

# A Tight Extremal Bound on the Lovász Cactus Number in Planar Graphs

**Parinya Chalermsook**

Aalto University, Espoo, Finland  
parinya.chalermsook@aalto.fi

**Andreas Schmid**

Max Planck Institute for Informatics, Saarbrücken, Germany  
aschmid@mpi-inf.mpg.de

**Sumedha Uniyal**

Aalto University, Espoo, Finland  
sumedha.uniyal@aalto.fi

---

## Abstract

A *cactus* graph is a graph in which any two cycles are edge-disjoint. We present a constructive proof of the fact that any plane graph  $G$  contains a cactus subgraph  $C$  where  $C$  contains at least a  $\frac{1}{6}$  fraction of the triangular faces of  $G$ . We also show that this ratio cannot be improved by showing a tight lower bound. Together with an algorithm for linear matroid parity, our bound implies two approximation algorithms for computing “dense planar structures” inside any graph: (i) A  $\frac{1}{6}$  approximation algorithm for, given any graph  $G$ , finding a planar subgraph with a maximum number of triangular faces; this improves upon the previous  $\frac{1}{11}$ -approximation; (ii) An alternate (and arguably more illustrative) proof of the  $\frac{4}{9}$  approximation algorithm for finding a planar subgraph with a maximum number of edges.

Our bound is obtained by analyzing a natural local search strategy and heavily exploiting the exchange arguments. Therefore, this suggests the power of local search in handling problems of this kind.

**2012 ACM Subject Classification** Mathematics of computing → Graph theory

**Keywords and phrases** Graph Drawing, Matroid Matching, Maximum Planar Subgraph, Local Search Algorithms

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2019.19

**Related Version** Full Version: <https://arxiv.org/abs/1804.03485>.

**Funding** *Parinya Chalermsook*: Part of this work was done while PC and AS were visiting the Simons Institute for the Theory of Computing. It was partially supported by the DIMACS/Simons Collaboration on Bridging Continuous and Discrete Optimization through NSF grant #CCF-1740425. Parinya has been supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 759557) and by Academy of Finland Research Fellows, under grant number 310415 and 314284.

*Sumedha Uniyal*: Partially supported by Academy of Finland under the grant agreement number 314284.

## 1 Introduction

*Linear matroid parity* (introduced in various equivalent forms [21, 18, 15]) is a key concept in combinatorial optimization that includes many important optimization problems as special cases; probably the most well-known example is the *maximum matching problem*. The polynomial-time computability of linear matroid parity made it a popular choice as an algorithmic tool for handling both theoretical and practical optimization problems. An



© Parinya Chalermsook, Andreas Schmid, and Sumedha Uniyal;  
licensed under Creative Commons License CC-BY

36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019).

Editors: Rolf Niedermeier and Christophe Paul; Article No. 19; pp. 19:1–19:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



important special case of linear matroid parity, the graphic matroid parity problem, is often explained in the language of *cacti* (see e.g. [9]), a graph in which any two cycles must be edge-disjoint. In 1980, Lovász [21] initiated the study of  $\beta(G)$  (sometimes referred to as the *cactus number* of  $G$ ), the maximum value of the number of triangles in a cactus subgraph of  $G$ , and showed that it generalizes maximum matching and can be reduced to linear matroid parity, therefore implying that  $\beta(G)$  is polynomial-time computable<sup>1,2</sup>.

Cactus graphs arise naturally in many applications<sup>3</sup>; perhaps the most relevant example in the context of approximation algorithms is the Maximum Planar Subgraph (MPS) problem: Given an input graph, find a planar subgraph with a maximum number of edges. Notice that, since any planar graph with  $n$  vertices has at most  $3n - 6$  edges, outputting a spanning tree with  $n - 1$  edges immediately gives a  $\frac{1}{3}$ -approximation algorithm. Generalizing the idea of finding spanning trees, one would like to look for a planar graph  $H$ , denser than a spanning tree, and at the same time efficiently computable. Calinescu et al. [3] showed that a cactus subgraph with a maximum number of triangles (which is efficiently computable via matroid parity algorithms) could be used to construct a  $\frac{4}{9}$ -approximation for MPS.

The  $\frac{4}{9}$ -approximation for MPS was achieved through an extremal bound of  $\beta(G)$  when  $G$  is a plane graph. In particular, it was proven that  $\beta(G) \geq \frac{1}{3}(n - 2 - t(G))$ , where  $n = |V(G)|$  and  $t(G) = (3n - 6) - |E(G)|$  (i.e. the number of edges missing for  $G$  to be a triangulated plane graph).

## 1.1 Our Results

In this work, we are interested in further studying the extremal properties of  $\beta(G)$  and exhibit stronger algorithmic implications. Our main result is summarized in the following theorem.

► **Theorem 1.** *Let  $G$  be a plane graph. Then  $\beta(G) \geq \frac{1}{6}f_3(G)$  where  $f_3(G)$  denotes the number of triangular faces in  $G$ . Moreover, a natural local search 2-swap algorithm achieves this bound.*

It is not hard to see that  $f_3(G) \geq 2n - 4 - 2t(G)$  where  $t(G)$  denotes the number of edges missing for  $G$  to be a triangulated plane graph. Therefore, we obtain the main result of [3] immediately.

► **Corollary 2.**  *$\beta(G) \geq \frac{1}{3}(n - 2 - t(G))$ . Hence, the matroid parity algorithm gives a  $\frac{4}{9}$ -approximation for MPS.*

Besides implying the MPS result, we exhibit further implications of our bound. Recently in [7], the authors introduced *Maximum Planar Triangles (MPT)*, where the goal is to find a plane subgraph with a maximum number of triangular faces. It was shown that an approximation algorithm for MPT naturally translates into one for MPS, where a  $\frac{1}{6}$  approximate MPT solution could be turned into a  $\frac{4}{9}$  approximate MPS solution. However, the authors only managed to show a  $\frac{1}{11}$  approximation for MPT.

Although the only change from MPS to MPT lies in the objective of maximizing the number of triangular faces instead of edges, the MPT objective seems much harder to handle, for instance, the extremal bound provided in [3] is not sufficient to derive any approximation algorithm for MPT.

<sup>1</sup> There are many efficient algorithms for matroid parity (both randomized and deterministic), e.g. [9, 22, 24, 12].

<sup>2</sup> When we study  $\beta(G)$ , notice that a cactus subgraph that achieves the maximum value of  $\beta(G)$  would only need to have cycles of length three (triangles). Such cacti are called *triangular cacti*.

<sup>3</sup> See for instance the wikipedia page [https://en.wikipedia.org/wiki/Cactus\\_graph](https://en.wikipedia.org/wiki/Cactus_graph).

Theorem 1 therefore implies the following result for MPT.

► **Corollary 3.** *A matroid parity algorithm gives a  $\frac{1}{6}$  approximation algorithm for MPT.*

Our conceptual contributions are the following:

1. Our result further highlights the extremal role of the cactus number in finding a dense planar structure, as illustrated by the fact that our bound on  $\beta(G)$  is more “robust” to the change of objectives from MPS to MPT. It allows us to reach the limit of approximation algorithms that matroid parity provides for both MPS and MPT.
2. Our work implies that local search arguments alone are sufficient to “almost” reach the best known approximation results for both MPS and MPT in the following sense: Matroid parity admits a PTAS via local search [19, 2]. 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 and  $\frac{1}{6} + \epsilon$  approximation for MPT. Therefore, this suggests that local search might be a promising candidate for such problems.
3. Finally, in some ways, our work can be seen as an effort to open up all the black boxes used in MPS algorithms with the hope of learning algorithmic insights that are crucial for making progress on this kind of problems. In more detail, there are two main “black boxes” hidden in the MPS result: (i) The use of Lovász min-max cactus formula in deriving the bound  $\beta(G) \geq \frac{1}{3}(n - 2 - t(G))$ , and (ii) the use of a matroid parity algorithm as a blackbox in computing  $\beta(G)$ . Our bound for  $\beta(G)$  is now purely combinatorial (and even constructive) and manages to by-pass (i).

**Related work.** On the hardness of approximation side, MPS is known to be APX-hard [3], while MPT is only known to be NP-hard [7]. In combinatorial optimization, there are a number of problems closely related to MPS and MPT. For instance, finding a maximum series-parallel subgraph [5] or a maximum outer-planar graph [3], as well as the weighted variant of these problems [4]; these are the problems whose objectives are to maximize the number of edges.

Perhaps the most famous extremal bound in the context of cactus is the min-max formula of Lovász [21] and a follow-up formula that is more illustrative in the context of cactus [25]. All these formulas generalize the Tutte-Berge formula [1, 26] that has been used extensively both in research and curriculum.

Another related set of problems has the objectives of maximizing the number of vertices, instead of edges. In particular, in the maximum induced planar subgraph (i.e. given graph  $G$ , one aims at finding a set of nodes  $S \subseteq V(G)$  such that  $G[S]$  is planar, while maximizing  $|S|$ .) This variant has been studied under a more generic name, called *maximum subgraph with hereditary property* [23, 20, 13]. This variant is unfortunately much harder to approximate:  $\tilde{\Omega}(|V(G)|)^4$  hard to approximate [14, 17]; in fact, the problems in this family do not even admit any FPT approximation algorithm [6], assuming the *gap exponential time hypothesis* (Gap-ETH).

## 1.2 Overview of Techniques

We give a high-level overview of our techniques. The description in this section assumes certain familiarity with how standard local search analysis is often done.

---

<sup>4</sup> The term  $\tilde{\Omega}$  hides asymptotically smaller factors.

Our algorithm works as follows. Let  $G$  be an input plane graph, and let  $\mathcal{C}$  be a cactus subgraph of  $G$  whose triangles correspond to triangular faces of  $G$ . The local search operation,  $t$ -swap, is done as follows: As long as there is a collection  $X \subseteq \mathcal{C}$  of  $\ell : \ell \leq t$  edge-disjoint triangles and  $Y$  such that  $(\mathcal{C} \setminus X) \cup Y$  contains more triangular faces of  $G$  than  $\mathcal{C}$  and it remains a cactus, we perform such an improvement step. A cactus subgraph is called locally  $t$ -swap optimal, if it can not be improved by a  $t$ -swap operation. Remark that the triangles chosen by our local search are only those which are triangular faces in the input graph  $G$  (we assume that the drawing of  $G$  is fixed.)

Our analysis is highly technical, although the basic idea is very simple and intuitive. We give a high-level overview of the analysis. We remark that this description is overly simplified, but it sufficiently captures the crux of our arguments. Let  $\mathcal{C}$  be the solution obtained by the local search 2-swap algorithm. We argue that the number of triangles in  $\mathcal{C}$  is at least  $f_3(G)/6$ . We remark that the 2-swap is required, as we are aware of a bad example  $H$  for which the 1-swap local search only achieves a bound of  $(\frac{1}{7} + o(1))f_3(H)$ . For simplicity, let us assume that  $\mathcal{C}$  has only one non-singleton component. Let  $S \subseteq V(G)$  be the vertices in such a connected component.

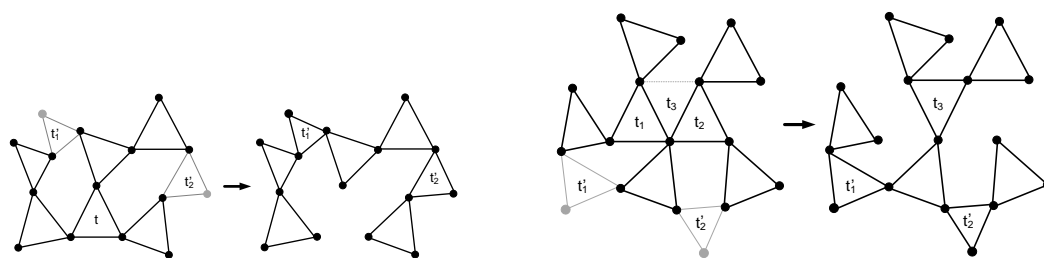
Let  $t$  be a triangle in  $\mathcal{C}$ . Notice that removing the three edges of  $t$  from  $\mathcal{C}$  breaks the cactus into at most three components, say  $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3$  that are pairwise vertex-disjoint, i.e. sets  $S_j = V(\mathcal{C}_j)$  are pairwise vertex-disjoint. Recall at this point that we would like to upper bound the number of triangles in  $G$  by six times  $\Delta$ , where  $\Delta$  is the number of triangles in the cactus  $\mathcal{C}$ . Notice that  $f_3(G)$  is comprised of  $f_3(G[S_1]) + f_3(G[S_2]) + f_3(G[S_3]) + q'$ , where  $q'$  is the number of triangles in  $G$  “across” the components  $S_j$  (i.e. those triangles whose vertices intersect with at least two sets  $S_i, S_j$ , where  $i \neq j$ ). Therefore, if we could somehow give a nice upper bound on  $q'$ , e.g. if  $q' \leq 6$ , then we could inductively use  $f_3(G[S_j]) \leq 6\Delta_j$  where  $\Delta_j$  is the number of triangles in  $\mathcal{C}_j$ , and that therefore

$$f_3(G) \leq 6(\Delta_1 + \Delta_2 + \Delta_3) + 6 \leq 6(\Delta - 1) + 6 = 6\Delta$$

and we would be done. However, it is not possible to give a nice upper bound on  $q'$  that holds in general for all situations. We observe that such a bound can be proven for some suitable choice of  $t$ : Roughly speaking, removing such a triangle  $t$  from  $\mathcal{C}$  would create a small “interaction” between components  $\mathcal{C}_j$  (i.e. small  $q'$ ). We say that such a triangle  $t$  is a *light* triangle; otherwise, we say that it is *heavy*. Let  $\mathcal{C}'$  be the current cactus we are considering. As long as there is a light triangle left in  $\mathcal{C}'$ , we would remove it (thus breaking  $\mathcal{C}'$  into  $\mathcal{C}'_1, \mathcal{C}'_2, \mathcal{C}'_3$ ) and inductively use the bound for each  $\mathcal{C}'_j$ . Therefore, we have reduced the problem to that of analyzing the base case of a cactus in which all triangles are heavy. Handling the base case of the inductive proof is the main challenge of our result.

We sketch here the two key ideas. Let  $S = V(\mathcal{C})$ . The first key idea is the way we exploit the locally optimal solution in certain parts of the graph  $G[S]$ . We want to point out; the fact that all triangles in  $\mathcal{C}$  are heavy is exploited crucially in this step. Recall that, each heavy triangle is such that its removal creates three components  $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$  with many “interactions” (i.e. many triangles across components) between them. This large amount of interaction is the main reason why we could not use induction before. However, intuitively, these triangles across components could serve as candidates for making local improvements. So the fact that there are many interactions would become our advantage in the local search analysis.

We briefly illustrate how we take advantage of heavy triangles. Let  $\mathcal{T}$  be the set of triangular faces in  $G$  that are not contained in  $\bigcup_i G[S_i]$ , so each triangle in  $\mathcal{T}$  has vertices in at least two subsets  $S_j, S_i$  where  $j \neq i$ . The local search argument would allow us to say that all triangles in  $\mathcal{T}$  have one vertex in  $S_i$ , one in  $S_j$  and one outside of  $S_1 \cup S_2 \cup S_3$ . This idea is illustrated in Figure 1a.



(a) A 1-swap operation. If there were two triangles  $t'_1, t'_2$  in  $\mathcal{T}$  between two different pairs of components  $S_j, S_i$  (where  $j \neq i$ ), we could remove  $t$  from  $\mathcal{C}$  and add  $t'_1, t'_2$  to get a better cactus.

(b) A 2-swap operation. Let  $t_1$  and  $t_2$  be two adjacent triangles in our cactus. If there was an edge between  $t_1$  and  $t_2$ , then there would exist a local improvement by removing  $t_1$  and  $t_2$  from  $\mathcal{C}$  and adding  $t'_1, t'_2$  and  $t_3$ .

■ **Figure 1** Two examples for the swap operations.

Moreover, we will argue that there are not too many triangular faces in  $G[S]$ , and we give a rough idea of how the exchange argument can be used in Figure 1b.

Finally, the ideas illustrated in both figures are only applied locally in a certain “region” inside the input planar graph  $G$ , so globally it is still unclear what would happen. Our final ingredient is a way to decompose the regions inside a plane graph into various “atomic” types. For each such atomic type, the local exchange argument is sufficient to argue optimally about the number of triangles in  $G$  in that region compared to that in the cactus. Combining the bounds on these atomic types gives us the desired result. This is the most technically involved part of the paper, and we present it gradually by first showing the analysis that gives  $\beta(G) \geq \frac{1}{7}f_3(G)$ . For this, we need to classify the regions into five atomic types. To prove the main theorem, that  $\beta(G) \geq \frac{1}{6}f_3(G)$ , we need a more complicated classification into thirteen atomic types.

**Organization of the paper.** In Section 2, we give a detailed overview for the proof of our main result. As the proof in full detail would be too long to fit in this extended abstract, we refer the interested reader to a full version on arXiv [8]. In Section 3, we present how to construct a planar graph for which the bound proven in Theorem 1 is tight. In addition we show how it implies the extremal bound provided in [3]. In Section 4, we point out possible directions for future research and extensions of our work.

## 2 Overview of the Proof

In this section, we give a formal overview of the structure of the proof of Theorem 1. Let our input  $G$  be a plane graph (a planar graph with a fixed drawing). Let  $\mathcal{C}$  be a locally optimal triangular cactus solution for the natural local search algorithm that uses 2-swap operations, as described in the previous section. Let  $\Delta(\mathcal{C})$  denote the number of triangular faces of  $\mathcal{C}$  which correspond to the triangular faces of  $G$ . We will show  $\Delta(\mathcal{C}) \geq f_3(G)/6$ . In general, we will use the function  $\Delta : G \rightarrow \mathbb{N}$  to denote the number of triangular faces in any plane graph  $G$ .

We partition the vertices in  $G$  into subsets based on the connected components of  $\mathcal{C}$ , *i.e.*  $V(G) = \bigcup_i S_i$  where  $\mathcal{C}[S_i]$  is a connected cactus subgraph of  $\mathcal{C}$ . For each  $i$ , where  $|S_i| \geq 1$ , let  $q(S_i)$  denote the number of triangular faces in  $G$  with at least two nodes in  $S_i$ . The following proposition holds by the 2-swap optimality of  $\mathcal{C}$  which implies  $f_3(G) = \sum_i q(S_i)$ .

► **Proposition 4.** *If  $\Delta(\mathcal{C}_i) \geq \frac{1}{6}q(S_i)$  for all  $i$ , then  $\Delta(\mathcal{C}) \geq \frac{1}{6}f_3(G)$ .*

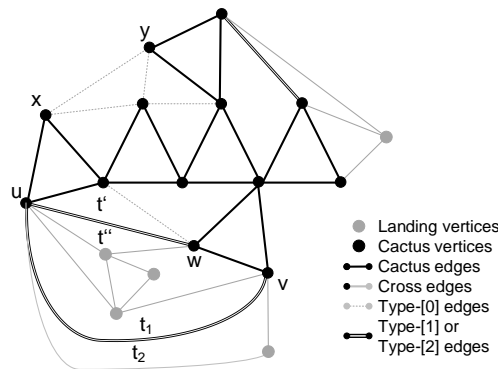
Therefore, it is sufficient to analyze any arbitrary component  $S_i$  where  $\mathcal{C}[S_i]$  contains at least one triangle of  $\mathcal{C}$  (if the component does not contain such a triangle it is just a singleton vertex) and show that  $\Delta(\mathcal{C}_i) \geq \frac{1}{6}q(S_i)$ . Thus, from now on, we fix such an arbitrary component  $S_i$  and denote  $S_i$  simply by  $S$ ,  $q(S_i)$  by  $q(S)$ , and  $\Delta(\mathcal{C}[S_i])$  by  $p$ . We will show that  $q \leq 6p$  through several steps.

### Step 1: Reduction to Heavy Cactus

In the first step, we will show that the general case can be reduced to the case where all triangles in  $\mathcal{C}$  are *heavy* (to be defined below). We refer to different types of vertices, edges and triangles in the graph  $G$  as follows:

- **Cactus.** All edges/vertices/triangles in the cactus  $\mathcal{C}[S]$  are called *cactus edges/vertices/triangles* respectively.
- **Cross.** Edges with exactly one end-point in  $S$  are called *cross edges*. Triangles that use one vertex outside of  $S$  are *cross triangles*. Notice that each cross triangle has exactly one edge in  $G[S]$ , that edge is called a *supporting edge* of the cross triangle. Similarly, we say that an edge  $e \in E(G[S])$  supports a cross triangle; such a cross triangle  $t$  contains exactly one vertex  $v$  in some component  $S_i \neq S$ . The component  $S_i$  is called the *landing component* of  $t$ . Similarly the vertex  $v$  is called the *landing vertex* of  $t$ .
- **type- $[i]$  edges.** An edge in  $G[S]$  that is not a cactus edge and does not support a cross triangle is called a *type- $[0]$  edge*. An edge in  $G[S]$  that is not a cactus edge and supports  $i$  cross triangle(s) is called a *type- $[i]$  edge*.

Therefore, each edge in  $G[S]$  is a cactus, type-0, type-1 or type-2 edge. The introduced naming convention makes it easier to make important observations like the following (see Figure 2 for an illustration of our naming convention).



■ **Figure 2** Various types of edges, vertices and triangles. Here the cross triangles  $t''$  and  $t_1$  have the same landing component.

► **Observation 5.** *Triangles that contribute to the value of  $q$  are of the following types: (i) the cactus triangles; (ii) the cross triangles; and (iii) the “remaining” triangles that connect three cactus vertices using at least one type-0, type-1 or type-2 edge, and do not have a cross triangle drawn inside.*

**Types of cactus triangles and Split cacti.** Consider a (cactus) triangle  $t$  in  $\mathcal{C}$ . For  $i \in \{0, 1, 2, 3\}$ , we say that  $t$  is of type- $i$  if exactly  $i$  of its edges support a cross triangle. Let  $p_i$  denote the number of type- $i$  cactus triangles, so we have that  $p_0 + p_1 + p_2 + p_3 = p$ .

We denote the operation of deleting the edges of  $t$  from a connected cactus  $\mathcal{C}[S]$  by *splitting*  $\mathcal{C}[S]$  at  $t$ . The resulting three smaller triangular cacti (denoted by  $\{\mathcal{C}_v^t\}_{v \in V(t)}$ ) are referred to as the *split cacti* of  $t$ . For each  $v \in V(t)$ , let  $S_v^t := V(\mathcal{C}_v^t)$  be the *split component* containing  $v$ . Let  $u, v \in V(t) : u \neq v$ . Denote by  $B_{uv}^t$  the set of type-1 or type-2 edges having one endpoint in  $S_u^t$  and the other in  $S_v^t$ . Now we are ready to define the concept of heavy and light cactus triangles, which will be crucially used in our analysis.

**Heavy and light cactus triangles.** We say that a cactus triangle  $t$  is *heavy* if either there are at least four cross triangles supported by  $E(t) \cup \bigcup_{uv \in E(t)} B_{uv}^t$  or there are at least three cross triangles supported by the edges in one set  $B_{uv}^t \cup uv$  for some  $uv \in E(t)$  and no cross triangle supported by the rest of the sets  $B_{ww'}^t \cup ww'$  for each  $ww' \in E(t)$ . Otherwise, the triangle is *light*. Intuitively, the notion of a light cactus triangle  $t$  captures the fact that, after removing  $t$ , there is only a small amount of “interaction” between the split components.

We will abuse the notations a bit by using  $S$  instead of  $V[S]$ . Recall, that we denote by  $q(S)$  the total number of triangular faces in  $G$  with exactly two vertices in  $S$ . We denote by  $p(S)$  the total number of triangles in the cactus  $\mathcal{C}[S]$ .

**Function  $\varphi$ .** Consider a set  $S \subseteq V(G)$  and a drawing of  $G[S]$  (since we are talking about a fixed drawing of the plane graph  $G$ , this is well-defined). Denote by  $\ell(S)$  the length of the outer-face  $f_S$  of the graph  $G[S]$ . We define  $\varphi(S)$  as the number of edges on the outer-face that do not support any cross triangle drawn on the outer-face, so we have  $0 \leq \varphi(S) \leq \ell(S)$ .

The main ingredients of Step 1 are encapsulated in the following theorem.

► **Theorem 6** (Reduction to heavy triangles). *Let  $\gamma \geq 6$  be a real number, and  $\varphi$  be as described above. If  $q(S) \leq \gamma p(S) - \varphi(S)$  for all  $S$  for which  $\mathcal{C}[S]$  is a connected cactus that contains no light triangle, then  $q(S) \leq \gamma p(S) - \varphi(S)$  for all  $S$ .*

Therefore, if we manage to show the bound  $q(S) \leq \gamma p(S) - \varphi(S)$  for the heavy cactus, it will follow that  $q \leq \gamma p$  in general (due to non-negativity of function  $\varphi$ ). In other words, this gives a reduction from the general case to the case when all cactus triangles are heavy. We end the description of Step 1 by presenting the description of  $\varphi$ .

## Step 2: Skeleton and Surviving Triangles

Now, we focus on the case when there are only heavy triangles in the given cactus, and we will give a formal overview of the key idea we use to derive the bound  $q(S) \leq 6p(S) - \varphi(S)$ , which in combination with Theorem 6, gives our main Theorem 1. For convenience, we refer to the terms  $p(S)$  and  $q(S)$  as simply  $p$  and  $q$  respectively.

**Structures of heavy triangles.** Using local search’s swap operations, the light and heavy triangles behave in a very well structured manner. The following proposition summarizes these structures for heavy triangles.

► **Proposition 7.** *Let  $t$  be a cactus triangle in cactus  $\mathcal{C}[S]$ .*

■ *If  $t$  is heavy, then  $t$  is either type-0 or type-1.*



## 19:8 A Tight Extremal Bound on the Lovász Cactus Number in Planar Graphs

- If  $t$  is a heavy type-1 triangle and the edge  $uv \in E(t)$  supports the cross triangle supported by  $t$ , then  $B_{ww'}^t = \emptyset$  for all  $ww' \in E(t) \setminus \{uv\}$  and the total number of cross triangles supported by edges in  $B_{uv}^t$  is greater than or equal to two.
- If  $t$  is a heavy type-0 triangle, then there is an edge  $uv \in E(t)$  such that  $B_{ww'}^t = \emptyset$  for all  $ww' \in E(t) \setminus \{uv\}$  and the total number of cross triangles supported by edges in  $B_{uv}^t$  is greater than or equal to three.

By Proposition 7 we can only have type-0 and type-1 cactus triangles in  $\mathcal{C}$ . Moreover, for each such heavy triangle  $t$ , the type-1 or type-2 edges in  $G[S]$  only connect vertices of two split components of  $t$ .

Let  $a_i$  be the number of edges of type- $i$ . Notice that the number of non-cactus edges in  $G[S]$  is  $\sum_i a_i = |E(G[S])| - 3p$ .

**Skeleton graph  $H$ .** Let  $A$  be the set of all type-0 edges in  $G[S]$  and  $H := H[S] := G[S] \setminus A$ . Thus  $H[S]$  contains only cactus or type-1 or type-2 edges.

Each face  $f$  of  $H$  possibly contains several faces of  $G$ , so we will refer to such a face as a *super-face*. At high-level, our plan is to analyze each super-face  $f$ , providing an upper bound on the number of triangular faces of  $G$  drawn inside  $f$ , and then sum over all such  $f$  to retrieve the final result. We call  $H$  a *skeleton graph* of  $G$ , whose goal is to provide a decomposition of the faces of  $G$  into structured super-faces. Denote by  $\mathcal{F}$  the set of all super-faces (except for the  $p$  faces corresponding to cactus triangles).

Let  $f$  be a super-face. Denote by  $survive(f)$  the number of triangular faces of  $G$  drawn inside  $f$  that do not contain any cross triangles. Now we do a simple counting argument for  $q$  using the skeleton  $H$  as follows: (i) There are  $p$  cactus triangles in  $H$ , (ii) There are  $p_1 + a_1 + 2a_2$  cross triangles supported by edges in  $G[S]$ , and (iii) There are  $\sum_{f \in \mathcal{F}} survive(f)$  triangular faces in  $G$  that were not counted in (i) or (ii). Combining this, we obtain:

$$q \leq p + (p_1 + a_1 + 2a_2) + \sum_{f \in \mathcal{F}} survive(f) \quad (1)$$

The first and second terms are expressed nicely as functions of  $p$ 's and  $a$ 's, so the key is to achieve the best upper bound on the third term in terms of the same parameters. Roughly speaking, the intuition is the following: When  $a_2$  or  $a_1$  is high (there are many edges in  $G[S]$  supporting cross triangles), the second term becomes higher. However, each cross triangle would need to be drawn inside some face in  $G[S]$ , therefore decreasing the value of the term  $\sum_{f \in \mathcal{F}} survive(f)$ . Similar arguments can be made for  $p_1$ . Therefore, the key to a tight analysis is to understand this trade-off.

**The structure of super-faces.** Let  $f \in \mathcal{F}$  be a super-face. Recall that an edge in the boundary of  $f$  is either a type-1 or type-2 edge, or a cactus edge. We aim for a better understanding of the value of  $survive(f)$ . In general, this value can be as high as  $|E(f)| - 2$ , e.g. if  $G[V(f)]$  is a triangulation of the region bounded by the super-face  $f$  using type-0 edges. However, if some edge in the boundary of  $f$  supports a cross triangle whose landing component is drawn inside of  $f$  in  $G$ , this would decrease the value of  $survive(f)$ , by *killing* the triangular face adjacent to it, hence the term *survive*.

The following observation is crucial in our analysis:

- **Observation 8.** Consider each edge  $e \in E(f)$ . There are two possible cases:
- Edge  $e$  is a type-1 or type-2 or cactus edge and supports a cross triangle drawn in  $f$ .
  - Edge  $e$  is a type-1 or type-2 or cactus edge and does not support any cross triangle drawn in  $f$ .



Edges lying in the first case are called occupied edges (the set of such edges in  $E(f)$  is denoted by  $Occ(f)$ ), while the others are called free edges in  $f$  (the set of free edges in  $E(f)$  is denoted by  $Free(f)$ ). The length of  $f$  can be written as  $|E(f)| = |Occ(f)| + |Free(f)|$ .

A very important quantity for our analysis is  $\mu(f) = \frac{1}{2} \cdot |Occ(f)| + |Free(f)|$ , roughly bounding the value of  $survive(f)$  (within some small constant additives terms.)

We will assume without loss of generality that  $survive(f)$  is the maximum possible value of surviving triangles that can be obtained by drawing type-0 edges in  $f$ , so  $\mu(f)$  is a function that depends only on the bounding edges in  $f$ . We define  $gain(f) = \mu(f) - survive(f)$ , which is again a function that only depends on bounding edges of  $f$ . Intuitively, the higher the term  $gain(f)$ , the better for us (since this would lower the value of  $survive(f)$ ), and in fact, it will later become clear that  $gain(f)$  roughly captures the “effectiveness” of a local exchange argument on the super-face  $f$ . Hence, it suffices to show that  $\sum_{f \in \mathcal{F}} gain(f)$  is sufficiently large. The following proposition makes this precise:

► **Proposition 9.**  $\sum_{f \in \mathcal{F}} survive(f) = (3p - 0.5p_1 + 1.5a_1 + a_2) - \sum_{f \in \mathcal{F}} gain(f)$

**Proof.** Notice that  $\sum_{f \in \mathcal{F}} \mu(f)$  can be analyzed as follows:

- Each cactus triangle is counted three times (once for each of its edges), and for a type-1 triangle, one of the three edges contribute only one half. Therefore, this accounts for the term  $3p - 0.5p_1$ .
- Each type-1 or type-2 edge is counted two times (once per super-face containing it in its boundary). For a type-2 edge, the contribution is always half (since it always is accounted in  $Occ(f)$ ). For a type-1 edge, the contribution is half on the occupied case, and full on the free case. Therefore, this accounts for the term  $1.5a_1 + a_2$ .

Overall we get,  $\sum_{f \in \mathcal{F}} \mu(f) = 3p - 0.5p_1 + 1.5a_1 + a_2$ , which finishes the proof. ◀

Combining this proposition with Equation 1, we get:

$$q \leq 4p + 0.5p_1 + 2.5a_1 + 3a_2 - \sum_{f \in \mathcal{F}} gain(f) \tag{2}$$

**A warm-up: Using the gains to prove a weaker bound.** To recap, after Step 1 and Step 2, we have reduced the analysis to the question of lower bounding  $\sum_{f \in \mathcal{F}} gain(f)$ . We first illustrate that we could get a weaker (but non-trivial) result compared to our main result by using a generic upper bound on the gains. In Step 3, we will show how to substantially improve this bound, achieving the ratio of our main Theorem 1 which is tight.

► **Lemma 10.** *For any super-face (except for the outer-face) in  $\mathcal{F}$ , we have  $gain(f) \geq 1.5$ .*

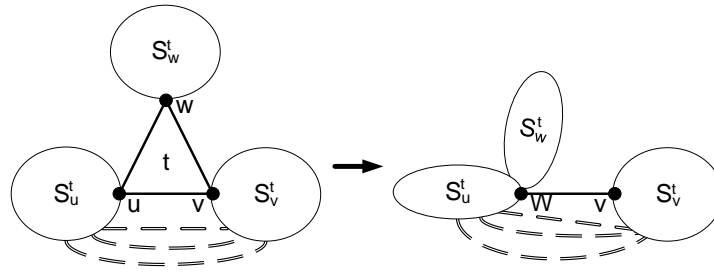
As the outer (super-)face  $f_0$  of  $H[S]$  is special, we can achieve a lower bound on the quantity  $gain(f_0)$  that depends on  $\varphi(S)$ . This is captured by the following lemma.

► **Lemma 11.** *For the outer-face  $f_0$ , we have that  $gain(f) \geq \varphi(S) - 1$ .*

$$\sum_{f \in \mathcal{F}} gain(f) \geq \varphi(S) - 1 + 1.5(|\mathcal{F}| - 1) = \varphi(S) + 1.5|\mathcal{F}| - 0.5 \tag{3}$$

The following lemma upper bounds the number of skeleton faces (i.e. super-faces of the skeleton.)

► **Lemma 12.**  $|\mathcal{F}| = a_1 + a_2 + 1 \leq 2p - 2$ .



■ **Figure 3** An example of the contraction transformation.

**Proof.** Proposition 7 allows us to modify the graph  $H$  into another simple planar graph  $\tilde{H}$  such that the claimed upper bound on  $|\mathcal{F}|$  will follow simply from Euler’s formula.

Let  $t$  be a cactus triangle where  $V(t) = \{u, v, w\}$  and  $uw \in E(t)$  be such that the edge set  $B_{uw}^t$  is empty, as guaranteed in Proposition 7. For every cactus triangle  $t$  we contract the edge  $uw$  into one new vertex  $W$ . Note that this operation creates two parallel edges with endpoints  $W$  and  $v$  in the resulting graph. To avoid multi-edges in the resulting graph  $\tilde{H}$  we remove one of them (see Figure 3 for an illustration of this operation). Since  $B_{uw}^t$  is empty this operation cannot create any other multi-edges in  $\tilde{H}$ . In addition the contraction of an edge maintains planarity, hence after each such transformation the graph remains simple and planar. As a result of applying the above operation to all cactus triangles, the graph  $\tilde{H}$  has  $p + 1$  vertices and  $p$  edges corresponding to the contracted triangles. By Euler’s formula the number of edges in  $\tilde{H}$  is at most  $3(p + 1) - 6 = 3p - 3$ , which implies that  $a_1 + a_2 \leq 2p - 3$ , and as  $|\mathcal{F}| = a_1 + a_2 + 1$  we get that  $|\mathcal{F}| \leq 2p - 2$ . ◀

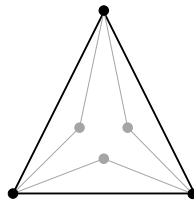
Combining the trivial gains (i.e. Inequality 3) with Inequality 2, we get

$$q \leq (4p + 0.5p_1 + 2.5a_1 + 3a_2) - (\varphi(S) + 1.5(a_1 + a_2 + 1) - 2.5) = 4p + 0.5p_1 + a_1 + 1.5a_2 - \varphi(S) + 1$$

Now, using Lemma 12 and the trivial bound that  $p_1 \leq p$ , we get  $q(S) \leq 4.5p + 1.5(a_1 + a_2) - \varphi(S) + 1 \leq 7.5p(S) - \varphi(S)$ , therefore implying a factor 7.5 upper bound.

### Step 3: Upper Bounding Gains via Super-Face Classification

In this final step, we show another crucial idea that allows us to reach a factor 6. Intuitively, the most difficult part of lower bounding the total gain is the fact that the value of  $gain(f)$  is different for each type of super-face, and one cannot expect a strong “universal” upper bound that holds for all of them. For instance, Figure 4 shows a super-face with  $gain(f) = 1.5$ , so strictly speaking, we cannot improve the generic bound of 1.5.



■ **Figure 4** A super-face  $f \in \mathcal{F}$  having  $gain(f) = 1.5$ ;  $\mu(f) = 1.5$  and  $survive(f) = 0$ .

This is where we introduce our final ingredient, that we call *classification scheme*. Roughly, we would like to “classify” the super-faces in  $\mathcal{F}$  into several types, each of which has the same gain. Analyzing super-faces with similar gains together allows us to achieve a better result.

**Super-face classification scheme.** We are interested in coming up with a set of rules  $\Phi$  that classifies  $\mathcal{F}$  into several types. We say that the rule  $\Phi$  is a  $d$ -type classification if the rules classifies  $\mathcal{F}$  into  $d$  sets  $\mathcal{F} = \bigcup_{j=1}^d \mathcal{F}[j]$ . Let  $\vec{\chi}$  be a vector such that  $\chi[i] = |\mathcal{F}[i]|$ . We would like to prove a good lower bound on the gain for each such set. We define the gain vector by  $\vec{gain}$  where  $\vec{gain}[i] = \min_{f \in \mathcal{F}[i]} gain(f)$ . The total gain can be rewritten as:

$$\sum_{f \in \mathcal{F}} gain(f) = \vec{gain} \cdot \vec{\chi}$$

Notice that, the total gain value  $\vec{gain} \cdot \vec{\chi}$  would be written in terms of the  $\chi[j]$  variables, so we would need another ingredient to lower bound this in terms of variables  $p$ 's and  $a$ 's. Therefore, another component of the classification scheme is a set of *valid linear inequalities*  $\Psi$  of the form  $\sum_{j=1}^d C_j \chi[j] \leq \sum_{j \in \{0,1\}} d_j p_j + \sum_{j \in \{1,2\}} d'_j a_j$ . This set of inequalities will allow us to map the formula in terms of  $\chi[j]$  into one in terms of only  $p$ 's and  $a$ 's.

A classification scheme is defined as a pair  $(\Phi, \Psi)$ . We say that such a scheme certifies the proof of factor  $\gamma$  if it can be used to derive  $q(S) \leq \gamma p(S) - \varphi(S)$ . Given a fixed classification scheme and a gain vector, we can check whether it certifies a factor  $\gamma$  by using an LP solver (although in our proof, we would show this derivation.)

Our main result is a scheme that certifies a factor 6. Since the proof is complicated, we also provide a simpler, more intuitive proof that certifies a factor 7 first.

► **Theorem 13.** *There is a 5-type classification scheme that gives a factor 7.*

We remark that the analysis of factor 7 only requires a cactus that is locally optimal for 1-swap.

► **Theorem 14.** *There is a 13-type classification scheme that gives a factor 6.*

**Intuition.** The classification scheme would intuitively set the rules to separate the super-faces that would benefit from local search's exchange argument from those that would not. Therefore, for the good cases, we would obtain a much better gain, e.g., in one of our classification type,  $gain(f)$  is as high as 4.5. In the bad cases that there is no such benefit, we would still use the lower bound of 1.5 that holds in general for any super-face.

### 3 On the Strength of Our Result

#### 3.1 Our Bound is Almost Tight

In this section, we show that there exists a graph  $G$  for which  $\beta(G) \leq (\frac{1}{6} + o(1))f_3(G)$ . We show this indirectly using a family of graphs presented in [7], as stated in the following lemma.

► **Lemma 15** ([7]). *There is a family of  $n$ -vertex planar graphs  $\{H_n\}_{n \in \mathbb{Z}}$  for which there exist a maximal cactus subgraph  $C_n$  of  $H_n$  such that  $\frac{f_3(C_n)}{f_3(H_n)} \leq \frac{1}{12} + o_n(1)$ .*

In [7], this family of graphs is used to show that a maximal cactus (not maximum) is not sufficient to improve over the best known greedy strategies when approximating MPT. In the context of this paper we use  $C_n$  to compare it to a maximum cactus for  $H_n$  to prove the following.

► **Theorem 16.** *Let  $H_n$  be the graph family as in Lemma 15. Then,  $\frac{\beta(H_n)}{f_3(H_n)} \leq \frac{1}{6} + o_n(1)$ .*

**Proof.** By Lemma 15, it suffices to argue that  $f_3(C_n) \geq \frac{\beta(H_n)}{2}$ . Let  $C_n^*$  be an optimal cactus with  $\beta(H_n)$  triangles. Notice that for any triangle  $t$  in  $C_n$ ,  $E(t)$  intersects at most two other triangles in  $C_n^*$ . If all three edges of  $t$  were to be used by three different triangles in  $C_n^*$ , this would contradict the cactus property. Moreover, if  $t$  does not intersect any triangle in  $C_n^*$  this would imply that one of its edges would complete a cycle if added to  $C_n^*$ . By these two observations we can use a simple counting scheme to upper-bound the number of triangles in  $C_n^*$  depending on the number of triangles in  $C_n$ . We iteratively add triangles of  $C_n$  to  $C_n^*$  and count in every step how many triangles in  $C_n^*$  need to be removed to maintain the cactus property. For every triangle in  $C_n$  that intersects  $C_n^*$  in one or two edges, we have to remove at most two triangles from  $C_n^*$ . For every triangle in  $C_n$ , that does not intersect  $C_n^*$  in any edge, we have to break a cycle in the resulting  $C_n^*$  by deleting one other triangle from it. In each iteration we therefore destroy at most two triangles from the original  $C_n^*$  and therefore get  $f_3(C_n^*) \leq 2f_3(C_n)$ . This concludes the proof as  $f_3(C_n) \geq f_3(C_n^*)/2 = \beta(H_n)/2$ . ◀

### 3.2 Comparison to the Previous Bound

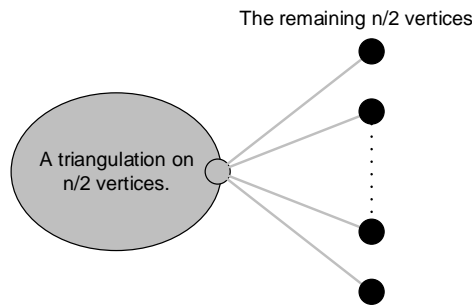
One integral part to derive the improved approximation ration for MPS in [3] was to show that for any given planar graph  $G = (V, E)$  with  $n = |V|$  vertices and  $|E| = 3n - 6 - t(G)$  edges, we have:

► **Theorem 17** ([3]). *Let  $G$  be as above, then  $\beta(G) \geq \frac{1}{3}(n - t(G) - 2)$ .*

As removing one edge from a triangulated planar graph merges exactly two faces, we can easily derive a lower bound that depends on  $t(G)$ , for the number of triangular faces in  $G$ :

$$f_3(G) \geq 2n - 2t(G) - 4$$

By Theorem 1, we have that  $\beta(G) \geq \frac{1}{6}f_3(G)$ . Combining these two facts implies Theorem 17.



■ **Figure 5** Bad example which shows that a extremal bound like the one in in [3] for MPS does not necessarily imply a similarly strong result to MPT.

We end this section by showing that the bound in [3] alone is not sufficient for approximating MPT. To this end we construct a graph in which  $\frac{1}{3}(n - t(G) - 2) \leq 0$ , even though  $f_3(G) = \Theta(n)$ , Let  $G$  be a planar graph with  $n$  vertices, where  $\frac{n}{2}$  vertices form a triangulated planar subgraph. Let  $v$  be a vertex on the outer-face of this triangulated structure. The remaining  $\frac{n}{2}$  vertices are embedded in the outer-face and are incident to exactly one edge each, with the other endpoint being  $v$  (see Figure 5 for an illustration

of this construction). Therefore by Euler's formula, the number of edges in this graph is equal to  $3\binom{n}{2} - 6 + \frac{n}{2} = 2n - 6$  and thus  $t(G) = n$ , while the number of triangular faces is  $f_3(G) = 2\binom{n}{2} - 4 - 1 = n - 5$ .

## 4 Conclusions and Open Problems

Our work implies that a natural local search algorithm gives a  $(\frac{4}{9} + \epsilon)$ -approximation for MPS and a  $\frac{1}{6} + \epsilon$  approximation for MPT. To be more precise, when given any graph  $G$ , we follow the  $t$ -swap local search strategy for  $t = O(1/\epsilon)$ : Start from any cactus subgraph  $H$ . Try to improve it by removing  $t$  triangles and adding  $(t + 1)$  triangles in a way that ensures that the graph remains a cactus subgraph. A local optimal solution will always be a  $(\frac{4}{9} + \epsilon)$  approximation for MPS and a  $(\frac{1}{6} + \epsilon)$  approximation for MPT.

Knowing this fact, there is an obvious candidate algorithm for improving over the long-standing best approximation factor for MPS. We call a graph  $H$  a diamond-cactus if every block in  $H$  is either a diamond<sup>5</sup> or a triangle. Start from any diamond-cactus subgraph  $H$  of  $G$  and then try to improve it by removing  $t$  triangles from  $H$  and adding  $(t + 1)$  triangles, maintaining the fact that  $H$  is a diamond-cactus subgraph. We conjectured that this algorithm gives a better than  $\frac{4}{9}$ -approximation for MPS, but we suspect that the analysis will require substantially new ideas.

Another interesting direction is to see whether there is a general principle that captures a denser planar structure than cactus subgraphs by going above matroid parity in the hierarchy of efficiently computable problems. For instance, are diamond-cactus subgraphs captured by matroid parity? Or can it be formulated as an even more abstract structure than matroids (e.g. commutative rank [2]) that can still be computed efficiently? We believe that studying this direction will lead to a better understanding of algebraic techniques for finding dense planar structures.

Finally, the absence of LP-based techniques in this problem domain seems rather unfortunate. There have been some experimental studies recently, but the theoretical understanding of what can be proven formally in the context of power of relaxation is certainly lacking [16, 10, 11]. Is there a convex relaxation that allows us to find a relatively dense planar subgraph (e.g.  $(3 - \epsilon)$ -approximation for MPS using LP-based techniques)?

---

## References

- 1 Claude Berge. *La theorie des graphes*. Paris, France, 1958.
- 2 Markus Bläser, Gorav Jindal, and Anurag Pandey. Greedy Strikes Again: A Deterministic PTAS for Commutative Rank of Matrix Spaces. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 33:1–33:16, 2017.
- 3 Gruia Călinescu, Cristina G Fernandes, Ulrich Finkler, and Howard Karloff. A better approximation algorithm for finding planar subgraphs. *Journal of Algorithms*, 27(2):269–302, 1998.
- 4 Gruia Calinescu, Cristina G Fernandes, Howard Karloff, and Alexander Zelikovsky. A new approximation algorithm for finding heavy planar subgraphs. *Algorithmica*, 36(2):179–205, 2003.
- 5 Gruia Călinescu, Cristina G Fernandes, Hemanshu Kaul, and Alexander Zelikovsky. Maximum series-parallel subgraph. *Algorithmica*, 63(1-2):137–157, 2012.

---

<sup>5</sup> A diamond subgraph is a graph that is isomorphic to the graph resulting from deleting any single edge from a  $K_4$ .

- 6 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-ETH to FPT-inapproximability: Clique, dominating set, and more. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 743–754. IEEE, 2017.
- 7 Parinya Chalermsook and Andreas Schmid. Finding Triangles for Maximum Planar Subgraphs. In *WALCOM: Algorithms and Computation, 11th International Conference and Workshops, (WALCOM'17), Proceedings.*, pages 373–384, 2017.
- 8 Parinya Chalermsook, Andreas Schmid, and Sumedha Uniyal. A Tight Extremal Bound on the Lovász Cactus Number in Planar Graphs. *CoRR*, abs/1804.03485, 2018. [arXiv:1804.03485](https://arxiv.org/abs/1804.03485).
- 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):10, 2014.
- 10 Markus Chimani, Ivo Hedtke, and Tilo Wiedera. Exact Algorithms for the Maximum Planar Subgraph Problem: New Models and Experiments. In *17th International Symposium on Experimental Algorithms, SEA 2018, June 27-29, 2018, L'Aquila, Italy*, pages 22:1–22:15, 2018. doi:10.4230/LIPIcs.SEA.2018.22.
- 11 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)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ESA.2018.19.
- 12 Harold N Gabow and Matthias Stallmann. An augmenting path algorithm for linear matroid parity. *Combinatorica*, 6(2):123–150, 1986.
- 13 Magnús M Halldórsson. Approximations of weighted independent set and hereditary subset problems. In *Graph Algorithms And Applications 2*, pages 3–18. World Scientific, 2004.
- 14 Johan Håstad. Clique is hard to approximate within  $1 - \epsilon$ . *Acta Mathematica*, 182(1):105–142, 1999.
- 15 T.A. Jenkyns. *Matchoids : a Generalization of Matchings and Matroids*. Thesis (Ph.D.)–University of Waterloo, 1974.
- 16 M. Jünger and P. Mutzel. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica*, 16(1):33–59, July 1996. doi:10.1007/BF02086607.
- 17 Subhash Khot and Ashok Kumar Ponnuswami. Better inapproximability results for maxclique, chromatic number and min-3lin-deletion. In *International Colloquium on Automata, Languages, and Programming*, pages 226–237. Springer, 2006.
- 18 Eugene L Lawler. *Combinatorial optimization: networks and matroids*. Courier Corporation, 1976.
- 19 Jon Lee, Maxim Sviridenko, and Jan Vondrák. Matroid matching: the power of local search. *SIAM Journal on Computing*, 42(1):357–379, 2013.
- 20 John M Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.
- 21 László Lovász. Matroid matching and some applications. *Journal of Combinatorial Theory, Series B*, 28(2):208–236, 1980.
- 22 László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.
- 23 Carsten Lund and Mihalis Yannakakis. The approximation of maximum subgraph problems. In *International Colloquium on Automata, Languages, and Programming*, pages 40–51. Springer, 1993.
- 24 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.
- 25 Zoltán Szigeti. On a min-max theorem of cacti. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 84–95. Springer, 1998.
- 26 William T Tutte. The factorization of linear graphs. *Journal of the London Mathematical Society*, 1(2):107–111, 1947.