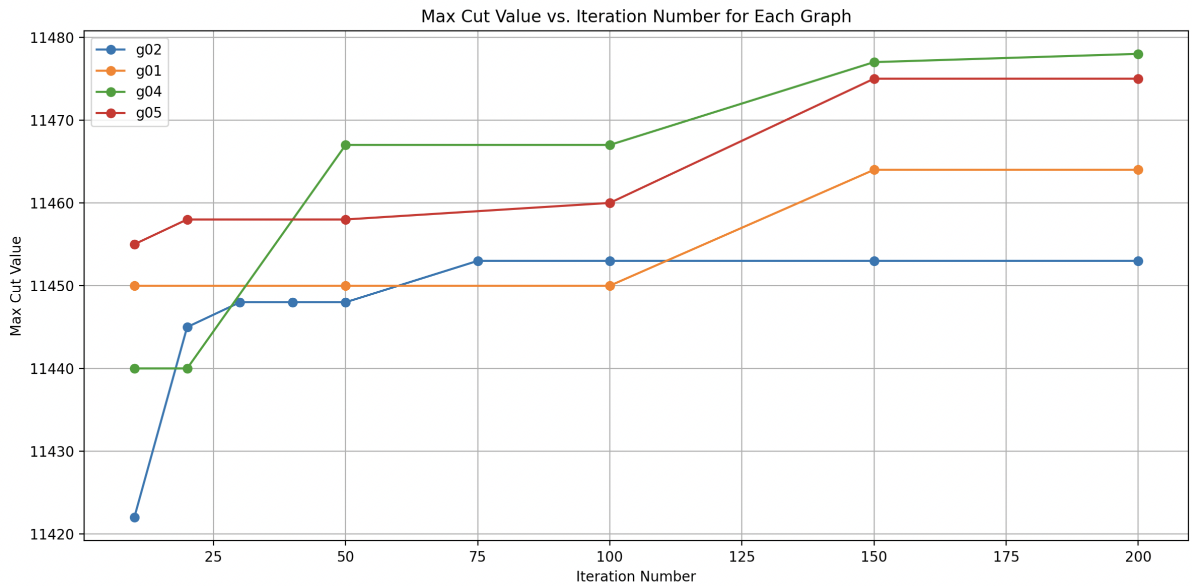


Figure 3: GRASP for increasing number of iterations

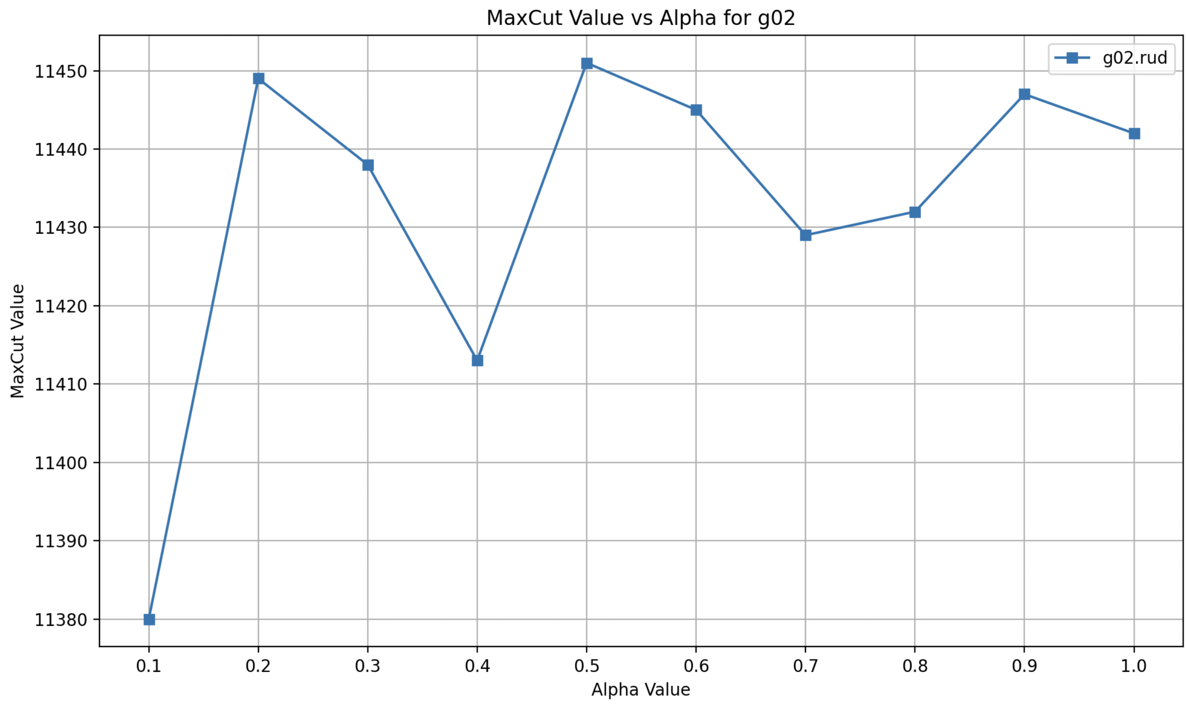


Figure 4: GRASP for increasing number of α

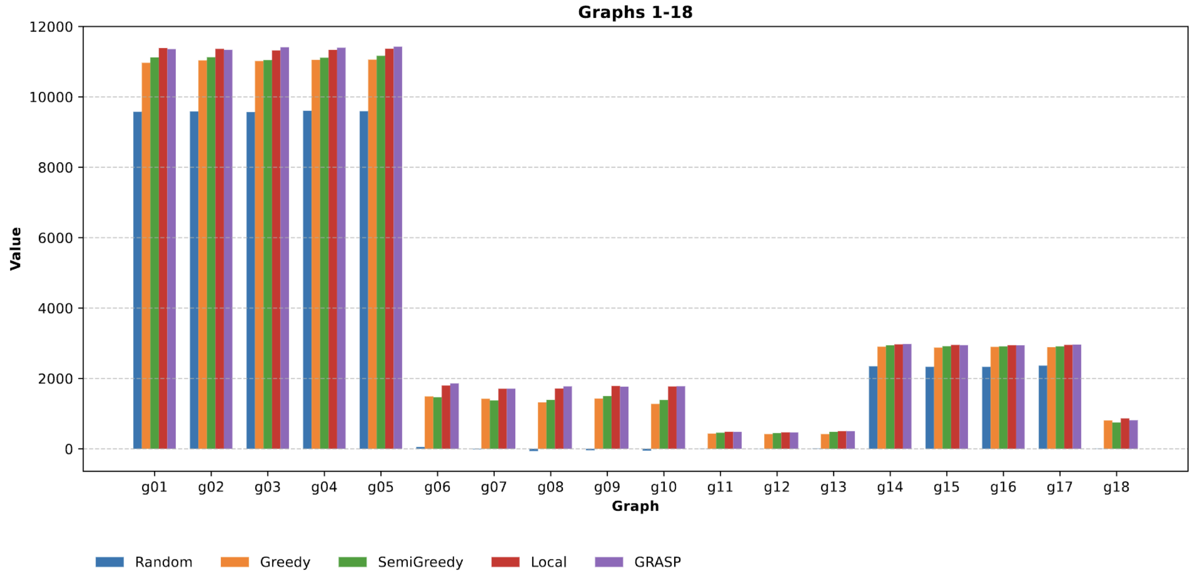
3 Summary

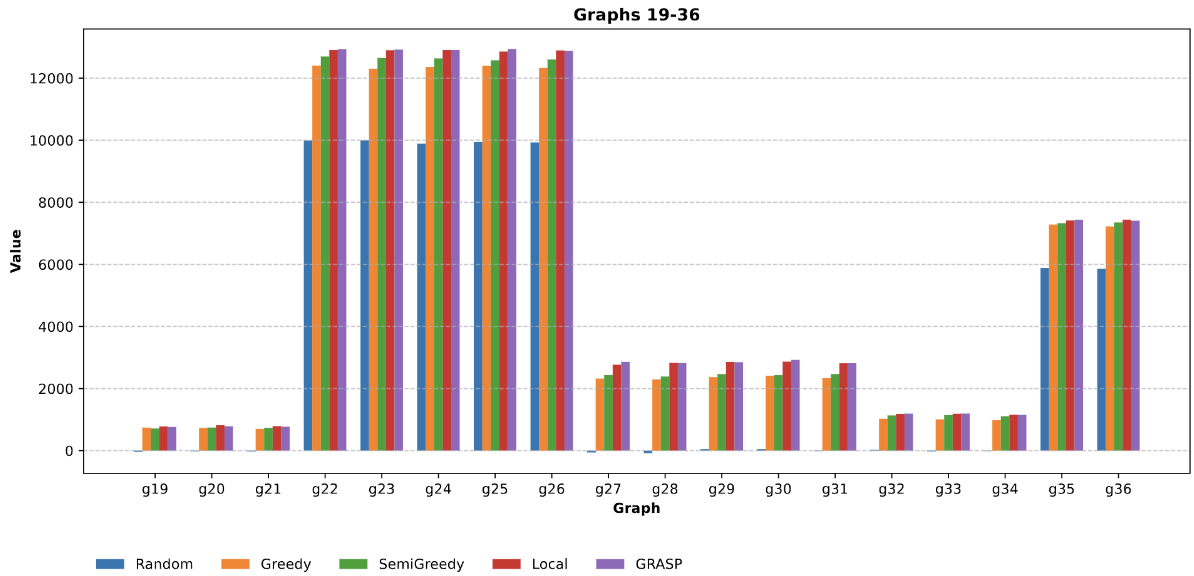
Among all three Constructive algorithms, Simple randomized performed worst overall for all the graphs as it should be because it tends nothing to get the most optimal solution. For some graph with negative edge weights, it performed so bad that the value came negative which is shoen in the bar plot below.

The greedy performed better in sense of finding local best compared to randomized.As it always focus on local optima, it doesn't give the best solution for maxcut.

The semi greedy here approaches for both greedy and randomized way and thow the best performance among the three constructive algorithms which has been shown in the benchmark plot.

Using local search approach and Semi greedy construction GRASP outperformed all other techniques with the hishest outcome and almost near to the known best value or upperbound.





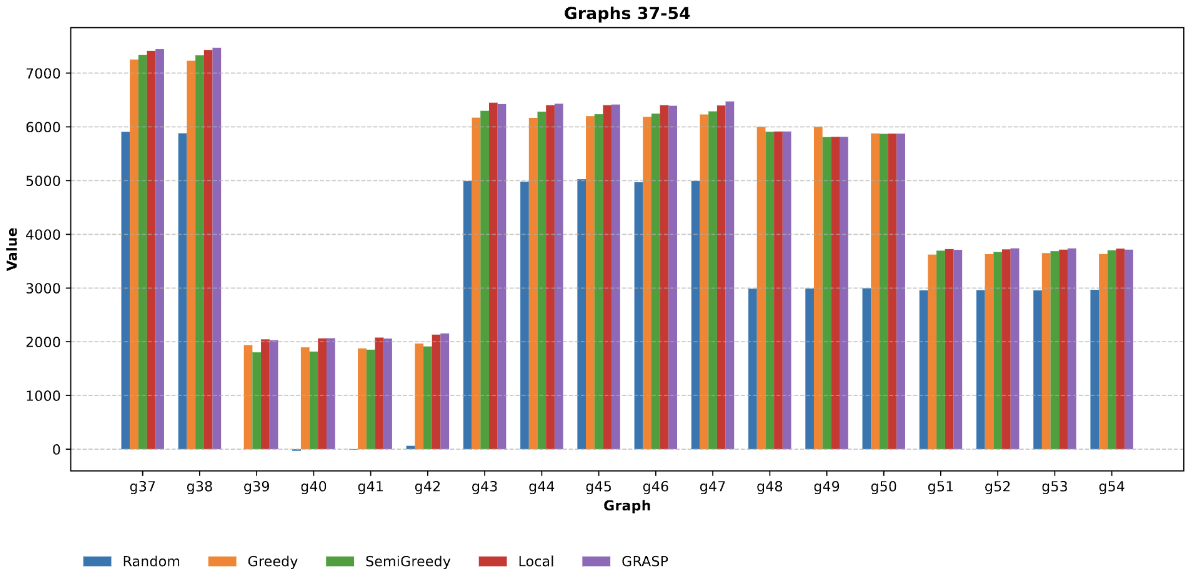


Figure 5: Benchmark score comparison for all algorithms

4 Conclusion

In conclusion, the choice of construction phase significantly influences the behavior and performance of the GRASP algorithm in solving the max-cut problem. The randomized approach offers wide exploration potential but might require more iterations and optimiza- tion steps to reach competitive solutions. The greedy approach provides quick convergence but might get trapped in local optima. The semi-greedy approach finds a middle ground between the two, aiming to balance exploration and exploitation. The average best value of local search, which summarizes the quality of initial solutions, underscores the importance of the construction phase's impact on the overall algorithm's efficiency and effectiveness.

Besides, the GRASP algorithm performs better and better with increasing number of itertaions but take larger time for execution as it combines semi greedy construction phase and local search implementation for each iterations.Though the maxcut value increases with interations but i think it is not that much notable compared to requird computational cost.