LITERATURE REVIEW: Implementation of the Parallel Algorithm for Planar Subgraph Isomorphism

Marc Vicuna
School of Computer Science
Carleton University
Ottawa, Canada K1S 5B6
marcvicuna@cmail.carleton.ca

October 6, 2021

1 Introduction

The Subgraph Isomorphism problem consists of looking for the occurrence of a pattern graph H with k vertices in a target graph G with n vertices. Subgraph Isomorphism is a generalization of many NP-complete problems, such as finding a Maximum Clique, Longest Path or Hamiltonian Cycle. Thus, the Subgraph Isomorphism is itself NP-complete[9], but it is also proven to be NP-complete in subcases of bounded degree graphs[7] and planar graphs[8]. Planar Subgraph Isomorphism algorithms have a large scope of application, including graph databases[15], electronic design[4] and astronomy[13].

The latter case, the Planar Subgraph Isomorphism problem, where both the target graph G is planar, has seen some interesting developments. Recently, the publication *Parallel Planar Subgraph Isomorphism and Vertex Connectivity*[11] gave an improved algorithm for this problem, with an original set of techniques. utilizing parallel computing and fixed-parameter tractable (FPT) techniques.

We implemented the algorithm to verify the improved bounds of the algorithm and optimize the work time, to evaluate its use in practice. We focus on implementing the algorithm, confirming or infirming theoretical results and finding implementation-driven efficiency improvements.

2 Literature Review

2.1 Planar Subgraph Isomorphism

At time of publication, the best algorithms for Planar Subgraph Isomorphism are listed in Table 1 (found in [11]), with their bounds on work and depth. Note the complexity is written with n as vertex number in G, and k as vertex number in H. In practice, n is much greater than k, thus, algorithms presented relatively prioritize lowering complexity in n.

To begin, the Color Coding algorithm [1] from 1995 is introduces many of the techniques used in the other algorithms, and its principles are still used in many of the current algorithms for finding Maximum Clique (more in Section 2.3). Moreover, it has the lowest bound on depth, keeping it relevant even today. Then, Eppstein's[5] algorithm lowered the work bound greatly and instituted the structural foundations for all the other algorithms

Table 1: Bounds for deciding planar subgraph isomorphism

	Work	**Depth**
Alon et al.*[1]	$e^k n^{\Theta(\sqrt{k})} \log n$	$\Theta(k \log n)$
Eppstein [5]	$O(2^{3k \log_2(3k+1)} n$	$\Theta(kn)$
Dorn [3]	$O(2^{18.81k} n)$	$O(2^{18.81k} n)$
Formin et al.*[6]	$2^{O(3k \log k)} n^{O(1)}$	$2^{O(3k \log k)} n^{O(1)}$
Gianinazzi et al.*[11]	$O(2^{3k \log_2(3k+1)} n \log n)$	$O(k \log^2 n)$

(*) Monte Carlo algorithm, bounds with high probability.

(especially Gianinazzi's algorithm). Both Dorn[3] and Formin's[6] algorithms improve on the k work complexity while worsening the k depth complexity, but Formin's article also set theoretical lower bounds on the algorithms considering sequential techniques.

2.2 Gianinazzi's algorithm

To overcome those bounds, Gianinazzi's algorithm changed the playing field by matching Eppstein k work complexity with an improvement in n depth complexity (from linear to logarithmic, at the cost of n near linear work. Whether this is a significant improvement is arguable, but these bounds are obtained through the original use of both parallel and FPT techniques.Contrary to previous algorithms, it is not based on characteristics unique to planar graphs and also works on all minor-closed families of graphs of locally bounded treewidth. Moreover, the algorithm can easily be extended to disconnected pattern graphs and the capacity to list all occurences.

Moreover, each step of the algorithm is on its own an area of active research (Monte Carlo algorithms, reachability problems, unsupervised clustering), indicating improvement opportunity at any step. Gianinazzi's algorithm has three steps:

1. Low diameter decomposition of G into vertex-disjoint random clusters. This step allows the parallel execution of subsequent steps.

2. Cover with BFS for decomposition to suitable subpaths.

3. Using dynamic programming and reachability properties within and between subgraphs of G, finding valid partial matches.

We refer to these steps as Step 1, Step 2 and Step 3.

2.3 Implementations

Implementations of the Subgraph Isomorphism algorithms are numerous, the latest of which the VF2++ algorithm[14], with high efficiency. However, implementations optimized for solving the planar case are scarce, . However, techniques found in the algorithms for Planar Subgraph Isomorphism are found in some implementation of Maximum Clique problems, as shown here: [2] [10] [12].

References

[1] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, July 1995.

[2] Lijun Chang. Efficient maximum clique computation over large sparse graphs. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 529–538, New York, NY, USA, 2019. Association for Computing Machinery.

[3] Frederic Dorn. Planar subgraph isomorphism revisited. *CoRR*, abs/0909.4692, 2009.

[4] Jorge Echavarria, Alicia Morales-Reyes, René Cumplido, Miguel A. Salido, and Claudia Feregrino-Uribe. IP-cores watermarking scheme at behavioral level using genetic algorithms. *Engineering Applications of Artificial Intelligence*, 104:104386, 2021.

[5] David Eppstein. Subgraph isomorphism in planar graphs and related problems. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '95, page 632–640, USA, 1995. Society for Industrial and Applied Mathematics.

[6] Fedor V. Fomin, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 515–524, 2016.

[7] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, STOC '74, page 47–63, New York, NY, USA, 1974. Association for Computing Machinery.

[8] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, STOC '74, page 47–63, New York, NY, USA, 1974. Association for Computing Machinery.

[9] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman amp; Co., USA, 1990.

[10] Lukas Gianinazzi, Maciej Besta, Yannick Schaffner, and Torsten Hoefler. Parallel algorithms for finding large cliques in sparse graphs. In *Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '21, page 243–253, New York, NY, USA, 2021. Association for Computing Machinery.

[11] Lukas Gianinazzi and Torsten Hoefler. Parallel planar subgraph isomorphism and vertex connectivity. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '20, page 269–280, New York, NY, USA, 2020. Association for Computing Machinery.

[12] A. Haverly and S. López. Implementation of grover's algorithm to solve the maximum clique problem. In *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 441–446, 2021.

[13] David S Johnson. The np-completeness column: An ongoing guide. *Journal of Algorithms*, 11(1):144–151, 1990.

[14] Alpár Jüttner and Péter Madarasi. Vf2++—an improved subgraph isomorphism algorithm. *Discrete Applied Mathematics*, 242:69–81, 2018. Computational Advances in Combinatorial Optimization.

[15] David Luaces, José R.R Viqueira, José M Cotos, and Julián C Flores. Efficient access methods for very large distributed graph databases. *Information sciences*, 573:65–81, 2021. Publisher: Elsevier Inc.