Local Optimization Algorithms for Maximum Planar Subgraph

Gruia Călinescu
Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA

Sumedha Uniyal
Department of Computer Science, Aalto University, Finland

Abstract

Consider the NP-hard problem of, given a simple graph G, to find a planar subgraph of G with the maximum number of edges. This is called the Maximum Planar Subgraph problem and the best known approximation is 4/9 and is obtained by sophisticated Graphic Matroid Parity algorithms. Here we show that applying a local optimization phase to the output of this known algorithm improves this approximation ratio by a small $\epsilon = 1/747 > 0$. This is the first improvement in approximation ratio in more than a quarter century. The analysis relies on a more refined extremal bound on the Lovász cactus number in planar graphs, compared to the earlier (tight) bound of [5, 8].

A second local optimization algorithm achieves a tight ratio of 5/12 for Maximum Planar Subgraph without using Graphic Matroid Parity. We also show that applying a greedy algorithm before this second optimization algorithm improves its ratio to at least $91/216 < 4/9$. The motivation for not using Graphic Matroid Parity is that it requires sophisticated algorithms that are not considered practical by previous work. The best previously published [7] approximation ratio without Graphic Matroid Parity is $13/33 < 5/12$.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases planar graph, maximum subgraph, approximation algorithm, matroid parity, local optimization

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.38

Related Version *Full Version*: https://hal.science/hal-04629835

Funding Gruia Călinescu's work performed in part while visiting Northwestern University. Sumedha's work performed in part at Aalto University, Finland, and in part at Northwestern University, United States of America. She was supported by the Academy of Finland (grant agreement number 314284).
Sumedha Uniyal: Sumedha Uniyal passed away in February 2020. Her husband, Prajna Uniyal, email uniyal.prajna@gmail.com, has approved this submission.

Acknowledgements Gruia thanks Michael Pelsmajer for many comments that improved the presentation. Gruia thanks the anonymous SODA 2024 reviewer B who suggested using a positive linear combination (instead of case analysis) when computing the ϵ-improvement of Algorithm MTLK4, which lead to a slightly simpler proof with a bigger ϵ. Gruia also thanks Konstantin Makarychev for hints on finding the best linear combination.

1 Introduction

Maximum Planar Subgraph (MPS) is the following problem: given a graph G, find a planar subgraph of G with the maximum number of edges. MPS is known to be NP-complete [24] and APX-hard [5], meaning that there exists a small $\epsilon > 0$ such that a polynomial-time algorithm with approximation ratio of at least $1 - \epsilon$ for MPS implies that $P = NP$. In this paper all graphs are undirected, nonempty, finite, simple graphs unless otherwise noted.

Besides theoretical appeal, MPS has applications in circuit layout, facility layout, and graph drawing (see [17, 31, 27, 23, 14, 12, 21] and the references contained there). Poranen [27] uses simulated annealing and interestingly reports that using the output of better approximation algorithms as the start of simulated annealing obtains better practical results. Chimani, Klein, and Wiedera [13] also report benefits of using better approximation algorithms for MPS.

We assume that the reader has basic knowledge of planar graphs and approximation algorithms. For a quarter century, the best known approximation algorithm for MPS has a tight approximation ratio[1]of 4/9 [5] and uses polynomial-time algorithms for Graphic Matroid Parity[2] to construct a maximum triangular cactus (a graph all of whose blocks[3]are triangles[4]), followed by adding edges to connect the components of this cactus without creating any new cycles. Graphic Matroid Parity has polynomial-time algorithms [26, 25, 9, 18, 19] and allows us to find a triangular cactus with maximum number of triangles.

In this paper we show that applying a local optimization phase after finding a maximum triangular cactus achieves an approximation ratio of $(4/9) + \epsilon$, for a small $\epsilon > 0$. A second local optimization algorithm achieves a tight ratio of 5/12 for MPS without using Graphic Matroid Parity. We also show that applying a "greedy" algorithm before this second local optimization algorithm improves its ratio to at least 91/216. This variant has an upper limit of $3/7 < 4/9$ on its approximation ratio. The motivation for not using Graphic Matroid Parity is that it requires sophisticated algorithms that are not considered practical by Chimani and Wiedera [14][5]. The best previously published [7] approximation ratio without Graphic Matroid Parity is $13/33 < 5/12$.

Besides these theoretical guarantees, both of our local optimizations "make sense" (since each local improvement step increases the size of the output by one) and can be used in heuristics, or as a starting point of simulated annealing. Other ideas that make sense are further discussed in Conclusions (Section 5).

Previous Theoretical Work. A planar graph on n vertices has at most $3n - 6$ edges. As we can assume that the input graph is connected (or else we can run an algorithm separately on the connected components of the input), simply outputting a spanning tree achieves a tight approximation ratio of 1/3. Dyer, Foulds, and Frieze [16] proved that the Maximal Planar Subgraph, which simply outputs an inclusion-maximal planar subgraph, has performance ratio 1/3. Cimikowski [15] proved that the heuristics of Chiba, Nishioka, and Shirakawa [10] and of Cai, Han, and Tarjan [2] have performance ratios not exceeding 1/3.

Călinescu, Fernandes, Finkler, and Karloff [5] were the first to improve this. Their first algorithm (GT, from Greedy-Triangles) (greedily) adds triangles, as long as the graph stays a triangular cactus. Then the algorithm greedily adds edges connecting the components of the cactus, without creating any new cycles. This "connecting" phase appears at the end of all the approximation algorithms and we will not mention it from now on. The GT algorithm has a tight approximation ratio of 7/18. Their second algorithm (MT, from Maximum-Triangular Cactus) employs exact algorithms for Graphic Matroid Parity (i.e., [25]) to construct a triangular cactus with a maximum number of triangles. This algorithm achieves a tight approximation of 4/9, and the analysis is long and complicated. Another claim of the 4/9 ratio was sent in a personal communication in 1996 or 1997 by Danny Raz, and is also proven in Chalermsook, Schmid, and Uniyal [8][6]; these proofs are also long and/or complicated.

1. *tight* in this paper means that there are matching examples for the approximation ratio of the algorithm
2. See Section 2 for the exact definition
3. We use the standards definition of *block*, explicitly stated in Section 2
4. A simple graph is a *cactus* if any two distinct simple cycles have at most one vertex in common. A triangular cactus as defined by us is a cactus.
5. Also, we could not find any implementation.
6. [8] also has the sentence "Therefore, combining this with our bound implies that local search arguments are sufficient to get us to a $\frac{4}{9} + \epsilon$ approximation for MPS", but no proof for this sentence; their previous arguments give a $\frac{4}{9} - \epsilon$ approximation.

Poranen [28] proposes two new approximation algorithms but their ratio is below 4/9 [11]. Chalermsook and Schmid [7] obtain a 13/33 ratio without using Graphic Matroid Parity. Note that $13/33 > 7/18$. We have examples showing that this algorithm has ratio at most $11/27 < 5/12$. See Figure 1 for a graphical representation of approximation ratios.

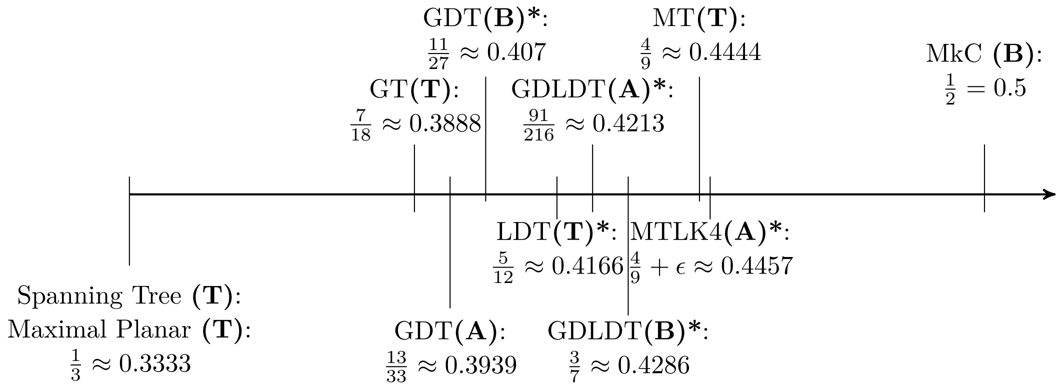

Figure 1 Ratios for various approaches. **(T)** says that the result is tight with matching example. **(A)** says that an algorithm is proven to achieve that ratio. **(B)** states an upper bound for the approximation ratio of an algorithm. New results marked by *.

Techniques discussion. A graph A is a kt-structure if every block in every non-trivial connected component of A is either a K_4 or a triangle. A diamond is the graph resulting from deleting any single edge from a K_4. A graph A is a dt-structure if every block in every non-trivial connected component of A is either a diamond or a triangle. A graph A is a 4-structure if every block in A has at most 4 vertices. All 4-structures are planar graphs. See Figure 2 for illustrations.

4-Structures can have more edges than triangular cacti, but finding a 4-structure with maximum number of edges is known to be NP-hard [4] (easy reduction from 3D-Matching [20]). It is known [3] that even if we fix k and allow blocks of size up to k in our output, and could find the k-structure with maximum number of edges and planar blocks (algorithm MkC), we would still get an approximation ratio smaller than 1/2.

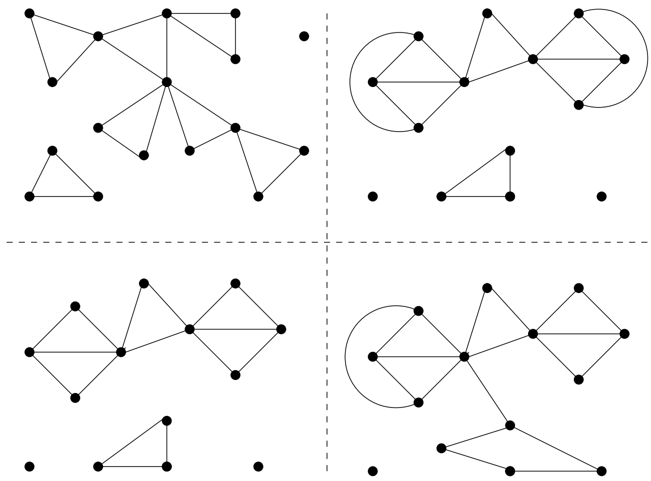

Figure 2 In the upper left, a triangular cactus. In the upper right, a kt-structure. In the lower left, a dt-structure. In the lower right, a 4-structure.

Local optimization is a powerful technique for unweighted maximization problems. For example, Lee, Sviridenko, and Vondrák [22] prove that a basic local optimization algorithm achieves a $(1-\epsilon)$-approximation for Matroid Parity, a generalization of Graphic Matroid Parity that is NP-hard [29]. Using local optimization for MPS was suggested by [8].

Our first algorithm, denoted by MTLK4 (Maximum-Triangular-Local-K4) from now on, starts with a maximum triangular cactus and replaces, as long as we still have a kt-structure, at most two triangles by a K_4. Previous approximation algorithms for MPS have not been able to exploit K_4's in getting a better approximation ratio, and in fact our analysis giving that the approximation ratio of MTLK4 is at least $(4/9)+\epsilon$ is complicated and not tight[7]. The analysis relies on Theorem 5, which gives a more refined extremal bound on the Lovász cactus number in planar graphs, compared to the earlier (tight) bound of [5, 8].

The algorithm of Chalermsook and Schmid [7] (which we call GDT, from Greedy-Diamonds-Triangles) consists of greedily adding diamonds followed by greedily adding triangles (while staying a dt-structure), and has approximation ratio between 13/33 and 11/27. Our 5/12 approximation ratio of our second local optimization algorithm, denoted by LDT (Local-Diamonds-Triangles) from now on, is based on the fact that LDT does "break" diamonds to add more diamonds, assuming a triangle is left from our broken diamond. While this 5/12-approximation is tight, it can be improved by greedily adding diamonds (while staying a dt-structure)[8], followed by LDT. We call this algorithm GDLDT. We prove that GDLDT has an approximation ratio between 91/216 and 3/7.

While our algorithms are fairly simple, their analyses are long and complicated. The reader may want to start with the (7/18)-approximation of [5] (also presented in [4]) for a simple non-trivial algorithm with a somewhat simple analysis.

One can also use local optimization to approximate Graphic Matroid Parity, as described in [22]. As pointed out in [8], this leads to a $4/9-\epsilon$ approximation for MPS, as outlined below. The local optimization algorithm used by [22], adapted to our terminology, replaces sets of k triangles by sets of $k+1$ triangles, as long as it keeps a triangular cactus. We have (so far unpublished, and involving a different set of authors) evidence that swapping two triangles for one gives a ratio of at least 11/27. Theorem 1.1 of [8] also mentions a 2-swap algorithm, but it swaps only triangles of a plane graph and is not analyzed as an MPS algorithm. Let n be the number of vertices in the input graph, and M be the number of triangles in the input graph. The analysis of [22] requires $k = 5^{\lceil 1/(2\epsilon) \rceil}$ to achieve a $(1-\epsilon)$-approximation for Matroid Parity. To achieve a 5/12 approximation for MPS, one would need an $\epsilon = 1/4$ above: if the optimum is triangulated and the maximum triangular cactus has $n/3$ triangles (which we know that it can happen, from [5]), then we need a triangular cactus with almost $n/4$ triangles in the algorithm's cactus to get a ratio of 5/12, which means we must approximate Graphic Matroid Parity with a factor of 3/4. Directly using the analysis of [22] gives an $O(M^{25})$-time algorithm. We do not fully understand [1] and maybe it provides a better analysis of the swapping heuristics and achieves a $O(n^{4+3/\epsilon})$ running time. This would be $O(n^{16})$ time to get a ratio of 5/12 for MPS. By comparison, our LDT algorithm has an implementation that runs in $O(n^5)$ time.

2 Preliminaries

Generally we follow West [32] for terminology and notation. Given a graph $G = (V, E)$ and $V' \subseteq V$, we denote by $G[V']$, the *induced* subgraph of G with vertex set given by V' (we keep in this subgraph with vertex set V' all the edges of G with both endpoints in V').

---

[7] We believe that another page of arguments will make our ϵ a very little bit bigger.

[8] This is exactly the start of the GDT algorithm of [7].

A *triangle* in a graph is a C_3, a cycle of length three. A graph is *planar* if it can be drawn in the 2-dimensional plane so that no two edges meet in a point other than a common end. A triangle of a plane graph is a *facial triangle* if the triangle is the boundary of a face of the plane graph. We call a face of a plane graph a *triangular face* if its boundary is a triangle.

A *cut-vertex* (resp., *bridge*) is a vertex (resp., edge) whose deletion increases the number of components. A graph is said to be a *biconnected graph*, if it has at least three vertices and it does not contain any cut-vertices. A *block* is a maximal connected subgraph without a cut-vertex. Thus, every block is either a maximal biconnected subgraph, or a bridge.

Graphic Matroid Parity is the following problem: given a multigraph $G' = (V', E')$ and a partition of the edge set E' into pairs of distinct edges $\{f, f'\}$, find a (simple) forest $F \subseteq E'$ with the maximum number of edges, such that $f \in F$ if and only if $f' \in F$, for all $f \in E'$.

Graphic Matroid Parity algorithms can be used to construct a maximum triangular cactus in a given graph [25]. Indeed, given a graph $G = (V, E)$, construct G' by having $V' = V$ and for each triangle T of G, let e, e' be any pair of distinct edges in T. Add two new edges f and f' to E', f with the same endpoints as e, and f' with the same endpoints as e'. Pair f with f' in the partition of E'. The straightforward lemma 2.13 of [5] states that a forest F of G' as above with $2p$ edges exists if and only if a triangular cactus with p triangles exists in G.

3 Local Optimization after Max Triangular Cactus

Here is our first approximation algorithm, Algorithm **MTLK4**: Start with a maximum triangular cactus. As long as possible, if there exists a K_4 in the input graph and at most two edge-disjoint triangles in our kt-structure such that removing the (one or two) triangles and adding the K_4 results in a kt-structure, do this replacement. See figures 3 and 4 for illustrations. Each such replacement increases the cyclomatic number[9] of the kt-structure, and thus this is a polynomial-time algorithm. Precisely, there are $O(n^4)$ K_4's to consider and for each we can look at $O(n^2)$ ways to choose two existing triangles for replacement (these two triangles come from a graph with $O(n)$ edges). So we can achieve each replacement in $O(n^7)$ time, for a total running time, excluding the Matroid Parity algorithm, of $O(n^8)$.

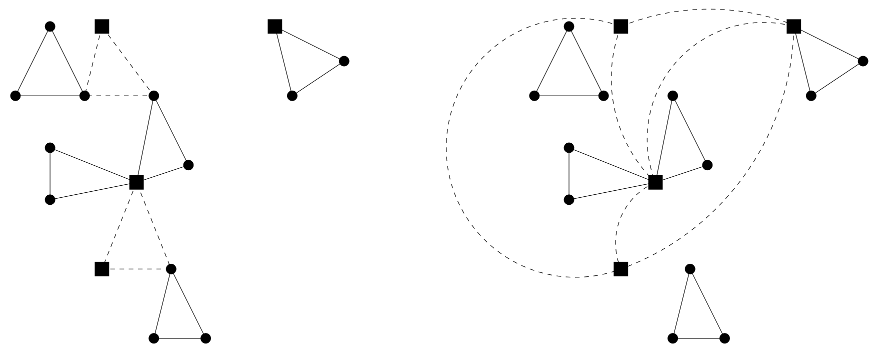

Figure 3 On the left side, the current kt-structure, with two connected components. There is a K_4 in the input graph, with its four vertices represented by small filled squares. Two triangles, dashed, can be removed to disconnect these four vertices, after which the K_4 (with edges represented by dashed cycle arcs) can be be added resulting in another kt-structure (on the right side, with three connected components) with a higher cyclomatic number.

[9] The cyclomatic number (also called circuit rank) of an undirected graph is the minimum number of edges that must be removed from the graph to break all its cycles, making it into a tree or forest.

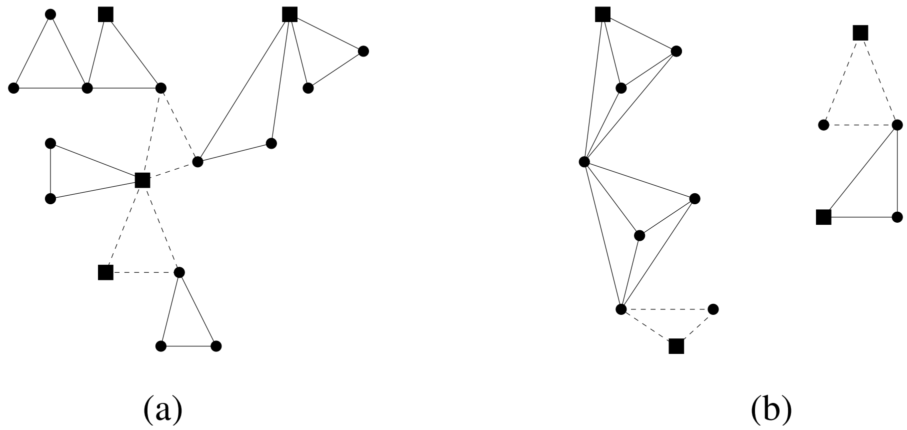

Figure 4 Both (a) and (b) show kt-structures where, for each of them, two triangles (dashed) can be removed to allow a K_4, whose vertices are represented by small filled squares, to be added resulting in a kt-structure.

We believe that an $O(n^5)$-running time version of the algorithm exists, based on the following idea. One needs to read the proof of the approximation ratio first, as we modify the algorithm. We only check for each K_4 if there is no strong bond, and its four vertices are not all in the same component of the current kt-structure. This check can be done in $O(1)$ after computing connected components by Depth First Search (and this has to be done only once after each local improvement). A K_4 that does not pass this check is not used for the local improvement, even though in MTLK4 it could possibly lead to improvement (see Figure 4 (a)). However, we cannot get a $O(n^4)$ running time by showing that a K_4 that was found not useful once will not become useful in the future. We have a counterexample. The rest of the section will be spent on the approximation ratio analysis.

First, some intuition. Let $\beta(\bar{G})$ be the number of triangles in a maximum triangular cactus in graph $\bar{G}$. It is known from [5] (see also Inequality (2) below) that, if $\tilde{H}$ is a planar graph on n vertices with $|E(\tilde{H})| = 3n - 6 - t$, then $\beta(\tilde{H}) \geq (1/3)(n - t - 2)$. With G the input graph and H an optimum solution (a maximum planar subgraph of G), we obtain that the output of the MT algorithm (which, recall, computes the triangular cactus with maximum number of triangles, and then connects the components of this cactus) has $n - 1 + \beta(G)$ edges (since every triangle gains one edge over a spanning tree), and we obtain a ratio of 4/9 from $\beta(G) \geq \beta(H)$. With θ such that H has $3n - 6 - \theta$ edges, from simple algebraic manipulation, we would have an $\epsilon > 0$ improvement over 4/9 if $\theta \geq \epsilon' n$ for some $\epsilon' > 0$. But we cannot count on this happening.

So now we assume that θ is close to 0. As we use K_4's, the second question is what happens if H (the optimum solution as above) does not have any K_4's. The bound $\beta(H) \geq (1/3)(n - \theta - 2)$ is not tight anymore, and in fact with $\theta = 0$ we are able to prove that $\beta(H) \geq (3/7)n - 1$ when H has no K_4's. This is good as $(n + (3/7)n - 2)/(3n - 6) \geq 10/21 > 4/9$.

So we assume that θ is close to 0 and H has K_4's. To quantify the benefits of these K_4's, we introduce some notation (see Figure 5 for an illustration):

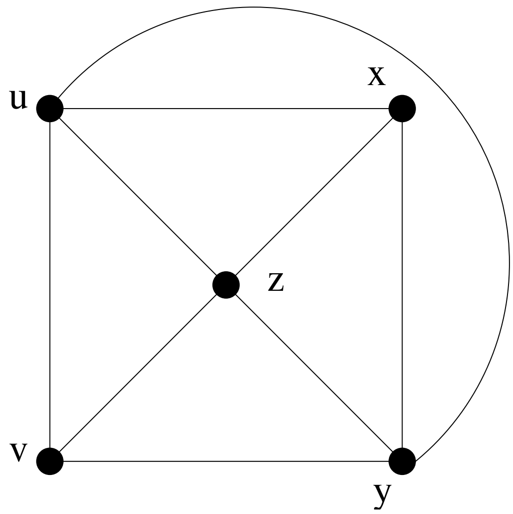

Figure 5 In this triangulated plane graph, x and v are the cubic (degree three) vertices.

▶ **Definition 1.** *Call a vertex v of a triangulated plane graph $\tilde{H}$* cubic *if it has degree three (so we have a K_4 formed by v and its neighbors). Let $c(\tilde{H})$ be the number of cubic vertices in $\tilde{H}$.*

Theorem 5 below obtains that $\beta(\tilde{H}) \geq (3/7)n - (1/7)c(\tilde{H}) - 1$, with $n = |V(\tilde{H})|$. With $\tilde{H}$ a triangulated supergraph of H on the same vertex set, we have that $\tilde{H}$ contains exactly θ edges not in H. Then we can obtain that $\beta(H) \geq (3/7)n - (1/7)c(\tilde{H}) - 1 - \theta$. Now, if $c(\tilde{H})$ is also small in addition to θ being close to 0, we have again a ratio bigger than 4/9. And when $c(\tilde{H})$ is "large" compared to n, and θ is close to 0, then there are enough K_4's in H for the local optimization phase of MTLK4 to use, as shown after Theorem 5. We are done with providing some intuition. As small graphs are easy to handle, we assume that $n = |V(G)| > 4$ in this section. On our way to Theorem 5 we need a lot of notation.

Double-partitions. Let $\bar{G} = (V, E)$ be a graph on n vertices. Let $\mathcal{P} = \{V_1, \ldots, V_k\}$ be a partition of the vertices of $\bar{G}$ into *vertex classes*, and $\mathcal{Q} = \{E_1, \ldots, E_m\}$ be a partition of the edges of $\bar{G}$ into *edge colors*. We say that an edge color and a vertex class are incident if at least one of the edges of the edge color is incident to at least one of the vertices of the vertex class. For $1 \leq i \leq m$, let u_i denote the number of vertex classes V_j of $\mathcal{P}$ incident to edge color E_i of $\mathcal{Q}$. We say that "double-partition" $(\mathcal{P}, \mathcal{Q})$ *covers* a triangle if the triangle has at least two vertices in the same vertex class of $\mathcal{P}$ (when we also say that the triangle is *covered* by this vertex class) or all three edges in the same edge color of $\mathcal{Q}$ (when we also say that the triangle is *covered* by this edge color). We call the double-partition $(\mathcal{P}, \mathcal{Q})$ *valid for $\bar{G}$* if every triangle of $\bar{G}$ is covered. Set[10]:

$$\Phi(\mathcal{P}, \mathcal{Q}) = n - k + \sum_{i=1}^{m} \lfloor \frac{u_i - 1}{2} \rfloor. \quad (1)$$

Since $k \leq n$ and $u_i \geq 1$ for all i, we have that $\Phi(\mathcal{P}, \mathcal{Q}) \geq 0$, and note that there is always a valid double-partition $(\mathcal{P}, \mathcal{Q})$ for $\bar{G}$ (e.g., $k = 1$ and $m = 1$ and $V_1 = V(\bar{G})$ and $E_1 = E(\bar{G})$).

[10] The formula given here, which differs from that in [25] by having floors, is correct. See also [30] or Theorem 11.3.2 of [25].

Call a vertex class a *trivial vertex class* if it has just one vertex, call this vertex a *singleton*, and an edge color a *trivial edge color* if it has just one edge. The *contribution* of an edge color E_i of $\mathcal{Q}$ to $\Phi(\mathcal{P}, \mathcal{Q})$ is the quantity $\lfloor \frac{u_i - 1}{2} \rfloor$. The *contribution* of $\mathcal{P}$ to Φ is defined to be $n - k$. According to Lovász and Plummer [25], we have:

▶ **Proposition 2** (Theorem 11.3.6 in [25]). *The number of triangles in a maximum triangular cactus in a graph $\bar{G}$ is equal to the minimum of $\Phi(\mathcal{P}, \mathcal{Q})$ taken over all valid double-partitions $(\mathcal{P}, \mathcal{Q})$ for $\bar{G}$.*

Let $\tilde{H}$ be a planar graph with $n \geq 3$ vertices. Embed $\tilde{H}$ in the plane without crossing edges, obtaining a plane graph. Let t be the number of edges *missing* for this embedding to be triangulated (meaning that adding t edges to $\tilde{H}$ would result in a triangulated planar graph). A triangulated plane graph has $3n - 6$ edges, if $n \geq 3$. So $t = (3n - 6) - |E(\tilde{H})|$; t does not depend on the embedding. The 4/9-approximation of [5] is based on:

▶ **Proposition 3** (Theorem 2.3 of [5], also equivalent to Corollary 1.2 of [8]). *Let $\tilde{H}$ be a connected planar graph with $n \geq 3$ vertices. Let t be the number of missing edges, defined as above. Then*

$$\Phi(\mathcal{P}, \mathcal{Q}) \geq \frac{1}{3}(n - 2 - t),$$

for all valid double-partitions $(\mathcal{P}, \mathcal{Q})$ for $\tilde{H}$.

This bound is known to be asymptotically tight when $t = 0$, see for example [5] where their 4/9-approximation is proven tight. Recall that $\beta(\tilde{H})$ is the number of triangles in a maximum triangular cactus in $\tilde{H}$. From Propositions 2 and 3 we obtain:

$$\beta(\tilde{H}) \geq \frac{1}{3}(n - 2 - t), \tag{2}$$

If a double-partition $(\mathcal{P}, \mathcal{Q})$ is valid for plane graph $\tilde{H}$, then it must cover all the facial triangles of $\tilde{H}$.

▶ **Definition 4.** *Call a double-partition $(\mathcal{P}, \mathcal{Q})$ of a plane graph $\tilde{H}$* p-valid *if it covers the facial triangles of $\tilde{H}$.*

Note that a valid double-partition $(\mathcal{P}, \mathcal{Q})$ of a graph $\bar{G}$ is p-valid for any plane embedding of $\bar{G}$. The main technical result of this section is Theorem 5 below, which gives a lower bound for $\Phi(\mathcal{P}, \mathcal{Q})$ for all p-valid double partitions $(\mathcal{P}, \mathcal{Q})$ of a triangulated plane graph $\tilde{H}$ in terms of n and $c(\tilde{H})$. The accounting part of the lower bound is proven assuming that the double partition $(\mathcal{P}, \mathcal{Q})$ satisfies a number of conditions. Ideally, we would want that $\mathcal{P}$ would have only one vertex class P_1 with more than one vertex (we are not able to ensure this condition); this condition would make the accounting much simpler.

We will enumerate the actual conditions later. To get that $(\mathcal{P}, \mathcal{Q})$ satisfy these actual conditions, we *modify* $(\mathcal{P}, \mathcal{Q})$, which really means obtaining from a p-valid $(\mathcal{P}, \mathcal{Q})$ another double partition $(\mathcal{P}', \mathcal{Q}')$ such that $\Phi(\mathcal{P}, \mathcal{Q}) \geq \Phi(\mathcal{P}', \mathcal{Q}')$ and such that $(\mathcal{P}', \mathcal{Q}')$ is also p-valid for $\tilde{H}$. It is fine to make such modifications since, once we prove a lower bound for $(\mathcal{P}', \mathcal{Q}')$, then it holds for $(\mathcal{P}, \mathcal{Q})$ as well.

Assuming that $\tilde{H}$ is a triangulated plane graph, one can easily verify based on the definition of $\Phi(\mathcal{P}, \mathcal{Q})$ (Equation (1)) that we can do the following **modifications**:

1. Any edge of $\tilde{H}$ such that both facial triangles that contain this edge are covered by vertex classes is put into an edge color by itself (we can do this since this trivial edge color contributes 0 to Φ, and the contribution of any other edge color does not increase). In particular, any edge that is incident to two vertices in the same vertex class is put into an edge color by itself.
2. If we have two distinct edge colors both incident to the same two (or more) distinct vertex classes, these two edge colors can be merged into one edge color.
3. If an edge color Q_i can be partitioned into two edge colors Q' and Q'' such that all the facial triangles covered by Q_i are covered by Q' or Q'', and Q' and Q'' are incident to at most one common neighbor among the vertex classes, split Q_i into Q' and Q''.

Recall Definition 1. One main technical result is:

▶ **Theorem 5.** *Let $\tilde{H}$ be a triangulated plane graph with $n \geq 3$ vertices, and let $(\mathcal{P}, \mathcal{Q})$ be a double-partition p-valid for $\tilde{H}$. Then:*

$$\Phi(\mathcal{P}, \mathcal{Q}) \geq \frac{3}{7}n - \frac{1}{7}c(\tilde{H}) - 1. \tag{3}$$

Due to space limitations, we only give some intuition for the proof, which appears in the full version of the paper. Equation (3) is tight (excluding a small additive term), as shown by the following two constructions. Both of them provide useful examples for the intuition behind the proof. Consider a triangulation W on r vertices containing no K_4. From Euler's formula we get that W has $2r - 4$ faces. In the first construction, insert one cubic vertex in every face. We have $n = r + (2r - 4)$ and $c = 2r - 4$. We take $\mathcal{P}$ to have only one non-trivial vertex class, W, and $\mathcal{Q}$ to have only trivial edge colors. Thus $\Phi(\mathcal{P}, \mathcal{Q}) = r - 1 = (3/7)n - (1/7)c - (1/7)$.

In the second construction, insert three new vertices into every face of W as in Figure 6. We have $n = r + 3(2r - 4) = 7r - 12$ and $c = 0$. We take $\mathcal{P}$ to have only one non-trivial vertex class, W, and $\mathcal{Q}$ to have $2r - 4$ non-trivial edge colors, each with the edges embedded strictly in each face of W. Each of these non-trivial edge colors is incident to exactly 4 vertex classes: the three singletons (singletons are vertices) inside the face of W, and the non-trivial class comprised of W. Thus $\Phi(\mathcal{P}, \mathcal{Q}) = r - 1 + 2r - 4 = (3/7)n + (1/7)$.

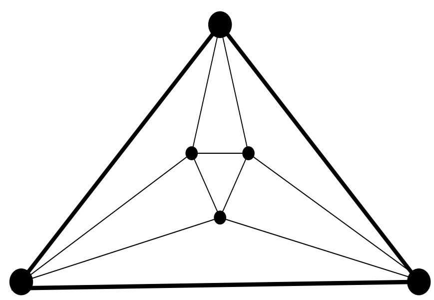

Figure 6 A face of W is shown in solid edges.

Intuition. To show some of the proof ideas, we make the simplifying assumption that $\mathcal{P}$ has only one non-trivial part, which we call R (we could not find a way to enforce this assumption and ended up with a much longer proof). We also make the simplifying assumptions that $r := |R| > 2$ and that $\tilde{H}[R]$ is a connected graph, even though the arguments below can be adapted without these two assumptions. First, apply Modification 1 such that any edge of $\tilde{H}[R]$ is put into an edge color by itself.

For a face F of $\tilde{H}[R]$, call S_F the set of singletons embedded strictly inside F. We discuss now the case $|S_F| > 1$. Due to $\tilde{H}$ being triangulated, $\tilde{H}[S_F]$ is connected. Walking around $s \in S_F$ in clockwise manner, we get a circular list L of vertices that are not all in the same vertex class. When we switch in L from one class to another, we have one facial triangle T_1 of $\tilde{H}$ that is not covered by a vertex class. Later in the walk, when again we switch in L from one vertex class to another, we have another facial triangle T_2 of $\tilde{H}$ that is not covered by a vertex class. If there are two distinct edge colors Q_1 and Q_2 of $\mathcal{Q}$ covering T_1 and T_2, then there are two distinct vertex classes incident to both Q_1 and Q_2 and we can apply Modification 2. Continue this traversal of L and get that all the facial triangles incident to s that are not covered by R are covered by one single edge color. $\tilde{H}[S_F]$ is connected, and we can do a depth-first search traversal and as we meet the vertices of S_F, we do the walk-around described above for each of them to obtain that all the facial triangles of $\tilde{H}$ embedded in F that are not covered by R are covered by one single edge color which we call Q_F. By construction, Q_F meets all the singletons of S_F and the vertex class R.

If edge color Q_F as above also covers facial triangles outside F, then we can apply Modification 3, where Q' consist of the edges of Q_F embedded strictly inside F and Q'' consists of the other edges of Q_F; note that only R among the vertex classes can be incident to both Q' and Q''. After doing this for all the faces of $\tilde{H}[R]$, we get that any non-trivial edge color is the Q_F for some face F with $|S_F| > 1$ of $\tilde{H}[R]$, and all the edges of Q_F are embedded strictly inside F. Note that when $|S_F| = 0$ or $|S_F| = 1$, all the triangular faces of $\tilde{H}$ embedded in F are covered by R. And all the triangular faces of $\tilde{H}$ are embedded in some face F of $\tilde{H}[R]$. We now have:

$$\Phi(\mathcal{P},\mathcal{Q}) = (r-1) + \sum_{F \text{ face of } \tilde{H}[R]} \lfloor |S_F|/2 \rfloor = (r-1) + \sum_{i \geq 0} f_i \lfloor \frac{i}{2} \rfloor, \tag{4}$$

where f_i is the number of faces of $\tilde{H}[R]$ each with i singletons embedded inside. Let c_1 be the number of triangular faces of $\tilde{H}[R]$ with exactly one singleton embedded inside. As any such singleton is a cubic vertex,

$$c_1 \leq c. \tag{5}$$

Let d_1 be the number of non-triangular faces of $\tilde{H}[R]$ with one singleton inside, so that $f_1 = c_1 + d_1$. We would have $2r - 4$ triangular faces of $\tilde{H}[R]$ if it were triangulated. A non-triangulated face can be replaced by two or more triangulated faces by adding "fake" edges, and thus from which we deduce:

$$2r \geq c_1 + 2d_1 + f_3 + f_5, \tag{6}$$

and therefore

$$4r \geq 2c_1 + 4d_1 + 2f_3 + 2f_5 \geq 2c_1 + 3d_1 + 2f_3 + f_5. \tag{7}$$

Using the equation above and Equation (5) we obtain:

$$\frac{4}{7}r \geq \frac{2}{7}c_1 + \frac{3}{7}d_1 + \frac{2}{7}f_3 + \frac{1}{7}f_5 = \frac{3}{7}f_1 + \frac{2}{7}f_3 + \frac{1}{7}f_5 - \frac{1}{7}c_1 \geq \frac{3}{7}f_1 + \frac{2}{7}f_3 + \frac{1}{7}f_5 - \frac{1}{7}c.$$

For $i \in \{0, 2, 4, 6, 7, 8, 9, \ldots\}$, we have $\lfloor i/2 \rfloor \geq \frac{3}{7}i$. Using this, the equation above, Equation (4), and $n = |V(\tilde{H})| = r + \sum_{i \geq 0} i \cdot f_i$, we obtain:

$$\Phi(\mathcal{P},\mathcal{Q}) = (r-1) + \sum_{i \geq 0} f_i \lfloor \frac{i}{2} \rfloor \geq \frac{3}{7}(r + \sum_{i \geq 0} i \cdot f_i) - \frac{1}{7}c - 1 = \frac{3}{7}n - \frac{1}{7}c - 1,$$

or Equation (3). We are done providing intuition for the proof of Theorem 5.

We continue the analysis of the approximation ratio of Algorithm MTLK4. As before, G denotes the input graph and H an optimum solution (a maximum planar subgraph of G). Let $\tilde{H}$ be a triangulated plane supergraph of H, and θ be $|E(\tilde{H})| - |E(H)|$. As above, the value of the optimum solution is $|E(H)| = 3n - 6 - \theta$. Below, c counts the number of cubic vertices in $\tilde{H}$.

From Theorem 5 and Proposition 2 we immediately obtain

$$\beta(G) \geq \beta(H) \geq \beta(\tilde{H}) - \theta \geq \frac{3}{7}n - \frac{1}{7}c - \theta - 1 \tag{8}$$

as one edge of $E(\tilde{H}) \setminus E(H)$ can be part of only one triangle of the maximum triangular cactus of $\tilde{H}$.

Let A be the kt-structure produced by MTLK4. Say that there is a *strong bond* between two vertices if there is a path between them in A with all edges in K_4's of A. Call a cubic vertex v of $\tilde{H}$ *blocked* if $K_4(v)$, the K_4 formed by v and its three neighbors in $\tilde{H}$, is not added to the kt-structure. A cubic vertex v is blocked because of one of the following:

- one of the edges of $K_4(v)$ is not in G (also not in H). We call v *absent*.
- all the edges of $K_4(v)$ are in G (and in H - since H is a maximum planar subgraph of G) and at least two of the four vertices of $K_4(v)$ are connected by a strong bond. We call v *neutralized*.
- all the edges of $K_4(v)$ are in G (and in H) and there are no strong bonds between any of the vertices of $K_4(v)$, and all four of the vertices of $K_4(v)$ are in the same connected component of A. We call v *subdued*.

Indeed, if all the edges of $K_4(v)$ are in G (and in H) and there is no strong bond between any two vertices of $K_4(v)$ and only three of its vertices are in the same component of A, then we could remove at most two triangles from A to disconnect these three vertices, and add $K_4(v)$. And if two of the vertices of $K_4(v)$ are in one component of A and the other two in another component of A, then we can remove one triangle from each of these two components and add $K_4(v)$.

We use c_a, c_s, c_n to denote the number of absent, subdued, and neutralized cubics respectively. Let $A'_1, \ldots, A'_{q'}$ be the non-trivial sub-kt-structures obtained from A by removing all the triangles that are not part of K_4's. If we have j'_i K_4's in A'_i, then A'_i has $3j'_i + 1$ vertices, each two of them connected by a strong bond. All the strong bonds are obtained this way. Let γ be the number of K_4's in A. Then $\gamma = \sum_{i=1}^{q'} j'_i$.

Let B'_i be $\tilde{H}[V(A'_i)]$. Then B'_i has at most $3(1 + 3j'_i) - 6 = 9j'_i - 3$ edges. For a cubic v to be neutralized, one of the four edges of $K_4(v)$ must be an edge of some B'_i. Recall than $n > 4$ and that $\tilde{H}$ is a triangulated plane graph. Then cubic vertices cannot be adjacent in $\tilde{H}$ and one can easily check that one edge e of $\tilde{H}$ can participate in at most two $K_4(u)$. Precisely, if one of the endpoints of e is a cubic, then this endpoint is the only possible u above, and if not the only possible u above are the two vertices that each forms a facial triangle with e.

Thus the total number of neutralized cubics is at most

$$c_n \leq 2\sum_{i=1}^{q'}(9j'_i - 3) \leq 18\gamma. \tag{9}$$

Also by the reasoning above,

$$c_a \leq 2\theta. \tag{10}$$

Let α be the number of triangles in A and recall that γ is the number of K_4's in A. Let A have non-trivial components $A_1, \ldots, A_q$, each with α_i triangles and γ_i K_4's. Let $B_i = \tilde{H}[V(A_i)]$. Thus $|V(B_i)| = |V(A_i)| = 1 + 2\alpha_i + 3\gamma_i$. We count how many subdued cubics $c_s(i)$ are in $V(A_i)$. Remove all these cubics from B_i and we get a planar graph D_i with $1+2\alpha_i+3\gamma_i-c_s(i)$ vertices and where each subdued cubic vertex of A_i is the only vertex of $\tilde{H}$ embedded in a triangular face of D_i (recall that $n>4$ and cubic vertices cannot be adjacent in $\tilde{H}$). Since D_i can have at most $2(1+2\alpha_i+3\gamma_i-c_s(i))-4$ triangular faces:

$$c_s(i) \leq 2(1+2\alpha_i+3\gamma_i-c_s(i))-4$$

from which we deduce:

$$c_s \leq \frac{4}{3}\alpha+2\gamma-\frac{2}{3}q.$$

As $\alpha \leq \beta(G)$ (recall that $\beta(G)$ is the number of triangles in a maximum triangular cactus of the graph G), adding the equation above to Equations (9) and (10) we obtain: $c \leq 20\gamma+\frac{4}{3}\beta(G)+2\theta$, which we rewrite as

$$\gamma \geq -\frac{1}{15}\beta(G)-\frac{1}{10}\theta+\frac{1}{20}c. \tag{11}$$

Now we have all the ingredients to obtain our $((4/9)+\epsilon)$-approximation. As before, G denotes the input graph and H an optimum solution (a maximum planar subgraph of G). Let $\tilde{H}$ be a triangulated plane supergraph of H, and θ be $|E(\tilde{H})|-|E(H)|$. As above, the value of the optimum solution is $|E(H)|=3n-6-\theta$. From Inequality (3) for H, we obtain:

$$\beta(G) \geq \beta(H) \geq \frac{1}{3}(n-2-\theta).$$

Multiply the inequality above by 141/166, Inequality (8) by 21/166, and Inequality (11) by 60/166 (we solved by hand a small linear program to obtain these numbers) and add them up to obtain:

$$\beta(G)+\gamma \geq \beta(G)+\frac{60}{166}\gamma \geq \frac{28}{83}n-\frac{37}{83}\theta-1.$$

As every K_4 has at least one more edge compared to the two triangles it replaces, our output has at least $n-1+\beta(G)+\gamma$ edges. Based on the inequality above, one can check:

$$n-1+\beta(G)+\gamma \geq \frac{111}{83}n-\frac{37}{83}\theta-2 \geq \frac{37}{83}(3n-\theta-6).$$

In conclusion, we have:

▶ **Theorem 6.** *Algorithm MTLK4 is a* $(37/83)$*-approximation for MPS.*

The improvement in approximation ratio is $\epsilon = 1/747$.

4 Local Optimization without Max Triangular Cactus

Recall that a diamond subgraph is a graph that is isomorphic to the graph resulting from deleting any single edge from a K_4. We call the two vertices of degree three in a diamond as the bases of the diamond, and the two vertices of degree two as the tips of the diamond. We call the edge between the two bases of a diamond the base-edge of the diamond. A diamond of a plane graph is a *facial diamond* if the two triangles of the diamond are facial triangles.

In this section, we present **LDT**, a local optimization algorithm (inspired by the one of [6]) using only diamonds and triangles that is a (5/12)-approximation algorithm for MPS on general graphs. There is a tight example for this approximation ratio. The input is a graph G, which we assume to be connected as one can run any approximation algorithm on separate connected components if needed.

Algorithm LDT. We maintain a spanning subgraph A (thus $V(A) = V(G)$) that is a dt-structure in G, starting with $E(A) = \emptyset$. The algorithm has two phases.

Phase I. The goal of this phase is to increase the cyclomatic number of A. For each connected component A' of A, the algorithm keeps a weighted tree $T_{A'}$ whose vertex set is $V(A')$ and edge set is as follows (see Figure 7 for an illustration). For every diamond of A, the edge connecting the bases is included in $T_{A'}$ with weight 2. For each of the two tips of every diamond, include in $T_{A'}$ an edge between this tip and one of the bases, with weight 1. For every triangle of A', include in $T_{A'}$ two of its edges, each with weight 1. One can easily check that $T_{A'}$ is a spanning tree of A'. Let $F(A)$ be the spanning forest on $V(G)$ obtained by taking all the edges of $E(T_{A'})$ for all the components A' of A.

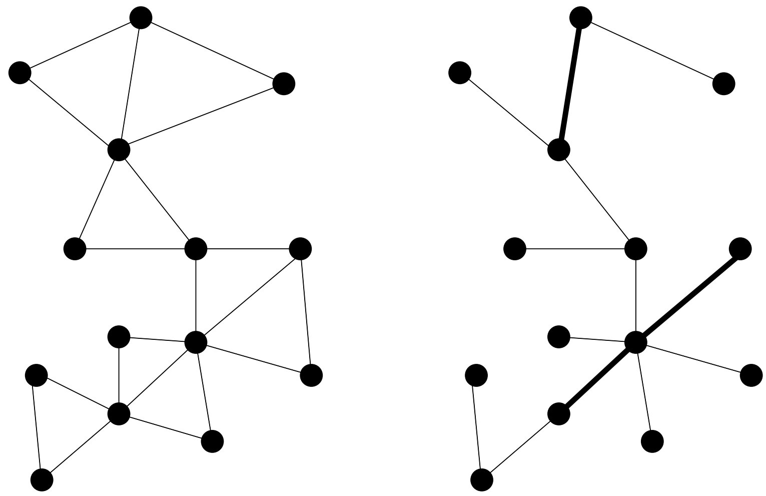

Figure 7 The tree $T_{A'}$ obtained from a connected component A' of A. Edges of weight 2 are thick.

One local optimization step is obtained as follows: if there exist a triangle of G with its three vertices in three different components of A, then add this triangle to A, and resume. If no such triangle exists, go through all the diamonds D of G. Go through all the edges $e \in E(F(A))$ of weight 1. Let $F_e(A)$ be the spanning forest of G whose edge set is $E(F(A)) \setminus \{e\}$. If the four vertices of D are in four components of $F_e(A)$, and e was in $E(F(A))$ as an edge of a triangle, then remove from A this triangle and add D. If the four vertices of D are in four components of $F_e(A)$, and e was in $E(F(A))$ as an edge that connects a tip of a diamond D' to one of the bases of D', then remove from A the two edges connecting this tip from the bases (leaving a triangle from the diamond D') and add D.

Phase II. As long as possible, greedily add edges connecting various components in graph $(V(A), E(A))$ to get the output L (L is connected since the input G is connected).

See Figure 8 for an example of a local optimization step. One can easily check that applying the local optimization step of Phase I keeps A a dt-structure and its cyclomatic number indeed increases.

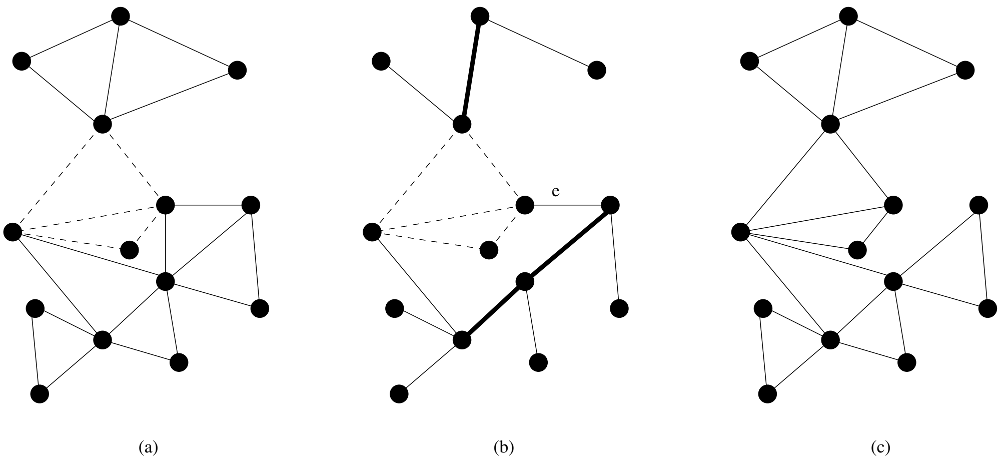

Figure 8 (a) shows in solid line segments the current dt-structure A. A diamond D, represented by the dashed line segments, is considered. (b) A forest $F(A)$ is represented by thick solid line segments having weight 2 and thin solid line segments having weight 1. We can see that by removing from $F(A)$ the weight-1 edge e, the four vertices of the diamond D are in four different components. (c) The resulting dt-structure after the local optimization step.

Running time. We discuss the running time of one local improvement, or finding out that no such improvement exists. There are $O(n)$ edges in the current dt-structure A. Using Depth First Search, computing the connected components of the current dt-structure can be done in $O(n)$. There are $O(n^3)$ triangles in the input graph. Using the information given by the connected components of the current dt-structure, it takes constant time to check whether a triangle can be used for improving the dt-structure (this involves finding, for each vertex v of the triangle, the representative of the connected component containing v, and then checking if we have three distinct representatives). For the local optimization using diamonds, we use a more elaborate method to save a factor of n in the running time.

Let $\hat{F}(A)$ be the spanning forest on $V(G)$ obtained by taking all the edges of $E(T_{A'})$ that have weight 2, for all the components A' of A. We construct $\hat{F}(A)$ and compute its connected components in time $O(n)$. Then we go through all the $O(n^4)$ diamonds and for each such diamond D we check if its four vertices belong to at least three connected components of A. If no, D cannot be used. If yes, we check if the two nodes that are in the same connected component of A are also in the same connected component of $\hat{F}(A)$; if yes, then D cannot be used. If no, in time $O(n)$ we can find one edge $e \in F(A) \setminus \hat{F}(A)$ such that the four vertices of D are in four different components of $F_e(A)$. In time $O(n^4)$ we can easily update A to include D. There are $O(n)$ local improvement steps, for a total running time of $O(n^5)$.

Tight Example. We show that the approximation ratio does not exceed 5/12 by more than an $o(1)$ term. Consider a large triangular grid H as in Figure 9, which occupies a regular hexagon, and such that H has an odd number of vertices, $2p + 1$, with a large p. Let S be some triangular cactus, with p triangles, and with $V(S) = V(H)$. We make S edge-disjoint from H, and also we avoid having the endpoints of an edge of S at distance less than 2 in H. One can easily check that H has $4p - o(p)$ faces. In "every second" face of H add a "new" point as in the figure, so that points are not added in two faces of H that share an edge in H. Connect each of these new points to the three vertices of H lying on the border of the face of H where the new point is added. The input graph G consists of the union of H and S and these new points with their incident new edges. Thus G has $4p - o(p)$ vertices. The union of H and the new points induces an almost triangulated planar graph with $12p - o(p)$ edges. The triangular cactus S can be the solution produced by the Phase I of the algorithm, as we show in the next paragraph. Then our output has no diamonds and p triangles, for a total of at most $5p$ edges (the output has cyclomatic number of p and less than $4p$ vertices). By choosing p very large, we get a ratio as close to $5/12$ as we want.

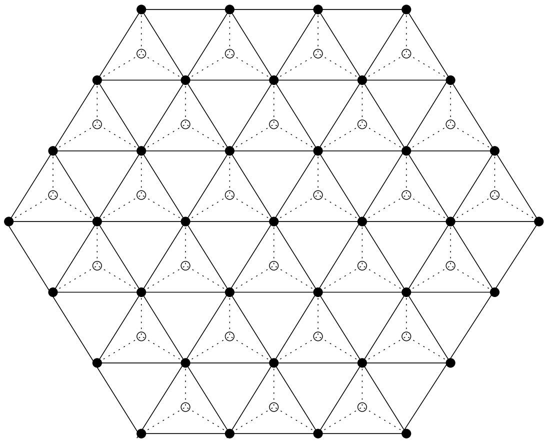

Figure 9 The planar graph H is given by filled circles representing the vertices and solid lines representing the edges. The new vertices are represented by empty circles, and the edges adjacent to these new vertices are represented by dashed lines.

It is easy to check that no triangle can be added to S. Now we check that no diamond can be swapped for a triangle in S according to Phase I of the algorithm. From our construction, any diamond of G has at least three vertices in $V(S)$, and for any edge e from $E(F(E(S)))$, the graph $F_e(E(S))$ will have one component with at least two of these three vertices.

Due to space limitations, the proof that the approximation ratio of LDT is at least $5/12$ and the proof of Theorem 8 are in the full version of the paper. We have:

▶ **Theorem 7.** *Algorithm LDT is an approximation algorithm with ratio of* $5/12$*, and can be implemented to run in time* $O(n^5)$*.*

▶ **Theorem 8.** *Algorithm GDLDT (greedily adding diamonds followed by our LDT algorithm) is an approximation algorithm with ratio of at least* $91/216$*, and at most* $3/7$*, and can be implemented to run in time* $O(n^5)$*.*

5 Conclusions and Discussion

We improved the approximation ratio for Maximum Planar Subgraph from $4/9$ to $4/9 + \epsilon$, for a small $\epsilon > 0$, by analyzing the application of a natural local optimization step after applying the previously known, Graphic-Matroid-Parity based (4/9)-approximation algorithm. Our analysis, while involved, is not tight and there may be room for a more significant improvement here. In particular, we do not believe that Equation (8) is tight. We conjecture that the tight (excluding a small additive term) bound that generalizes Theorem 5 is:

$$\Phi(\mathcal{P}, \mathcal{Q}) \geq \frac{3}{7}n - \frac{1}{7}c - \frac{10}{21}t \tag{12}$$

Here we are looking at double-partitions $(\mathcal{P}, \mathcal{Q})$ that cover all the facial triangles of a plane graph H with n vertices and t is the number of edges missing from H to be a triangulation.

As before, c is the number of cubic vertices in H, but without H being triangulated, we provide next a new definition of cubic. Say vertex v is adjacent to u, x, y forming a K_4 with v embedded strictly inside the triangle uxy. We call v cubic if none of the three triangles vux, vxy, vyu have a vertex z strictly inside that forms a K_4 with its neighbors. See Figure 10 for an example. This bound would be tight for $t = \Theta(n)$ (see the full version of the paper). Equation (12) would allow us to increase the ϵ to maybe 1/270 with the current techniques.

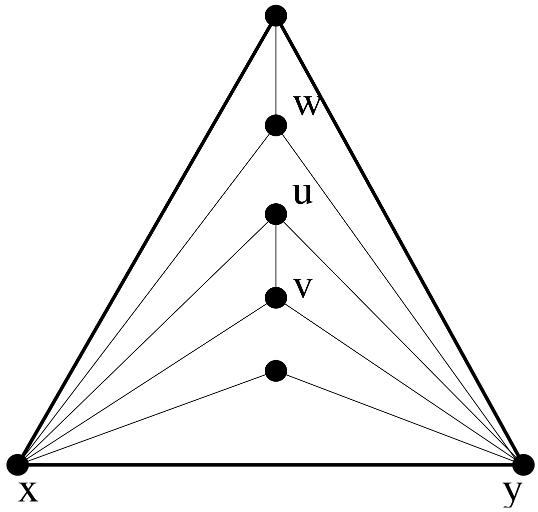

Figure 10 Here v is a cubic while w is not.

As in the previous work [5], we use the approach of using a few basic graphs for blocks in our output. This guarantees planarity and is at the basis of the analysis of all approximation algorithms for MPS with ratio bigger than 1/3. As long as one uses this approach, this paper suggests that the application of local optimization, even if it comes before or after other algorithms, is beneficial, and more powerful than Greedy algorithms (albeit slower).

An alternative would be to devise an approximation algorithm for Weighted Matroid 3-Parity, when the weights are 3 (from K_4's), 2 (from diamonds), and 1 (from triangles and C_4's). With such small integer weights, again local optimization seems the way to go.

If we are to use blocks of fixed size, it is not hard to see and known since [3] that one cannot achieve an approximation better than 1/2. But maybe the "spruces" of [6] can give us a ratio of 1/2 or better if used as the blocks of output.

References

1 Markus Blaeser, Gorav Jindal, and Anurag Pandey. A deterministic PTAS for the commutative rank of matrix spaces. *Theory of Computing*, 14:1–21, January 2018. doi:10.4086/toc.2018.v014a003.

2 Jiazhen Cai, Xiaofeng Han, and Robert E Tarjan. An $O(m \log n)$-time algorithm for the maximal planar subgraph problem. *SIAM Journal on Computing*, 22(6):1142–1162, 1993.

3 Gruia Călinescu and Cristina G. Fernandes. On the k-structure ratio in planar and outerplanar graphs. *Discrete Mathematics & Theoretical Computer Science*, 10(3):135–148, 2008. URL: http://www.dmtcs.org/dmtcs-ojs/index.php/dmtcs/article/view/961/2387, doi:10.46298/DMTCS.423.

4 Gruia Călinescu and Cristina G. Fernandes. Maximum planar subgraph. In Teofilo F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics, Second Edition, Volume 2: Contemporary and Emerging Applications*. Chapman and Hall/CRC, 2018.

5 Gruia Călinescu, Cristina G. Fernandes, Ulrich Finkler, and Howard J. Karloff. Better approximation algorithm for finding planar subgraphs. *Journal of Algorithms*, 27(2):269–302, 1998.

6 Gruia Călinescu, Cristina G. Fernandes, Hemanshu Kaul, and Alexander Zelikovsky. Maximum series-parallel subgraph. *Algorithmica*, 63(1-2):137–157, 2012. doi:10.1007/s00453-011-9523-4.

7 Parinya Chalermsook and Andreas Schmid. Finding triangles for maximum planar subgraphs. In Sheung-Hung Poon, Md. Saidur Rahman, and Hsu-Chun Yen, editors, *WALCOM: Algorithms and Computation*, pages 373–384, 2017.

8 Parinya Chalermsook, Andreas Schmid, and Sumedha Uniyal. A tight extremal bound on the Lovász cactus number in planar graphs. In *36th International Symposium on Theoretical Aspects of Computer Science*, 2019.

9 Ho Yee Cheung, Lap Chi Lau, and Kai Man Leung. Algebraic algorithms for linear matroid parity problems. *ACM Transactions on Algorithms (TALG)*, 10(3):1–26, 2014.

10 T Chiba, I Nishioka, and I Shirakawa. An algorithm of maximal planarization of graphs. In *Proc. IEEE Symp. on Circuits and Systems*, pages 649–652, 1979.

11 Markus Chimani, Ivo Hedtke, and Tilo Wiedera. Limits of greedy approximation algorithms for the maximum planar subgraph problem. In *Combinatorial Algorithms: 27th International Workshop, IWOCA 2016, Helsinki, Finland, August 17-19, 2016, Proceedings 27*, pages 334–346. Springer, 2016.

12 Markus Chimani, Ivo Hedtke, and Tilo Wiedera. Exact algorithms for the maximum planar subgraph problem: New models and experiments. *Journal of Experimental Algorithmics (JEA)*, 24:1–21, 2019.

13 Markus Chimani, Karsten Klein, and Tilo Wiedera. A note on the practicality of maximal planar subgraph algorithms. In Yifan Hu and Martin Nöllenburg, editors, *Graph Drawing and Network Visualization*, pages 357–364, Cham, 2016. Springer International Publishing.

14 Markus Chimani and Tilo Wiedera. Cycles to the rescue! novel constraints to compute maximum planar subgraphs fast. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, volume 112 of *LIPIcs*, pages 19:1–19:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ESA.2018.19.

15 Robert Cimikowski. An analysis of heuristics for graph planarization. *Journal of Information and Optimization Sciences*, 18(1):49–73, 1997.

16 M.E. Dyer, L.R. Foulds, and A.M. Frieze. Analysis of heuristics for finding a maximum weight planar subgraph. *European Journal of Operations Research*, 20:102–114, 1985.

17 Leslie R Foulds. *Graph theory applications*. Springer Science & Business Media, 2012.

18 Harold N. Gabow and Matthias F. Stallmann. An augmenting path algorithm for linear matroid parity. *Combinatorica*, 6:123–150, 1986.

19 H.N. Gabow and M. Stallmann. Efficient algorithms for graphic matroid intersection and parity. In *12th Colloq. on Automata, Language and Programming*, pages 210–220, 1985.

20 M. R. Garey and D. S. Johnson. *Computers and Intractability*. W.H. Freeman and Co., NY, 1979.

21 M. Junger and P. Mutzel. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica*, 16:33–59, 1996.

22 Jon Lee, Maxim Sviridenko, and Jan Vondrák. Matroid matching: The power of local search. *SIAM Journal on Computing*, 42(1):357–379, 2013. doi:10.1137/11083232X.

23 Annegret Liebers. Planarizing graphs—a survey and annotated bibliography. In *Graph Algorithms And Applications 2*, pages 257–330. World Scientific, 2004.

24 P.C. Liu and R.C. Geldmacher. On the deletion of nonplanar edges of a graph. In *10th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 727–738, 1977.

25 L. Lovász and M.D. Plummer. *Matching theory*. North-Holland, Amsterdam–New York, 1986.

26 James B Orlin. A fast, simpler algorithm for the matroid parity problem. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 240–258. Springer, 2008.

27 Timo Poranen. A simulated annealing algorithm for the maximum planar subgraph problem. *International Journal of Computer Mathematics*, 81(5):555–568, 2004.

28 Timo Poranen. Two new approximation algorithms for the maximum planar subgraph problem. *Acta Cybernetica*, 18(3):503–527, 2008. URL: `https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/3735`.

29 A. Schrijver. *Combinatorial Optimization*. Springer, 2003.

30 Z. Szigeti. On the graphic matroid parity problem. *J. Combin. Theory Ser. B*, 88:247–260, 2003.

31 Roberto Tamassia. Graph drawing. In *Handbook of computational geometry*, pages 937–971. Elsevier, 2000.

32 Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, September 2000.