

Edmonds' Matching Algorithm Project Proposal

AMIR H. DADPOUR, University of Waterloo, Canada

Edmonds' Matching Algorithm (also known as Edmonds' Blossom Algorithm) [5] is a classic algorithm to efficiently find a maximum matching in a graph by iteratively finding augmenting paths and *shrinking* odd cycles as needed. The algorithm is of historic interest as it was the first algorithm to perform this operation in polynomial time. Additionally, its linear programming description was a part of Jack Edmonds' major breakthrough in combinatorial optimization [4]. The key goal for this project is to have a functional implementation (description) of the algorithm in Rocq Prover followed by formally proving the correctness and termination of the algorithm for simple unweighted graphs. The major tasks lies in proving the *preservation* of the augmenting paths when shrinking the odd cycles. This will require a considerable work proving results regarding graph minors.

CCS Concepts: • **Mathematics of computing** → **Graph algorithms; Matchings and factors.**

1 DESCRIPTION

A matching for a graph $G = (V, E)$ is a set of edges M with the property that every vertex in the graph is only incident with at most one edge in M . Matchings can be used to model various problems in different fields of computer science [2]. A maximum matching for G is a matching which has the largest possible number of edges.

Berge's Theorem provides a useful characterization of maximum matchings based on the concept of augmenting paths [1]. For a matching M , an M -augmenting path is a path that starts from an unmatched vertex, alternates between edges in M and edges not in M and ends on another unmatched vertex. One can easily observe that the existence of such path will show that M is not maximum as the symmetric difference of the path with M yields a larger matching. Berge's Theorem states that a matching is maximum if and only if no augmenting path exists.

The idea of an algorithm to use this fact to start from an empty matching and iteratively find augmenting paths until there are non left seems intuitive, efficient and easy to prove. However, for a general graph, such algorithm will have difficulties dealing with odd cycles. The idea of Edmonds' Algorithm is use to *shrink* such odd cycles by contracting the edges.

It hence follows that the difficulty in proving the correctness and termination of Edmonds' Algorithm lies in proving that an augmenting path is preserved under the contraction of odd cycles, and so are the number of remaining augmenting paths. By proving such results in addition to the machinery needed to define the needed definitions and formally implement the algorithm, I plan to prove the algorithm in Rocq Prover.

2 FUNCTIONALITIES

To start with, the intended functionality are to be built on top of the current graph theory infrastructure in Rocq Prover and especially the definitions of graph minors [3]. Given these definitions, it remains to:

- (1) Formally define the concept of maximum matching using the Berge Theorem.
- (2) Formally state and implement the algorithm.
- (3) Formally define the concept of shrunk cycles.
- (4) Prove the *preservation* of augmenting paths under contraction of odd cycles.
- (5) Use these results show that the algorithm terminates and yields a maximum matching.

This list of functionalities are preliminary and will be more descriptive and concrete as the project proceeds.

3 ROADMAP

I expect the bulk of the work to be at the beginning of the proof to setup the definitions and the algorithm itself in a manner that is provable in Rocq Prover. Once done, I plan to sketch out proof on paper to get a better understanding of the sequence of lemmas and theorem I need for a complete proof. My intend is to finish all of the above before the presentation in March. Subsequently, I will spend the last weeks completing the actual proof (the code) in Rocq Prover.

- Review the graph theory library and its existing infrastructure. Prepare definitions of graphs, minors, and maximum matchings suitable for the proof. (Week of March 2)
- Implement the algorithm in Rocq Prover using the given definitions. (By March 9)
- Sketch out the proof and required lemmas on paper. (By March 16)
- Write the statements of lemmas and theorems in Rocq Prover based on the proof on paper. (By March 23)
- Prepare the presentation. (By March 26)
- Complete the proof in Rocq Prover. (By April 6)
- Complete the final report. (By April 12)

REFERENCES

- [1] Claude Berge. 1957. TWO THEOREMS IN GRAPH THEORY. *Proceedings of the National Academy of Sciences* 43, 9 (1957), 842–844. <https://doi.org/10.1073/pnas.43.9.842> arXiv:<https://www.pnas.org/doi/pdf/10.1073/pnas.43.9.842>
- [2] Horst Bunke. 2000. Graph matching: Theoretical foundations, algorithms, and applications. In *Proc. Vision Interface*, Vol. 2000. 82–88.
- [3] Christian Doczkal and Damien Pous. 2020. Graph Theory in Coq: Minors, Treewidth, and Isomorphisms. *Journal of Automated Reasoning* (2020). <https://doi.org/10.1007/s10817-020-09543-2>
- [4] Jack Edmonds. 1965. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics* (1965), 125. <https://api.semanticscholar.org/CorpusID:15379135>
- [5] Jack Edmonds. 1965. Paths, Trees, and Flowers. *Canadian Journal of Mathematics* 17 (1965), 449–467. <https://doi.org/10.4153/CJM-1965-045-4>

Received 16 February 2025