

IE 492: Project



Project Proposal

Generating Random Perfect Graphs

Advisor: Tınaz Ekim

Ali Can Milani	2018402171
Aral Dörtoğul	2018402108
Fatih Mehmet Yılmaz	2017402066

Industrial Engineering
Boğaziçi University

March 3, 2023

1 Problem Description

Graph theory is a mathematical field concerned with the study of graphs, which are mathematical structures consisting of nodes or vertices connected by edges or arcs. Graph theory has numerous applications in various fields such as computer science, engineering, physics, social sciences, and operations research.

Graphs are used to model many real-world situations, such as social networks, transportation networks, electrical circuits, and communication networks. For example, in social networks, graphs are used to represent people as nodes and relationships between them as edges. In transportation networks, graphs are used to model the routes between different cities as edges and the cities themselves as nodes. In electrical circuits, graphs are used to represent the connections between different components as edges and the components themselves as nodes.

Perfect graphs are a class of graphs that have several interesting properties, making them useful for modeling real-world problems in various fields such as transportation, scheduling, and communication networks. Some well-known subclasses of perfect graphs, including chordal and interval graphs, have been extensively studied in the literature due to their practical applications.

Graph algorithms for solving these problems mostly perform in exponential time. However, algorithms designed for specific graph classes such as perfect graphs perform in polynomial time. To verify these algorithms and evaluate their performance, algorithms need to be tested for the whole set of graphs of its specific class. So, generating random perfect graphs can be a practical solution to test these algorithms and analyze their performance. This study is for designing algorithms that will generate perfect graphs and for comparing those algorithms.

2 Literature Review

An odd hole is an induced cycle of length at least 5 with an odd number of vertices. An odd anti-hole is the complement of an odd hole. Chudnovsky, Robertson, Seymour, and Thomas [8] proved the Strong Perfect Graph Theorem (SPGT), which states that a graph is perfect if and only if it does not contain an odd hole or the complement of the odd hole. Therefore, determining whether a graph is perfect can be done by identifying the presence of an odd hole, which can be found either in the graph itself or in its complement. In other words, recognizing the perfectness of a graph is equivalent to

identifying whether there exists an odd hole in the graph or its complement.

In the literature, perfect graphs have been widely studied. In 1963, Claude Berge coined the term perfect graph, and he conjectured both the perfect graph theorem and the strong perfect graph theorem [2]. Creating random graphs in general is a starting point in the domain. The Erdos-Renyi model is a method for creating random graphs given the density and number of vertices of a graph [11]. This model selects an edge at random and adds it to the graph with a certain probability. The goal of generating a random graph is to ensure that every graph has a non-zero probability of being generated, and that the probability distribution ideally is uniformly distributed for all possible graphs.

Some recognition algorithms for perfect graph class are proposed in the literature. Chudnovsky et. al. proposed a recognition time algorithm for detecting perfect graphs that has a polynomial time complexity [7]. However, this algorithm fails to give the difference between an odd hole and an odd anti-hole. This results in the fact that with this algorithm, whether a graph is perfect or not can be decided but the reason behind this result cannot be understood.

In a paper by Chudnovsky, Seymour, and Vuskovic [6], a polynomial-time algorithm for detecting odd holes in graphs is described. This result is significant since in this paper, the reason behind the perfectness or not-perfectness can be explained. This algorithm can differentiate between an odd hole and the complement of the odd hole. In another paper by Oylum Şeker, Tınaz Ekim, and Z. Caner Taşkın [10], a perfect graph generation algorithm called **PerfectGen** is proposed using some available perfect graphs, which is enumerated up to 11 vertices by McKay [1] and perfection-preserving operations. The authors created perfect graphs with many vertices using operations that preserve perfection. Their approach involved combining small, already known perfect graphs using one of six specific operations that maintain perfection, resulting in a larger perfect graph. They repeated this process until the desired number of vertices was reached. However, this approach does not guarantee that every perfect graph can be generated uniformly at random.

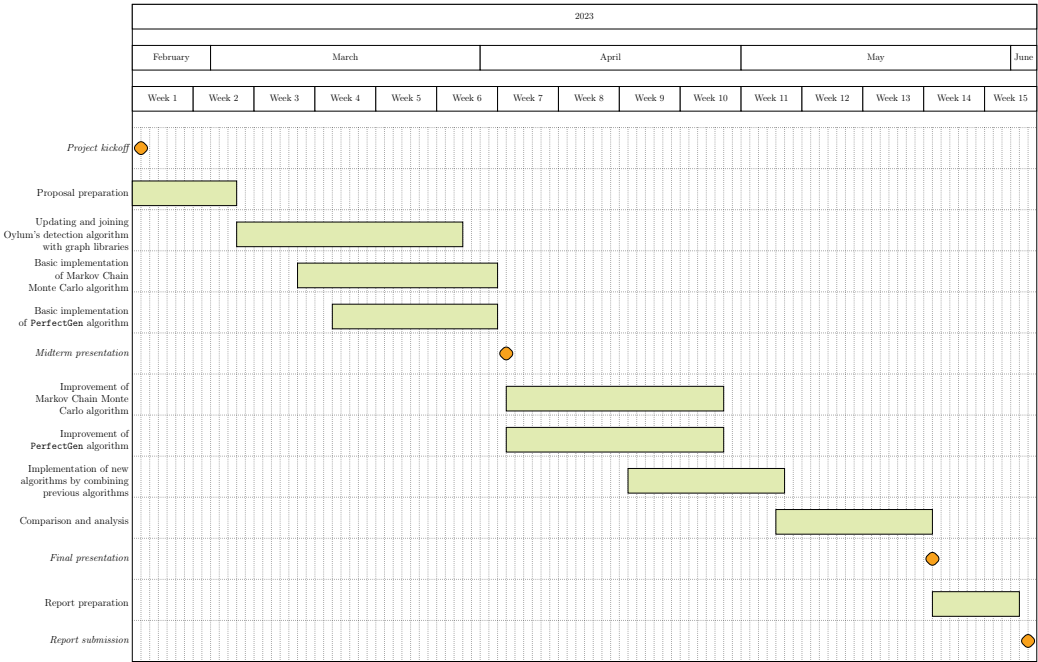
3 Methodology

In this project, one of the algorithms that will be used for random perfect generation follows a programming approach, where it will take any number of perfect graphs from a pool and combine them to generate a new perfect graph using specific combining methods. We will improve Algorithm **PerfectGen** by using possible new operations such as amalgam [5], 2-amalgam [3] or 2-join, double star cut-set or skew partition [9] in addition to the ones defined in the paper by Oylum Seker, Tınaz Ekim, and Z. Caner Taşkın [10]. Also, another possible improvement to Algorithm **PerfectGen** is adding more perfect graphs to the starting pool. This improvement will expand the range of potential random perfect graphs that can be generated.

Another one of the algorithms in the project will follow Markov Chain Monte Carlo methods. It will generate a random graph and make clever random edge changes. Then the polynomial-time recognition algorithm will be implemented from the literature [10], to check if the generated graph is a perfect graph, if not the algorithm will make clever random edge changes again and repeat the process. Almost no random graph generated firstly by using Erdos–Renyi Model will be perfect. Therefore, we will use a better technique, where C5-free or (Clean Odd Cycle)-free random graphs are generated, and then the edges are modified until reaching perfectness. This method is based on the idea that almost all C5-free graphs are perfect when the number of vertices goes to infinity [4].

At last, the perfect graph generators developed will be compared experimentally with respect to the run time and the sizes of the generated graphs, and other related parameters of the generated graphs such as number of maximal cliques, clique number, independence number.

4 Project Timeline



References

- [1] Brendon McKay. *Graphs*. 2016. URL: <http://users.cecs.anu.edu.au/%E2%88%BCbdm/data/graphs.html>.
- [2] Claude Berge. “Perfect Graphs”. In: *Six Papers on Graph Theory* (1963), pp. 1–21.
- [3] G. Cornuéjols and W.H. Cunningham. “Compositions for perfect graphs”. In: *Discrete Mathematics* 55.3 (1985), pp. 245–254. ISSN: 0012-365X. DOI: [https://doi.org/10.1016/S0012-365X\(85\)80001-7](https://doi.org/10.1016/S0012-365X(85)80001-7). URL: <https://www.sciencedirect.com/science/article/pii/S0012365X85800017>.
- [4] Hans Jürgen Prömel and Angelika Steger. “Almost all Berge Graphs are Perfect”. In: *Combinatorics, Probability and Computing* 1.1 (1992), pp. 53–79. DOI: 10.1017/S0963548300000079.
- [5] M. Burlet and J. Fonlupt. “Polynomial Algorithm to Recognize a Meyniel Graph”. In: *Topics on Perfect Graphs*. Ed. by C. Berge and V. Chvátal. Vol. 88. North-Holland Mathematics Studies. North-Holland, 1984, pp. 225–252. DOI: [https://doi.org/10.1016/S0304-0208\(08\)72938-4](https://doi.org/10.1016/S0304-0208(08)72938-4). URL: <https://www.sciencedirect.com/science/article/pii/S0304020808729384>.
- [6] Maria Chudnovsky et al. “Detecting an Odd Hole”. In: *J. ACM* 67.1 (Jan. 2020). ISSN: 0004-5411. DOI: 10.1145/3375720. URL: <https://doi.org/10.1145/3375720>.
- [7] Maria Chudnovsky et al. “Recognizing Berge Graphs”. In: *Combinatorica* 25 (Mar. 2005), pp. 143–186. DOI: 10.1007/s00493-005-0012-8.
- [8] Maria Chudnovsky et al. “The Strong Perfect Graph Theorem”. In: *Annals of mathematics* 164.1 (Jan. 2003), pp. 51–229. ISSN: 0003-486X. DOI: 10.4007/annals.2006.164.51.
- [9] Nicolas Trotignon. *Perfect graphs: a survey*. 2013. DOI: 10.48550/ARXIV.1301.5149. URL: <https://arxiv.org/abs/1301.5149>.
- [10] Oylum Şeker, Tınaz Ekim, and Zeki Caner Taşkın. “An Exact Cutting Plane Algorithm to Solve the Selective Graph Coloring Problem in Perfect Graphs”. In: *European Journal of Operational Research* 291(1) (2021), 67–83. DOI: <https://doi.org/10.1016/j.ejor.2020.09.017>.
- [11] Paul Erdos and Alfred Renyi. “On the evolution of random graphs”. In: *Publ. Math. Inst. Hungary. Acad. Sci.* 5 (1960), pp. 17–61.