Graph Partitioning Algorithms: A Comparative Study

Ziad Eliwa 900246124 - Mostafa Abdelwahed 900243064
Ali Mohamed 900246190 - Farouk Youssef 900243941

Abstract—

I. Introduction

Abstraction is one of the most important ideas of computer science. Computer scientists and researcher across the span of the study and modelling of real-world complex entities have utilized modelling them as *graphs*. In each case, there is an underlying network that encapsulate the important properties that help us understand these entities and draw meaningful insights from them. In addition to the huge contribution that graphs brought to the advancment in computer science, it brought with its complexities. Due to the enormous data graphs can represent, they often fall in the area of *NP Hard Problems*, starting from the most famous Travelling Salesman Problem (TSP) to graph coloring (k-coloring for k $\geq$ 3), Hamiltonian paths and more. Graph Partitioning stands as one of the most challenging problems in computer science. Graph partitioning is essential for the fields of parallel computing, high performance computing (HPC) and load balancing in networks, and VLSI in digital logic design where vertices resemble logical units and edges resemble wires. This paper goes into depth of some heurisitcs and greedy algorithms that allow local use of the power of graph partitioning. [1]

"The graph partitioning problem is increasing with the emergence of Big Data. Handling tremendous volumes of graph data requires an efficient graph processing system and especially a high-quality graph partitioning approachs to cope with graph application needs." [2].

II. Background

A. Theoritical Background

Graph: Let graph $G = (V, E)$ be weighted, undirected graph with vertices $v_i = 1, 2, \ldots n$. Define w_{ij} as the non-negative weight between v_i and v_j, $i \neq j$. If $(v_i, v_j) \in E$, then $w_{ij} > 0$, otherwise $w_{ij} = 0$.

Graph partition / k-partition:

Cut: A cut C is a partition of V which seperate into S and $V - S$. A cut-set of $C = (S, V - S)$ is defined as the set of edges $\{(u, v) | u \in S, v \in V - S\}$.

P vs NP: The P vs NP Problem is one of the mysteries of theoritical computer science and one of the seven millenium problems. P is the complexity class of problems that are solvable in polynomial time through deterministic algorithms. NP (Non-deterministic polynomial time) is a complexity class of problems that have no deterministic algorithms to solve it in polynomial time (Unless P = NP), while there exists

exponential time algorithm through brute force and complete search methods, but these are not solvable by today's machines, unless perhaps solved with novel ideas such as quantum computation.

NP-Hard Problems: In the class of NP Problems, some problems are called as *NP-Complete*. All NP-Complete problems are decision problems (i.e. with Yes or No answer) such as the existence of a hamiltonian path. Problems are called *NP-Hard* if they are at least as hard as NP-Complete problems. NP-Hard can be present in its own complexity class in which NP-Complete problems is a subset of it, but it remains undecidable.

The graph partition problem is considered NP-Hard as it can be reduced from Max-Cut problem which itself is NP-Hard. This paper discusses the approximation and heurisitcs that was motivated by the computational complexity of the problem.

Heuristics: Heuristic methods are greedy and approximate algorithms that are concerned to use the available resources to reach an enough solution to the problem. Exact algorithms allow for optimal solutions but are highly limited to small graphs.

B. Problem Definition

A graph partition or a k-partition of graph $G = (V, E)$ is partitioning V into k disjoint parts that cover V.

$$V_1 \cup V_2 \cup \cdots \cup V_k = V.$$

For a partition to be balanced, we have to satisfy the Balance constraint $|V_i| \in \{\lfloor n/k \rfloor, \lceil n/k \rceil\}, |V| = n$ and the number of edges with endpoints in different parts is minimized.

The optimization problem is to find such a partition:

$$\min_{V_1, \ldots, V_k} \text{cut}(V_1, \ldots, V_k) \quad \text{subject to the balance constraint.}$$

III. Literature Survey

IV. Experiment

V. Results and Discussion

VI. Conclusion & Future Work

References

[1] Graph partition problem - an overview — sciencedirect topics. (2025). https://www.sciencedirect.com/topics/computer-science/graph-partition-problem

[2] Sakouhi, C., Khaldi, A. & Ghezala, H. Distributed framework for high-quality graph partitioning. J Supercomput 81, 1418 (2025). https://doi-org.libproxy.aucegypt.edu/10.1007/s11227-025-07907-2