Report on Max-Cut Problem: Algorithm Comparison

Introduction

The Max-Cut problem is a well-known optimization problem that involves partitioning the vertices of a graph into two sets, such that the sum of the weights of the edges between the two sets is maximized. In this report, we compare the performance of three different algorithms for solving the Max-Cut problem: Randomized Partition, Greedy Heuristic, and Greedy Randomized Adaptive Search Procedure (GRASP).

Algorithms and Methodology

1. **Randomized Partition (Alpha = 0):**
    - This algorithm randomly assigns vertices to either of the two partitions without considering edge weights.
    - The resulting cut weight is computed based solely on the random assignments.
2. **Greedy Heuristic (Alpha = 1):**
    - The Greedy Heuristic algorithm aims to achieve a good solution by greedily selecting the best available vertex for each partition.
    - It considers a vertex and chooses in which set it will be the one that maximizes the cut weight.
3. **GRASP (Greedy Randomized Adaptive Search Procedure) (Alpha = 0 to 1):**
    - The GRASP algorithm is an iterative procedure that combines the greedy approach with randomization and local search.
    - The alpha parameter controls the balance between greediness and randomization in the algorithm.

Results and Analysis

Based on our experiments, we observed the following trends in the performance of the three algorithms:

- **Randomized Partition:**
    - This algorithm performs the worst among the three.
    - Since it assigns vertices randomly without considering edge weights, the resulting cut weight tends to be lower.
- **Greedy Heuristic:**
    - The Greedy Heuristic algorithm provides moderate performance.

    - It takes into account the weight of edges connected to each vertex, leading to better solutions than randomization.
- **GRASP (Greedy Randomized Adaptive Search Procedure):**
    - The GRASP algorithm consistently produces the best solutions.
    - The algorithm's ability to strike a balance between greediness and randomness allows it to explore a wider solution space.
    - By iteratively improving solutions through local search, GRASP converges to better solutions.

Conclusion

In conclusion, our experiments demonstrate that the Greedy Randomized Adaptive Search Procedure (GRASP) outperforms both the Randomized Partition and Greedy Heuristic approaches for the Max-Cut problem. GRASP's ability to adapt its balance between greediness and randomness, coupled with its local search phase, enables it to consistently find better solutions. While the Randomized Partition algorithm produces suboptimal results due to its randomness, the Greedy Heuristic provides intermediate results by considering edge weights.

Recommendations

For future research and applications, we recommend exploring different values of the alpha parameter in the GRASP algorithm to find an optimal balance between greediness and randomness. Additionally, considering more advanced local search techniques, such as simulated annealing or genetic algorithms, could further enhance the performance of the algorithms.