University of Glasgow | School of Computing Science

Summer Research Project

Approximate Maximum Matchings in Planar Graphs

Jake Macaleb Haakanson

August 18, 2022

Abstract

A matching in a graph $G = (V, E)$ is a subset of edges $M \subseteq E$ such that no edges in M share common vertices. The first polynomial-time algorithm for finding maximum matchings was introduced in the 1960s [6] and subsequent improvements have led to several $O(|E| \cdot \sqrt{|V|})$ algorithms [11, 2, 9] for exact maximum matchings. While the theoretical time complexity has been lowered, maximum matching algorithms may still be improved in practice, particularly for larger graphs. This report details an attempt to improve the speed of Edmond's blossom algorithm [6] by approximating maximum matchings in planar graphs which adhere to certain constraints, particularly that the graph is bound to degree 3 through "vertex splitting" [17, 13], and that the k-stars and k-double-stars are removed [5, 4]. Unfortunately due to the nature of the separator library used it was not possible to carry out experimentation on near-planar graphs, and therefore this report gives an overview of carrying out the proposed graph reductions and why they may be a desirable preprocessing step in finding maximum matchings.

Contents

1 Introduction and Background 1

2 Preliminaries 2

3 Graph Modifications 3
- 3.1 Vertex Splitting 3
- 3.2 k-Star and k-Double-Star Removal 4
- 3.3 Order of Reductions 5
- 3.4 r-Division of a Planar Graph 6

4 Matching Algorithm 7

5 Conclusion 8

Bibliography 9

1 Introduction and Background

Given a graph $G = (V, E)$ a matching in G is a subset of edges $M \subseteq E$ where edges in M share no common vertices. A *maximum matching* is a matching in G such that M is as large as possible and this will be the focus of this report. Graph matchings play a role in many parts of computer science and have found use in other fields such as chemistry and physics. Edmond's blossom algorithm, the first polynomial-time maximum matching algorithm, was introduced in the 1960s [6] and built upon earlier work in finding augmenting paths [1, 14]. A complex modification of this algorithm was introduced in the 1980s [11] with a time complexity of $O(|E| \cdot \sqrt{|V|})$ which is the fastest time complexity of other modern exact algorithms [2, 9]. This report contributes what is thought to be a novel approach to approximating maximum matchings in planar graphs through a combination of graph reductions and a graph decomposition. The first reduction, known as *vertex splitting* and discussed in section 3.1, was initially introduced in a fast approach in finding *perfect matchings*[1] in planar graphs of bounded degree [17]. This reduction also found later use in finding maximum matchings in planar graphs of bounded degree [13] furthering earlier work using a Guassian elimination approach [12]. The second of the reductions aims to remove specific subgraphs from a graph, the k-stars and k-double-stars as defined in definitions 3.2 and 3.3, this reduction is explained in more detail in section 3.2. This reduction was first employed in finding maximum matchings in a distributed computational setting [5] and proved a suitable approximate guarantee[2] can be achieved by this reduction. This reduction was also employed in a very recent approach to the maximum matching problem in a similar context [4]. The approach detailed in this report combines the two previous reductions with a graph decomposition based on the planar separator theorem [16, 10] and more in-depth details of this decomposition can be found in section 3.4. The planar separator theorem is an approach to partitioning a graph into disjoint subgraphs, each of which may contain no more than $\frac{2}{3} \cdot |V|$ vertices. This partitioning process is repeatedly applied to a graph and the resulting subgraphs until all subgraphs contain no more than some parameter r vertices [7]. A graph is planar if it can be embedded in the plane without edges intersecting. True planar graphs rarely occur in practical settings and therefore *near-planar* graphs may be used in their place. Near-planar graphs can include road network and autonomous systems graphs, as well as many more different types of graph. The rest of this report is as follows. Section 2 outlines some important definitions used throughout this report. Section 3 covers the reductions which ensure certain properties exist within the graph and details the approach to partitioning the graph for a "divide-and-conquer" approach to the matching problem, and Edmond's blossom algorithm is discussed in section 4. Unfortunately it was not possible to carry out any experimentation throughout this project, and therefore section 5 summarises the work accomplished during this project, the limitations of its approaches, and opportunities for further work or continuation of the project.

[1] A matching is perfect if all vertices in the graph are matched.
[2] This approximate guarantee is proven by Lemmas 5 and 6 of [5].

2 Preliminaries

A *graph* G is defined as $G = (V, E)$ where V represents a set of *vertices*, or *nodes*, and E represents a set of unordered pairs of vertices called *edges* where $E = \{(u, v) | u, v \in V \wedge u \neq v\}$. Vertices that share an edge are referred to as *adjacent*, and vertices are called *incident* to edges, or edges incident to vertices. The *degree* of a vertex, $deg(v)$ for some $v \in V$, is the number of edges in E incident to v. Further to this for some subset of vertices $S \subseteq V$, $deg(v, S)$ is the number of vertices in S that v is adjacent to in the original graph. This report considers *planar graphs*, graphs where edges may intersect only at their endpoints, or alternatively at common vertices.

Definition 2.1 (Planar Graph) *A graph $G = (V, E)$ is planar if its edges only intersect at their endpoints, or common vertices. There are some properties of planar graphs exploited throughout this report, in particular that the number of edges is bound by the number of vertices because $|E| \leq 3 \cdot |V| - 6$ or $|E| \leq 2 \cdot |V| - 4$ where there are no cycles of of length 3 [8].*

A *matching* in G is a set of edges $M \subseteq E$ such that edges in M share no common vertices. A vertex incident to an edge in M may be referred to as *saturated*, *M-saturated*, or *matched* and vertices not incident to edges in M are referred to as *unsaturated*, *unmatched*, or *exposed*.

Definition 2.2 (Maximum Matching) *A maximum matching (or maximum cardinality matching) $M \subseteq E$ in $G = (V, E)$ is the largest possible matching in G. A graph may contain many maximum matchings. If G only contains a single maximum matching it is known as the unique maximum matching. The cardinality of a maximum matching is written as $|M| = \beta(G)$.*

During decomposition the graph will be partitioned into subgraphs each of size r. This partitioning is enabled through *vertex separators* as defined below.

Definition 2.3 (Vertex Separator) *For a graph $G = (V, E)$ a vertex separator is a subset of vertices $S \subset V$ such that the removal of S from V would lead to the creation of two disjoint graphs G_1 and G_2 such that no edges in E connect vertices in G_1 with vertices in G_2.*

The parameter r governs the size of graph partitions in what is known as an r-division of a graph.

Definition 2.4 (r-division) *An r-division of a graph is the process of partitioning the graph into $\frac{n}{r}$ components with a maximum of r vertices in each component. In this context an r-division is created by the repeated application of the planar separator theorem onto a graph and the resulting subgraphs until each subgraph contains no more than r vertices.*

3 Graph Modifications

The maximum degree of graphs is bound to 3, this reduction is detailed in section 3.1 and will be referred to as *degree reduction*. The k-stars and k-double-stars will also be removed from graphs, this reduction is detailed in section 3.2 and will be referred to as *star reduction*.

3.1 Vertex Splitting

Vertex splitting [17, 13] is a technique through which the degree of a vertex is reduced by splitting it into a chain of smaller degree vertices. Using this technique the maximum degree of vertices in a graph may be bound. The problem then becomes one of finding maximum matchings in the modified graph of bounded degree. This reduction provides an efficient approach to bounding the run-time of the matching algorithm. The splitting operation, shown in figure 3.1, is repeatedly applied to a vertex until its degree is equal to 3. Degree reduction may be carried out in $O(n)$ time and adds $O(n)$ vertices as shown by theorem 1.

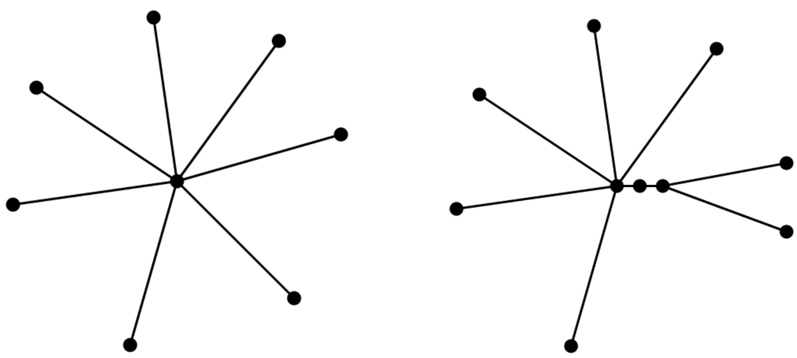


Figure 3.1: *A high degree vertex having its degree reduced. The vertex is replaced by 3 vertices creating a chain with two of the original neighbours forming edges with a newly created vertex. This operation reduces the degree of the vertex by 1. This figure is an adaptation of figures by [17, 13].*

Theorem 1 *Vertex splitting may be carried out in $O(n)$ and adds $O(n)$ new vertices where $n = |V|$ in $G = (V, E)$ [13].*

Proof 1 *Consider $W = \{v \in V | deg(v) > 3\}$. Figure 3.1 shows that splitting reduces a vertex' degree by 1, so $deg(w_i) - 3$ splits are required for each $w_i \in W$. The time complexity is then $O(|E|)$. Since the graph is planar time complexity is $O(n)$. Each split adds 2 vertices, therefore $O(|E|))$ vertices are added and using planarity as before, $O(n)$ vertices in the worst case.*

Figure 3.1 illustrates some important qualities in the graph $G = (V, E)$ after degree reduction occurs.

Definition 3.1 (Vertex Splitting Properties)

1. *Each time a vertex $w \in W$ is split, two vertices are added to the graph giving new vertices $v_1, ..., v_n$ leaving us with the set $N = W \cup \{v_1, ..., v_n\}$.*
2. *For the set of v's original neighbours $U = \{u | (u, v) \in E\}$, each vertex in U will be adjacent to exactly **one** vertex in N, so $deg(u_i, N) = 1 \; \forall u_i \in U$.*
3. *Vertices in N will be adjacent to **at most** two vertices in U, so $deg(v_i, U) \leq 2 \; \forall v_i \in N$.*
4. *Vertices in N will be adjacent to **at least** one other vertex in N, so $deg(v_i, N) \geq 1 \; \forall v_i \in N$.*

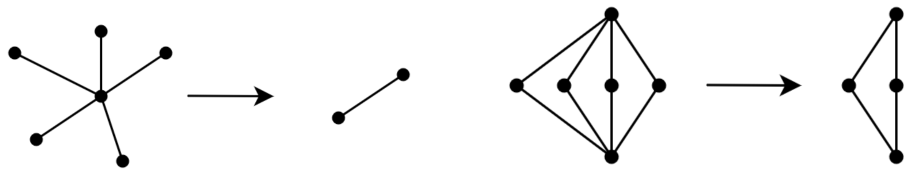
Maximum matchings in a graph should be the same size after degree reduction. This can be ensured by grouping reduced high degree vertices with their corresponding newly added vertices. With this grouping ensured, when one vertex in the group is matched then all vertices in the given group are marked as matched. Vertices within a grouping are also not allowed to be matched with one another as these edges are not representative of the original graph, however edges between groupings will represent the graph's original edges and will not change the size of the matching due to the grouping of vertices.

3.2 k-Star and k-Double-Star Removal

The star reduction has been proposed in order to ensure that $\beta(G) = \Omega(|V|)$ [5], or alternatively that $\beta(G) \geq c \cdot |V|$ for some $c \leq 1$, and is concerned with the absence of specific subgraphs, particularly the 2-stars and 3-double-stars which are defined in definitions 3.2 and 3.3. Using this reduction we can ensure a suitable approximate guarantee on the size of a maximum matching found later on.

Definition 3.2 (k-Star) *A graph $G = (V, E)$ contains a k-star if, for some vertex $v \in V$ and a set of vertices $\{v_1, ..., v_k\} \in V$, $(v, v_i) \in E$ and $\deg(v_i) = 1$ for all $i \leq k$.*

Definition 3.3 (k-Double-Star) *A graph $G = (V, E)$ contains a k-double-star if, for some pair of vertices $u_1, u_2 \in V$ and a set of vertices $\{v_1, ..., v_k\} \in V$, $(u_1, v_i) \in E$ and $(u_2, v_i) \in E$ and $deg(v_i) = 2$ for all $i \leq k$.*



(a) *k-Star with $k \geq 2$ being reduced.* ***(b)*** *k-Double-star with $k \geq 3$ being reduced.*

Figure 3.2: *Examples of both a 2-star reduction and a 3-double-star reduction.*

This reduction has also been employed in order to solve other graph and matching problems [4] and an efficient $O(n)$ algorithm has been developed and is shown in algorithm 1. This process is further illustrated by figure 3.2 and is enabled by the deletion of certain vertices.

```
Data: An undirected graph G = (V, E)
Result: The removal of k-stars and k-double-stars from the graph
begin
    for each v ∈ V do
        if deg(v) = 1 then
            u ⟵ v's neighbour
            t ⟵ (u, u)
            pass token t to u
            if u has more than one token t then
                G.remove(v)
        else if deg(v) = 2 then
            u1, u2 ⟵ v's neighbours
            t ⟵ (u1, u2)
            pass token t to both u1 and u2
            if u1 and u2 both have more than two tokens t then
                G.remove(v)
    end
end
```

Algorithm 1: k- and k-double- star removal algorithm [4]

3.3 Order of Reductions

It is important to assess in which order reductions should be carried out as one may affect the other. The aim is to maintain a maximum degree of 3 and for the graph to contain no k-stars or k-double-stars. It is clear that star reduction will not affect a prior degree reduction, as star reduction removes vertices and edges and therefore will not increase the degree of any vertex. Star reduction may also occur before degree reduction as degree reduction will not introduce k-stars or k-double-stars into the graph, this is illustrated by theorem 2.

Theorem 2 *Degree reduction cannot introduce 2-stars or 3-double-stars into a graph $G = (V, E)$ after star reduction has been carried out.*

Proof 2 *Consider $v \in V$ of high degree and the properties outlined in definition 3.1. At most one vertex in U can have degree one otherwise v contained a 2-star. $deg(u_i, V) \leq 2\ \forall u_i \in U$ since the maximum degree is 3 and $deg(u_i, N) = 1$. Consider $v_1 \in N$ with neighbours $u_1, u_2 \in U$, both sharing common neighbours $w_1, w_2 \in V$ of degree 2, then $deg(u_1) = 3$ and $deg(u_2) = 3$. It is impossible for u_1, u_2 to form a 3-double-star as their third neighbour v_1 has $deg(v_1) = 3$ because $deg(v_i, N) \geq 1$, and for u_1 and u_2 to have another common neighbour of degree 2 implies that the maximum degree in the graph is greater than 3.*

While the order of reductions has been shown to be irrelevant, performing the star reduction initially is motivated by a smaller resulting graph size and this is shown in figure 3.3.

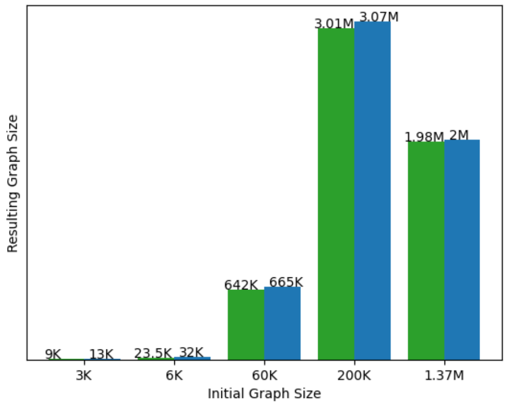


Figure 3.3: *Sizes of graphs before and after applying reductions. Green bars indicate an initial star reduction and blue bars show an initial degree reduction.*

3.4 r-Division of a Planar Graph

A graph decomposition is the modification of a graph in some way and in this context the graph will be partitioned into regions of size r through a process known as r-division [7]. This graph decomposition uses the planar separator theorem to separate the graph, and subsequent regions after separation, into regions with a maximum of r vertices in each. The planar separator theorem [10] provides an approach to partitioning a graph into three subgraphs A, B, and C, such that no edges connect vertices in A to vertices in B and no region is larger than $\frac{2}{3} \cdot n$ where n is the number of vertices in the graph or subgraph being partitioned. The subgraph C may be referred to as a *separator* as in definition 2.3 and an example vertex separator is shown in figure 3.4.

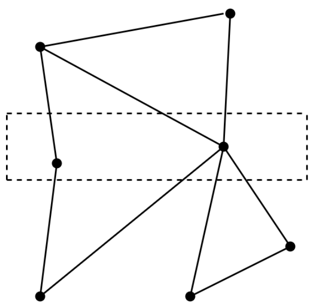


Figure 3.4: *An example separation showing the separator C outlined by a dashed box.*

In the r-division of a graph, two subgraphs G_1 and G_2 will be created such that the vertex set of G_1 and G_2 are $V_1 = A \cup C$ and $V_2 = B \cup C$ respectively. The vertices in the set C are then referred to as *boundary vertices* as they form the boundary between the regions of the partitioned graph. Edges between V_1 and V_2 will be removed leaving two disjoint subgraphs of the original graph G. While some subgraph has more than r vertices, this process will be repeatedly applied to that particular component and any resulting components with more than r vertices, leaving $\frac{|V|}{r}$ individual components. The maximum matching may then be found for each of these components and the matchings then combined in order to form an approximate maximum matching for the original graph G.

4 Matching Algorithm

The matching algorithm employed in this project is Edmonds' blossom algorithm [6]. This approach is based on finding augmenting paths which allow the algorithm to increase the size of the maximum matching by 1.

Definition 4.1 (Alternating and Augmenting Paths) *For a given matching M in G, an alternating path is a path consisting of edges in and not in M alternately. An augmenting path is an alternating path beginning and ending at exposed vertices, this means it must contain $2 \cdot k + 1$ edges, k of which belong to M.*

Given an augmenting path P it is possible to increase the size of a matching $|M|$ by 1 through "swapping" the matched edges with the unmatched edges, meaning the $k + 1$ unmtached edges become part of the matching and the k previously matched edges become unmatched, this process is known as *augmenting* along a path. It is known from Berge's theorem [1] that a matching is maximum if and only if an augmenting path cannot be found in a given graph, as this implies that the size of the matching may not be increased. The blossom algorithm is named as such as the algorithm is centred around contracting subgraphs, called *blossoms*, during the search for augmenting paths.

Definition 4.2 (Blossom) *Given a graph $G = (V, E)$, a blossom in G is a subgraph B that is a cycle of odd length such that k edges of the blossom belong to the matching and $k + 1$ edges of the blossom do not, and where an alternating path of even length leads to a vertex in the blossom, known as the base of the blossom.*

During the search for an augmenting path, if a blossom is discovered it will be contracted in order to find a path through the blossom such that the path remains alternating as it passes through the blossom. A blossom is discovered when two adjacent vertices on the blossom are found with the same root, that being the first vertex of the even-length alternating path leading into the blossom. Vertices on the cycle will be contracted into the base of the blossom, and vertices adjacent to contracted vertices will become adjacent to the base. Once a new augmenting path is found in the contracted graph, the blossom is expanded and the path is reconstructed through the blossom such that the path remains alternating and may be augmented such that no two matched edges will be adjacent.

5 Conclusion

Unfortunately it was not possible to carry out any experimentation during this project due to the nature of the graph separation library[1] used and the limited time remaining, although all code has been implemented for each reduction and is available. The graph separation library posed several problems due to the available graphs, as the separator expects graphs to be planar and not near-planar. While this is not an issue where planar graphs are available, it seems that the degree reduction may break planarity in the way it chooses vertices to add to chains when reducing a high-degree vertex. This is the primary limitation of this project and given more time it may be possible to modify or find an alternative approach to graph partitioning, however this task becomes a project in itself due to the complexity of graph separation algorithms. Another solution to this problem may be to implement the degree reduction process in such a way that planarity is maintained in the resulting graph, however this also seems to be a complex task. The project was also limited in the number of graph modifications and only a single matching algorithm. With more time available it may be desirable to compare the speed improvements among several different matching algorithms when applying selected reductions, and different combinations of reductions may be employed in order to assess the speed improvements of matching algorithms. Again this task seems quite time consuming as it would be necessary to prove the irrelevance of reduction order or alternatively an optimal order of reductions for each group of reductions to be employed. Given access to road network graphs it would have been nice to apply the reduction process to specialised types of planar or near-planar graphs such as road networks, as there exist specialised algorithms for these types of graphs, such as separators [15] and matching algorithms [3], that provide very low running times, however the availability of quality road network graphs seems to be quite limited.

[1] The graph separation library may be found at https://github.com/jeffythedragonslayer/lipton-tarjan

5 Bibliography

[1] Berge, C. Two theorems in graph theory. *Proceedings of the National Academy of Sciences 43*, 9 (1957), 842–844.

[2] Blum, N. A new approach to maximum matching in general graphs. In *International Colloquium on Automata, Languages, and Programming* (1990), Springer, pp. 586–597.

[3] Boyaci, B., Dang, T. H., and Letchford, A. N. On matchings, t-joins, and arc routing in road networks. *Networks 79*, 1 (2022), 20–31.

[4] Chang, Y.-J., and Su, H.-H. Narrowing the local-congest gaps in sparse networks via expander decompositions. *CoRR* (2022), 15–16.

[5] Czygrinow, A., Hańćkowiak, M., and Szymańska, E. Distributed approximation algorithms for planar graphs. In *Italian Conference on Algorithms and Complexity* (2006), Springer, pp. 296–307.

[6] Edmonds, J. Paths, trees, and flowers. *Canadian Journal of mathematics 17* (1965), 449–467.

[7] Federickson, G. N. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM Journal on computing 16*, 6 (1987), 1004–1022.

[8] Friedman, M., and Goob. *History of Folding in Mathematics*. Springer, 2018.

[9] Gabow, H. N., and Tarjan, R. E. Faster scaling algorithms for general graph matching problems. *Journal of the ACM (JACM) 38*, 4 (1991), 815–853.

[10] Lipton, R. J., and Tarjan, R. E. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics 36*, 2 (1979), 177–189.

[11] Micali, S., and Vazirani, V. V. An o (vl vl cl el) algoithm for finding maximum matching in general graphs. In *21st Annual Symposium on Foundations of Computer Science (sfcs 1980)* (1980), IEEE, pp. 17–27.

[12] Mucha, M., and Sankowski, P. Maximum matchings via gaussian elimination. In *45th Annual IEEE Symposium on Foundations of Computer Science* (2004), IEEE, pp. 248–255.

[13] Mucha, M., and Sankowski, P. Maximum matchings in planar graphs via gaussian elimination. *Algorithmica 45*, 1 (2006), 3–20.

[14] Norman, R. Z., and Rabin, M. O. An algorithm for a minimum cover of a graph. *Proceedings of the American Mathematical Society 10*, 2 (1959), 315–319.

[15] Schild, A., and Sommer, C. On balanced separators in road networks. In *14th International Symposium on Experimental Algorithms (SEA)* (2015), pp. 286–297.

[16] Ungar, P. A theorem on planar graphs. *Journal of the London Mathematical Society 1*, 4 (1951), 256–262.

[17] Wilson, D. B. Determinant algorithms for random planar structures. In *SODA* (1997), vol. 97, Citeseer, pp. 258–267.