# CSE 318 Assignment-02

## Solving the Max Cut Problem By GRASP


Dibya Jyoti Sarkar

2105084


**High Level Descriptions of the Algorithms:**

1. **Randomized Heuristic:**
   The algorithm starts with both partitions X and Y being empty. For each vertex v ∈ V , it is placed in either partition X or Y uniformly at random, with probability 0.5 each. The final cut is determined by the edges crossing between the two partitions. To get a reliable estimate of the cut size, the algorithm is run n times and the results are averaged

2. **Greedy Heuristic:**
   The algorithm starts by placing one vertex to each partition X and Y (initially both are empty) such that each contains an endpoint of the edge with the largest weight. The remaining |V |−2 vertices are then considered one by one. For each unassigned vertex, it is placed into the partition (X or Y ) where it contributes the most to the current partial cut. This placement is done greedily at each iteration.

3. **Semi Greedy Heuristic:**
   In Semi Greedy Heuristic, we introduce a greedy function and a "Restricted Candidate List" (RCL). For each vertex v, the greedy function is evaluated. Based on these values, candidate vertex is selected and inserted into the RCL. For building RCL, we used value based method. In this method, we select all candidates v with greedy value $\geq \mu$ where $\mu$ is:

$$\mu = w_{\{min\}} + \alpha \cdot \left(w_{\{max\}} - w_{\{min\}}\right)$$

and $\alpha$ is a tunable parameter varying from 0 to 1.
The greedy function value of v is :

$$\max\{\sigma_X(v), \sigma_Y(v)\}$$

Where $\sigma_X(v)$ and $\sigma_Y(v)$ are:

$$\sigma_X(v) = \sum_{\{u \in X\}} w_{\{vu\}}$$

$$\sigma_Y(v) = \sum_{\{u \in Y\}} w_{\{vu\}}$$

And $w_{\{min\}}$ and $w_{\{max\}}$ are:

$$w_{\{min\}} = \min\left\{ min_{\{v \in V'\}\sigma_{X(v)}}, min_{\{v \in V'\}\sigma_{Y(v)}} \right\}$$

$$w_{\{max\}} = \max\left\{ max_{\{v \in V'\}\sigma_{X(v)}}, max_{\{v \in V'\}\sigma_{Y(v)}} \right\}$$

4. **Local Search:**

We assume that we have obtained a feasible solution at the end of construction phase $S$ and $\bar{S}$ . In local search, we use an objective function to move vertex v from $S$ to $\bar{S}$ and vice versa. The objective function is:

$$\delta(v) = \begin{cases} \sigma_{S(v)} - \bar{\sigma}_{S(v)}, & if\ v \in S \\ \bar{\sigma}_{S(v)} - \sigma_{S(v)}, & if\ v \in \bar{S} \end{cases}$$

All possible vertices are investigated and best improving replaces current solution. The local search stops when no improving neighbor is found after evaluation of all possible moves (we are stuck in a local optima).

5. **GRASP(Greedy Randomized Adaptive Search Procedure):**

GRASP is a randomized multistart iterative method for computing good quality solutions of combinatorial optimization problems.

In GRASP, we have two phases:

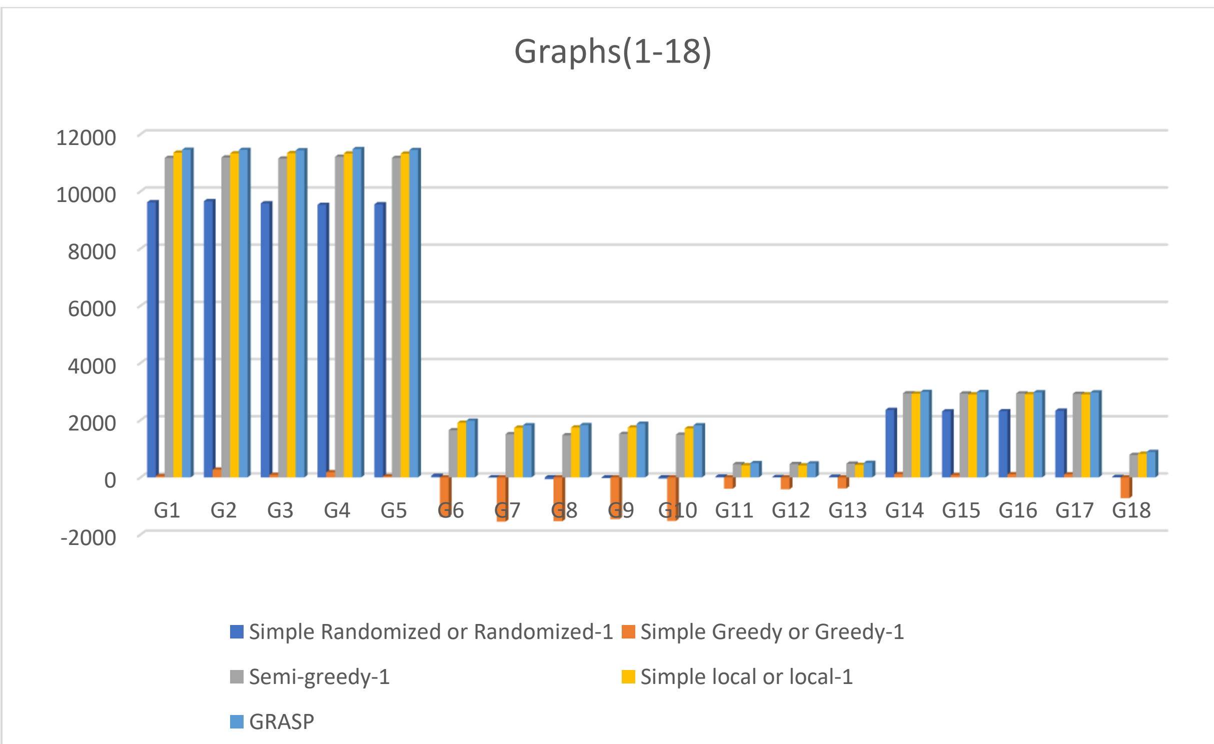
I) We have a construction phase where we use semi greedy method to select partitions

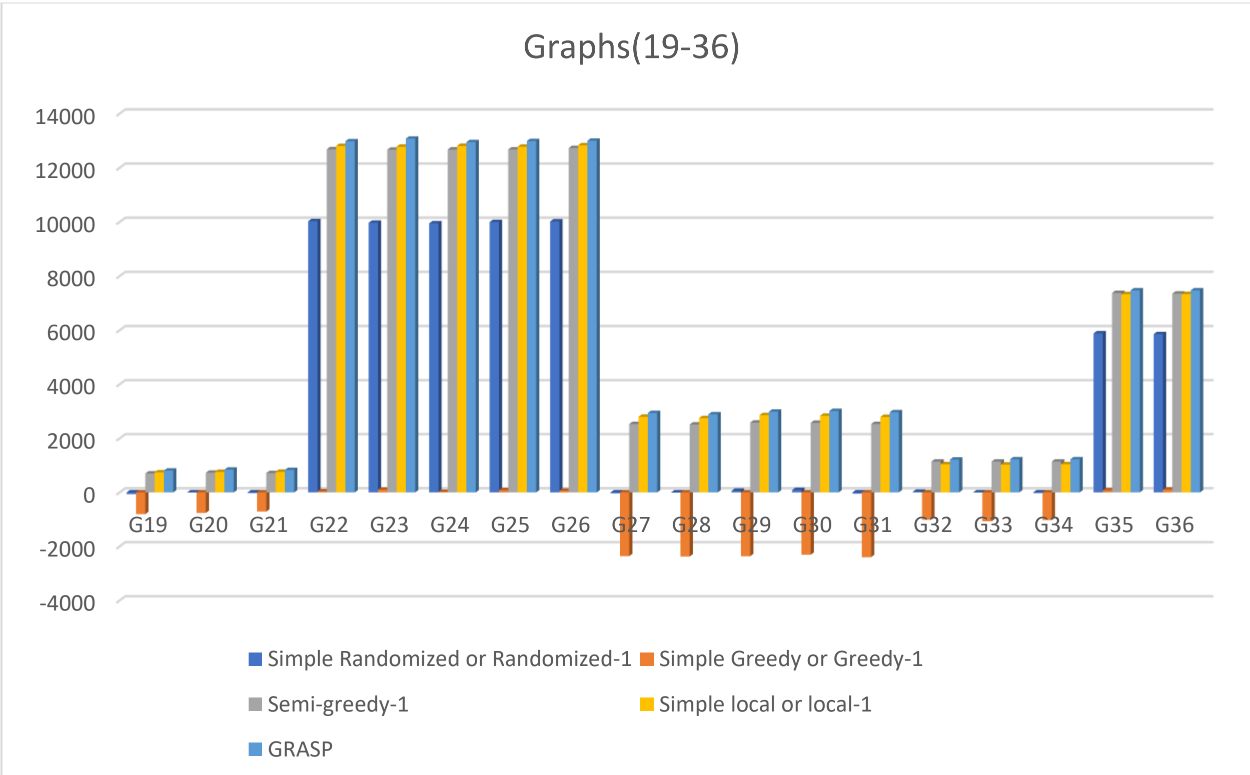
II) Then a local search phase is used to find locally optimized solution

## Comparison of Algorithms:

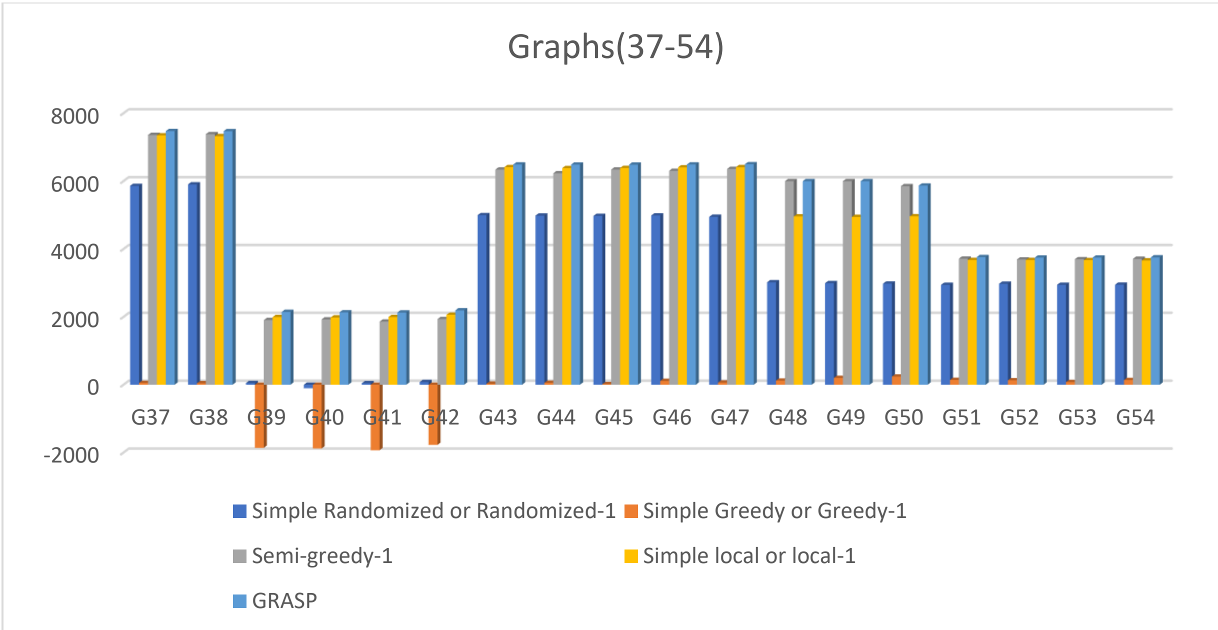
From the plots, it is clearly visible that GRASP performs very well than others where as Greedy's performance is very low compared to others. After GRASP, Semi-Greedy also performs well followed by local search and randomized heuristic.

## Plots:





Graphs(19-36)
14000
12000
10000
8000
6000
4000
2000
0
-2000
-4000
G19
G20
G21
G22
G23
G24
G25
G26
G27
G28
G29
G30
G31
G32
G33
G34
G35
G36
Simple Randomized or Randomized-1
Simple Greedy or Greedy-1
Semi-greedy-1
Simple local or local-1
GRASP


Graphs(37-54)
8000
6000
4000
2000
0
-2000
G37
G38
G39
G40
G41
G42
G43
G44
G45
G46
G47
G48
G49
G50
G51
G52
G53
G54
Simple Randomized or Randomized-1
Simple Greedy or Greedy-1
Semi-greedy-1
Simple local or local-1
GRASP