

# An Algorithm for the Graph Crossing Number Problem

[Extended Abstract]

Julia Chuzhoy<sup>\*</sup>  
Toyota Technological Institute,  
Chicago, IL 60637  
cjulia@ttic.edu

## ABSTRACT

We study the Minimum Crossing Number problem: given an  $n$ -vertex graph  $G$ , the goal is to find a drawing of  $G$  in the plane with minimum number of edge crossings. This is one of the central problems in topological graph theory, that has been studied extensively over the past three decades. The first non-trivial efficient algorithm for the problem, due to Leighton and Rao, achieved an  $O(n \log^4 n)$ -approximation for bounded degree graphs. This algorithm has since been improved by poly-logarithmic factors, with the best current approximation ratio standing on  $O(n \cdot \text{poly}(d) \cdot \log^{3/2} n)$  for graphs with maximum degree  $d$ . In contrast, only APX-hardness is known on the negative side.

In this paper we present an efficient randomized algorithm to find a drawing of any  $n$ -vertex graph  $G$  in the plane with  $O(\text{OPT}^{10} \cdot \text{poly}(d \cdot \log n))$  crossings, where  $\text{OPT}$  is the number of crossings in the optimal solution, and  $d$  is the maximum vertex degree in  $G$ . This result implies an  $\tilde{O}(n^{9/10} \cdot \text{poly}(d))$ -approximation for Minimum Crossing Number, thus breaking the long-standing  $\tilde{O}(n)$ -approximation barrier for bounded-degree graphs.

## Categories and Subject Descriptors

F.2.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms and Problems*

## General Terms

Theory, Algorithms

## 1. INTRODUCTION

A drawing of a graph  $G$  in the plane is a mapping, in which every vertex of  $G$  is mapped into a point in the plane, and

<sup>\*</sup>Supported in part by NSF CAREER award CCF-0844872 and Sloan Research Fellowship.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

STOC'11, June 6–8, 2011, San Jose, California, USA.

Copyright 2011 ACM 978-1-4503-0691-1/11/06 ...\$10.00.

every edge into a continuous curve connecting the images of its endpoints. We assume that no three curves meet at the same point, and no curve contains an image of any vertex other than its endpoints. A *crossing* in such a drawing is a point where the images of two edges intersect, and the *crossing number* of a graph  $G$ , denoted by  $\text{OPT}_{\text{cr}}(G)$ , is the smallest number of crossings achievable by any drawing of  $G$  in the plane. The goal in the Minimum Crossing Number problem is to find a drawing of the input graph  $G$  with minimum number of crossings. We denote by  $n$  the number of vertices in  $G$ , and by  $d_{\text{max}}$  its maximum vertex degree.

The concept of the graph crossing number dates back to 1944, when Pál Turán has posed the question of determining the crossing number of the complete bipartite graph  $K_{m,n}$ . This question was motivated his work at a brick factory (see Turán's account in [29]). Later, Anthony Hill (see [14]) has posed the question of computing the crossing number of the complete graph  $K_n$ . Since then, the problem has become a subject of intense study, with hundreds of papers written on the subject (see, e.g. the extensive bibliography maintained by Vrt'o [30].) Despite this enormous stream of results and ideas, some of the most basic questions about the crossing number problem remain unanswered. For example, the crossing number of  $K_{11}$  was established just a few years ago ([25]), while the answer for  $K_t, t \geq 13$ , remains elusive. We note that in general  $\text{OPT}_{\text{cr}}(G)$  can be as large as  $\Omega(n^4)$ , for example for the complete graph. In particular, one of the famous results in this area, due to Ajtai et al. [1] and Leighton [20] states that if  $|E(G)| \geq 4n$ , then  $\text{OPT}_{\text{cr}}(G) = \Omega(|E(G)|^3/n^2)$ .

In this paper we focus on the algorithmic aspect of the problem. The first non-trivial algorithm for Minimum Crossing Number was obtained by Leighton and Rao [21], who combined their breakthrough result on balanced separators with the techniques of Bhatt and Leighton [4] for VLSI design, to obtain an algorithm that finds a drawing of any bounded-degree  $n$ -vertex graph with at most  $O(\log^4 n) \cdot (n + \text{OPT}_{\text{cr}}(G))$  crossings. This bound was later improved to  $O(\log^3 n) \cdot (n + \text{OPT}_{\text{cr}}(G))$  by Even, Guha and Schieber [10], and the new approximation algorithm of Arora, Rao and Vazirani [3] for Balanced Cut gives a further improvement to  $O(\log^2 n) \cdot (n + \text{OPT}_{\text{cr}}(G))$ , thus implying an  $O(n \cdot \log^2 n)$ -approximation for Minimum Crossing Number on bounded-degree graphs. This result can also be extended to general graphs with maximum vertex degree  $d_{\text{max}}$ , where the approximation factor becomes  $O(n \cdot \text{poly}(d_{\text{max}}) \cdot \log^2 n)$ . Chuzhoy, Makarychev and Sidiropoulos [9] have recently improved this result to an  $O(n \cdot \text{poly}(d_{\text{max}}) \cdot \log^{3/2} n)$ -approximation. On

the negative side, the problem was shown to be NP-complete by Garey and Johnson [11], and remains NP-complete even on cubic graphs [15]. More surprisingly, even in the very restricted case, where the input graph  $G$  is obtained by adding a single edge to a planar graph, the problem is still NP-complete [6]. The NP-hardness proof of [11], combined with the inapproximability result for Minimum Linear-Arrangement [2], implies that there is no PTAS for Minimum Crossing Number unless NP has randomized subexponential time algorithms.

To summarise, although current lower bounds do not rule out the possibility of a constant-factor approximation for the problem, the state of the art, prior to this work, only gives an  $\tilde{O}(n \cdot \text{poly}(d_{\max}))$ -approximation. In view of this glaring gap in our understanding of the problem, a natural question is whether we can obtain good algorithms for the case where the optimal solution cost is low — arguably, the most interesting setting for this problem. A partial answer was given by Grohe [13], who showed that the problem is fixed-parameter tractable. Specifically, Grohe designed an exact  $O(n^2)$ -time algorithm, for the case where the optimal solution cost is bounded by a constant. Later, Kawarabayashi and Reed [19] have shown a linear-time algorithm for the same setting. Unfortunately, the running time of both algorithms depends super-exponentially on the optimal solution cost.

Our main result is an efficient randomized algorithm, that, given as input any  $n$ -vertex graph with maximum degree  $d_{\max}$ , with high probability produces a drawing of  $G$  with  $O((\text{OPT}_{\text{cr}}(G))^{10} \cdot \text{poly}(d_{\max} \cdot \log n))$  crossings. In particular, we obtain an  $O(n^{9/10} \cdot \text{poly}(d_{\max} \cdot \log n))$ -approximation for general graphs, and an  $\tilde{O}(n^{9/10})$ -approximation for bounded-degree graphs, thus breaking the long standing barrier of  $\tilde{O}(n)$ -approximation for this setting.

We note that many special cases of the Minimum Crossing Number problem have been extensively studied, with better approximation algorithms known for some. Examples include  $k$ -apex graphs [8, 9], bounded genus graphs [5, 16, 12, 17, 9] and minor-free graphs [32]. Further overview of work on Minimum Crossing Number can be found in the expositions of Richter and Salazar [27], Pach and Tóth [24], Matoušek [23], and Székely [28].

### Our results and techniques.

Our main result is summarized in the following theorem.

**Theorem 1.1** *There is an efficient randomized algorithm, that, given any  $n$ -vertex graph  $G$  with maximum degree  $d_{\max}$ , with high probability finds a drawing of  $G$  in the plane with  $O((\text{OPT}_{\text{cr}}(G))^{10} \cdot \text{poly}(d_{\max} \cdot \log n))$  crossings.*

Combining this theorem with the algorithm of Even et al. [10], we obtain the following corollary.

**Corollary 1.1** *There is an efficient randomized approximation algorithm for Minimum Crossing Number, whose approximation factor is  $O(n^{9/10} \cdot \text{poly}(d_{\max} \cdot \log n))$ .*

We now give an overview of our techniques. Instead of directly solving the Minimum Crossing Number problem, it is more convenient to work with a closely related problem – Minimum Planarization. In this problem, given a graph  $G$ ,

the goal is to find a minimum-cardinality subset  $E^*$  of edges, such that the graph  $G \setminus E^*$  is planar. The two problems are closely related, and this connection was recently formalized by [9], in the following theorem:

**Theorem 1.2 ([9])** *Let  $G = (V, E)$  be any  $n$ -vertex graph of maximum degree  $d_{\max}$ , and suppose we are given a subset  $E^* \subseteq E$  of edges,  $|E^*| = k$ , such that  $G \setminus E^*$  is planar. Then there is an efficient algorithm to find a drawing of  $G$  in the plane with at most  $O(d_{\max}^3 \cdot k \cdot (\text{OPT}_{\text{cr}}(G) + k))$  crossings.*

Therefore, in order to solve the Minimum Crossing Number problem, it is sufficient to find a good solution to the Minimum Planarization problem on the same graph. We note that an  $O(\sqrt{n \log n} \cdot d_{\max})$ -approximation algorithm for the Minimum Planarization problem follows easily from the Planar Separator theorem of Lipton and Tarjan [22] (see e.g. [9]), and we are not aware of any other algorithmic results for the problem. Our main technical result is the proof of the following theorem, which, combined with Theorem 1.2, implies Theorem 1.1.

**Theorem 1.3** *There is an efficient randomized algorithm, that, given an  $n$ -vertex graph  $G = (V, E)$  with maximum degree  $d_{\max}$ , finds a subset  $E^* \subseteq E$  of edges, such that  $G \setminus E^*$  is planar, and with high probability*

$$|E^*| = O((\text{OPT}_{\text{cr}}(G))^5 \text{poly}(d_{\max} \cdot \log n)).$$

We now describe our main ideas and techniques. Given an optimal solution  $\varphi$  to the Minimum Crossing Number problem on graph  $G$ , we say that an edge  $e \in E(G)$  is *good* iff it does not participate in any crossings in  $\varphi$ . For convenience, we consider a slightly more general version of the problem, where, in addition to the graph  $G$ , we are given a simple cycle  $X \subseteq G$ , that we call the *bounding box*, and our goal is to find a drawing of  $G$ , such that the edges of  $X$  do not participate in any crossings, and all vertices and edges of  $G \setminus X$  appear on the same side of the closed curve to which  $X$  is mapped. In other words, if  $\gamma_X$  is the simple closed curve to which  $X$  is mapped, and  $F_1, F_2$  are the two faces into which  $\gamma_X$  partitions the plane, then one of the faces  $F \in \{F_1, F_2\}$  must contain the drawings of all the edges and vertices of  $G \setminus X$ . Such a drawing is called a *drawing of  $G$  inside the bounding box  $X$* . Since we allow  $X$  to be empty, this is indeed a generalization of the Minimum Crossing Number problem. In fact, from Theorem 1.2, it is enough to find what we call a *weak solution* to the problem, namely, a small-cardinality subset  $E^*$  of edges with  $E^* \cap E(X) = \emptyset$ , such that there is a planar drawing of the remaining graph  $G \setminus E^*$  inside the bounding box  $X$ . Our proof consists of three major ingredients that we describe below.

The algorithm is iterative. Throughout the algorithm, we gradually remove some edges from the graph, and gradually build a planar drawing of the remaining graph. One of the central notions we use is that of *graph skeletons*. A skeleton  $K$  of graph  $G$  is simply a sub-graph of  $G$ , that contains the bounding box  $X$ , and has a unique planar drawing (for example, it may be convenient to think of  $K$  as being 3-vertex connected). Given a skeleton  $K$ , and a small subset  $E'$  of edges (that we eventually remove from the graph), we say that  $K$  is an *admissible skeleton* iff all the edges of  $K$  are good, and each connected component of  $G \setminus (K \cup E')$  contains only a small number of vertices (say, at most  $(1 -$

$1/\rho)n$ , for some balance parameter  $\rho$ ). Since  $K$  has a unique planar drawing, and all of its edges are good, we can find its unique planar drawing efficiently, and it must be identical to the drawing  $\varphi_K$  of  $K$  induced by the optimal solution  $\varphi$ . Let  $\mathcal{F}$  be the set of faces in this drawing. Since  $K$  only contains good edges, for each connected component  $C$  of  $G \setminus (K \cup E')$ , all edges and vertices of  $C$  must be drawn completely inside one of the faces  $F_C \in \mathcal{F}$  in  $\varphi$ . Therefore, if, for each such connected component  $C$ , we can identify the face  $F_C$  inside which it needs to be embedded, then we can recursively solve the problems induced by each such component  $C$ , together with the bounding box formed by the boundary of  $F_C$ . In fact, given an admissible skeleton  $K$ , we show that we can find a good assignment of the connected components of  $G \setminus (K \cup E')$  to the faces of  $\mathcal{F}$ , so that, on the one hand, all resulting sub-problems have solutions of total cost at most  $\text{OPT}_{\text{cr}}(G)$ , while, on the other hand, if we combine weak solutions to these sub-problems with the set  $E'$  of edges, we obtain a feasible weak solution to the original problem. The assignment of the components to the faces of  $\mathcal{F}$  is done by reducing the problem to an instance of the Min-Uncut problem. We defer the details of this part to later sections, and focus here on finding an admissible skeleton  $K$ .

Our second main ingredient is the use of well-linked sets of vertices, and well-linked balanced bi-partitions. Given a set  $S$  of vertices, let  $G[S]$  be the sub-graph of  $G$  induced by  $S$ , and let  $\Gamma(S)$  be the subset of vertices of  $S$  adjacent to the edges in  $E(S, \bar{S})$ . Informally, we say that  $S$  is  $\alpha$ -well-linked, iff every pair of vertices in  $\Gamma(S)$  can send one flow unit to each other, with overall congestion bounded by  $\alpha|\Gamma(S)|$ . We say that a bi-partition  $(S, \bar{S})$  of the vertices of  $G$  is  $\rho$ -balanced and  $\alpha$ -well-linked, iff  $|S|, |\bar{S}| \geq n/\rho$ , and both  $S$  and  $\bar{S}$  are  $\alpha$ -well-linked. Suppose we can find a  $\rho$ -balanced,  $\alpha$ -well linked bi-partition of  $G$  (it is convenient to think of  $\rho, \alpha = \text{poly}(d_{\max} \cdot \log n)$ ). In this case, we show a randomized algorithm, that w.h.p. constructs an admissible skeleton  $K$ , as follows. Let  $\mathcal{P}, \mathcal{P}'$  be the collections of the flow-paths in  $G[S]$  and  $G[\bar{S}]$  respectively, guaranteed by the well-linkedness of  $S$  and  $\bar{S}$ . Since the congestion on all edges is relatively low, only a small number of paths in  $\mathcal{P} \cup \mathcal{P}'$  contain bad edges. Therefore, if we choose a random collection of paths from  $\mathcal{P}$  and  $\mathcal{P}'$  with appropriate probability, the resulting skeleton  $K$ , obtained from the union of these paths, is unlikely to contain bad edges. Moreover, we can show that w.h.p., every connected component of  $G \setminus K$  only contains a small number of edges in  $E(S, \bar{S})$ . It is still possible that some connected component  $C$  of  $G \setminus K$  contains many vertices of  $G$ . However, only one such component  $C$  may contain more than  $n/2$  vertices. Let  $E'$  be the subset of edges in  $E(S, \bar{S})$ , that belong to  $C$ . Then, since the original cut  $(S, \bar{S})$  is  $\rho$ -balanced, once we remove the edges of  $E'$  from  $C$ , it will decompose into small enough components. This will ensure that all connected components of  $G \setminus (K \cup E')$  are small enough, and  $K$  is admissible.

Using these ideas, given an efficient algorithm for computing  $\rho$ -balanced  $\alpha$ -well-linked cuts, we can obtain an algorithm for the Minimum Crossing Number problem. Unfortunately, we do not have an efficient algorithm for computing such cuts. We can only compute such cuts in graphs that do not contain a certain structure, that we call *nasty vertex sets*. Informally, a subset  $S$  of vertices is a nasty set, iff  $|S| \gg |E(S, \bar{S})|^2$ , and the sub-graph  $G[S]$  induced by

$S$  is planar. We show an algorithm, that, given any graph  $G$ , either produces a  $\rho$ -balanced  $\alpha$ -well linked cut, or finds a nasty set  $S$  in  $G$ . Therefore, if  $G$  does not contain any nasty sets, we can compute the  $\rho$ -balanced  $\alpha$ -well-linked bi-partition of  $G$ , and hence obtain an algorithm for Minimum Crossing Number. Moreover, given **any** graph  $G$ , if our algorithm fails to produce a good solution to the Minimum Crossing Number problem on  $G$ , then w.h.p. it returns a nasty set of vertices in  $G$ .

The third major component of our algorithm is handling the nasty sets. Suppose we are given a nasty set  $S$ , and assume for now that it is also  $\alpha$ -well-linked for some parameter  $\alpha = \text{poly}(\log n)$ . Let  $\Gamma(S)$  denote the endpoints of the edges in  $E(S, \bar{S})$  that belong to  $S$ , and let  $|\Gamma(S)| = z$ . Recall that  $|S| \gg z^2$ , and  $G[S]$  is planar. Intuitively, in this case we can use the  $z \times z$  grid to “simulate” the sub-graph  $G[S]$ . More precisely, we replace the sub-graph  $G[S]$  with the  $z \times z$  grid  $Z_S$ , and identify the vertices of the first row of the grid with the vertices in  $\Gamma(S)$ . We call the resulting graph the *contracted graph*, and denote it by  $G_{|S}$ . Notice that the number of vertices in  $G_{|S}$  is smaller than that in  $G$ . When  $S$  is not well-linked, we perform a simple well-linked decomposition procedure to partition  $S$  into a collection of well-linked subsets, and replace each one of them with a grid separately. Given a drawing of the resulting contracted graph  $G_{|S}$ , we say that it is a *canonical drawing* if the edges of the newly added grids do not participate in any crossings. Similarly, we say that a planarizing subset  $E^*$  of edges is a weak canonical solution for  $G_{|S}$ , iff the edges of the grids do not belong to  $E^*$ . We show that the crossing number of  $G_{|S}$  is bounded by  $\text{poly}(d_{\max} \cdot \log n)\text{OPT}_{\text{cr}}(G)$ , and this bound remains true even for canonical drawings. On the other hand, we show that given any weak canonical solution  $E^*$  for  $G_{|S}$ , we can efficiently find a weak solution of comparable cost for  $G$ . Therefore, it is enough to find a weak feasible canonical solution for graph  $G_{|S}$ . However, even the contracted graph  $G_{|S}$  may still contain nasty sets. We then show that, given any nasty set  $S'$  in  $G_{|S}$ , we can find another subset  $S''$  of vertices in the original graph  $G$ , such that the contracted graph  $G_{|S''}$  contains fewer vertices than  $G_{|S}$ . The crossing number of  $G_{|S''}$  is again bounded by  $\text{poly}(d_{\max} \cdot \log n)\text{OPT}_{\text{cr}}(G)$  even for canonical drawings, and a weak canonical solution to  $G_{|S''}$  gives a weak solution to  $G$  as before.

Our algorithm then consists of a number of stages. In each stage, it starts with the current contracted graph  $G_{|S}$  (where in the first stage,  $S = \emptyset$ , and  $G_{|S} = G$ ). It then either finds a good weak canonical solution for problem  $G_{|S}$ , thus giving a feasible solution to the original problem, or returns a nasty set  $S'$  in graph  $G_{|S}$ . We then construct a new contracted graph  $G_{|S''}$ , that contains fewer vertices than  $G_{|S}$ , and becomes the input to the next stage.

### Organization.

We start with some basic definitions, notation, and general results on cuts and flows in Section 2. We then present a more detailed algorithm overview in Section 3. Section 4 is devoted to the graph contraction step, and the rest of the algorithm is outlined in Section 5. Our conclusions appear in Section 6. Due to lack of space, we are unable to provide a detailed description of the algorithm and the proof of its correctness. A full version of the paper, containing the detailed algorithm description and correctness proof is

## 2. PRELIMINARIES AND NOTATION

In order to avoid confusion, throughout the paper, we denote the input graph by  $\mathbf{G}$ , with  $|V(\mathbf{G})| = n$ , and maximum vertex degree  $d_{\max}$ . In statements regarding general arbitrary graphs, we will denote them by  $G$ , to distinguish them from the specific graph  $\mathbf{G}$ .

### 2.1 General Notation

We use the words “drawing” and “embedding” interchangeably. Given any graph  $G$ , a drawing  $\varphi$  of  $G$ , and any sub-graph  $H$  of  $G$ , we denote by  $\varphi_H$  the drawing of  $H$  induced by  $\varphi$ , and by  $\text{cr}_\varphi(G)$  the number of crossings in the drawing  $\varphi$  of  $G$ . Notice that we can assume w.l.o.g. that no edge crosses itself in any drawing. For any pair  $E_1, E_2 \subseteq E(G)$  of subsets of edges, we denote by  $\text{cr}_\varphi(E_1, E_2)$  the number of crossings in  $\varphi$  in which the images of the edges of  $E_1$  intersect the images of the edges of  $E_2$ , and by  $\text{cr}_\varphi(E_1)$  the number of crossings in  $\varphi$  in which the images of the edges of  $E_1$  intersect with each other. Given two disjoint sub-graphs  $H_1, H_2$  of  $G$ , we will sometimes write  $\text{cr}_\varphi(H_1, H_2)$  instead of  $\text{cr}_\varphi(E(H_1), E(H_2))$ , and  $\text{cr}_\varphi(H_1)$  instead of  $\text{cr}_\varphi(E(H_1))$ . If  $G$  is a planar graph, and  $\varphi$  is a drawing of  $G$  with no crossings, then we say that  $\varphi$  is a *planar* drawing of  $G$ . For a graph  $G = (V, E)$ , and subsets  $V' \subseteq V$ ,  $E' \subseteq E$  of its vertices and edges respectively, we denote by  $G[V']$ ,  $G \setminus V'$ , and  $G \setminus E'$  the sub-graphs of  $G$  induced by  $V'$ ,  $V \setminus V'$ , and  $E \setminus E'$ , respectively.

**Definition 2.1** Let  $\gamma$  be any closed simple curve, and let  $F_1, F_2$  be the two faces into which  $\gamma$  partitions the plane. Given any drawing  $\varphi$  of a graph  $G$ , we say that  $G$  is embedded inside  $\gamma$ , iff one of the faces  $F \in \{F_1, F_2\}$  contains the images of all edges and vertices of  $G$  (the images of the vertices of  $G$  may lie on  $\gamma$ ). Similarly, if  $C \subseteq G$  is a simple cycle, then we say that  $G$  is embedded inside  $C$ , iff the edges of  $C$  do not participate in any crossings, and  $G \setminus E(C)$  is embedded inside  $\gamma_C$  – the simple closed curve to which  $C$  is mapped.

Given a graph  $G$  and a bounding box  $X$ , we define the problem  $\pi(G, X)$ , that we use extensively.

**Definition 2.2** Given a graph  $G$  and a simple (possibly empty) cycle  $X \subseteq G$ , called the bounding box, a strong solution for problem  $\pi(G, X)$ , is a drawing  $\psi$  of  $G$ , in which  $G$  is embedded inside the bounding box  $X$ , and its cost is the number of crossings in  $\psi$ . A weak solution to problem  $\pi(G, X)$  is a subset  $E' \subseteq E(G) \setminus E(X)$  of edges, such that  $G \setminus E'$  has a planar drawing, in which it is embedded inside the bounding box  $X$ .

Notice that in order to prove Theorem 1.3, it is enough to find a weak solution for problem  $\pi(\mathbf{G}, X_0)$ , where  $X_0 = \emptyset$ , of cost  $O((\text{OPT}_{\text{cr}}(\mathbf{G}))^5 \text{poly}(d_{\max} \cdot \log n))$ .

**Definition 2.3** For any graph  $G = (V, E)$ , a subset  $V' \subseteq V$  of vertices is called a  $c$ -separator, iff  $|V'| = c$ , and the graph  $G \setminus V'$  is not connected. We say that  $G$  is  $c$ -connected iff it does not contain  $c'$ -separators, for any  $0 < c' < c$ .

We will use the following three well-known results:

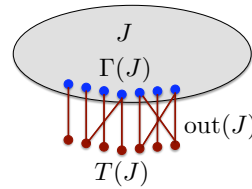
**Theorem 2.1** (Whitney [31]) Every 3-connected planar graph has a unique planar drawing.

**Theorem 2.2** (Hopcroft-Tarjan [18]) For any graph  $G$ , there is an efficient algorithm to determine whether  $G$  is planar, and if so, to find a planar drawing of  $G$ .

**Theorem 2.3** (Ajtai et al. [1], Leighton [20]) Let  $G$  be any graph with  $n$  vertices and  $m \geq 4n$  edges. Then  $\text{OPT}_{\text{cr}}(G) = \Omega(m^3/n^2) = \Omega(n)$ .

### 2.2 Well-linkedness

**Definition 2.4** Let  $G = (V, E)$  be any graph, and  $J \subseteq V$  any subset of its vertices. We denote by  $\text{out}_G(J) = E_G(J, V \setminus J)$ , and we call the edges in  $\text{out}_G(J)$  the terminal edges for  $J$ . For each terminal edge  $e = (u, v)$ , with  $u \in J$ ,  $v \notin J$ , we call  $u$  the interface vertex and  $v$  the terminal vertex for  $J$ . We denote by  $\Gamma_G(J)$  and  $T_G(J)$  the sets of all interface and terminal vertices for  $J$ , respectively, and we omit the subscript  $G$  when clear from context (see Figure 2.1).



**Figure 2.1:** Terminal edges, and terminal and interface vertices for set  $J$ .

**Definition 2.5** Given a graph  $G$ , a subset  $J$  of its vertices, and a parameter  $\alpha > 0$ , we say that  $J$  is  $\alpha$ -well-linked, iff for any partition  $(J_1, J_2)$  of  $J$ , if we denote by  $T_1 = \text{out}(J_1) \cap \text{out}(J)$ , and by  $T_2 = \text{out}(J_2) \cap \text{out}(J)$ , then  $|E(J_1, J_2)| \geq \alpha \cdot \min\{|T_1|, |T_2|\}$

Notice that if  $G$  is a connected graph and  $J \subset V(G)$  is  $\alpha$ -well-linked for any  $\alpha > 0$ , then  $G[J]$  must be connected. Finally, we define  $\rho$ -balanced  $\alpha$ -well-linked bi-partitions.

**Definition 2.6** Let  $G$  be any graph, and let  $\rho > 1, 0 < \alpha \leq 1$  be any parameters. We say that a bi-partition  $(S, \bar{S})$  of  $V(G)$  is  $\rho$ -balanced and  $\alpha$ -well-linked, iff  $|S|, |\bar{S}| \geq |V(G)|/\rho$  and both  $S$  and  $\bar{S}$  are  $\alpha$ -well-linked.

### 2.3 Canonical Vertex Sets and Solutions

As already mentioned in the Introduction, we will perform a number of graph contraction steps on the input graph  $\mathbf{G}$ , where in each such graph contraction step, a sub-graph of  $\mathbf{G}$  will be replaced with a grid. So in general, if  $H$  is the current graph, we will also be given a collection  $\mathcal{Z}$  of disjoint subsets of vertices of  $H$ , such that for each  $Z \in \mathcal{Z}$ ,  $H[Z]$  is the  $k_Z \times k_Z$  grid, for some  $k_Z \geq 2$ . We will also ensure that  $\Gamma_H(Z)$  is precisely the set of the vertices in the first row of the grid  $H[Z]$ , and the edges in  $\text{out}_H(Z)$  form a

matching between  $\Gamma_H(Z)$  and  $T_H(Z)$ . Given such a graph  $H$ , and a collection  $\mathcal{Z}$  of vertex subsets, we will be looking for solutions in which the edges of the grids  $H[Z]$  do not participate in any crossings. This motivates the following definitions of canonical vertex sets and canonical solutions.

Assume that we are given a graph  $G$  and a collection  $\mathcal{Z}$  of disjoint subsets of vertices of  $G$ , (but some vertices of  $G$  may not belong to any subset  $Z \in \mathcal{Z}$ ).

**Definition 2.7** *We say that a subset  $J \subseteq V$  of vertices is canonical for  $\mathcal{Z}$  iff for each  $Z \in \mathcal{Z}$ , either  $Z \subseteq J$ , or  $Z \cap J = \emptyset$ .*

We next define canonical drawings and canonical solutions w.r.t. the collection  $\mathcal{Z}$  of subsets of vertices:

**Definition 2.8** *Let  $G = (V, E)$  be any graph, and  $\mathcal{Z}$  any collection of disjoint subsets of vertices of  $G$ . We say that a drawing  $\varphi$  of  $G$  is canonical for  $\mathcal{Z}$  iff for each  $Z \in \mathcal{Z}$ , no edge of  $G[Z]$  participates in crossings. Similarly, we say that a solution  $E^*$  to the Minimum Planarization problem on  $G$  is canonical for  $\mathcal{Z}$ , iff for each  $Z \in \mathcal{Z}$ , no edge of  $G[Z]$  belongs to  $E^*$ .*

**Definition 2.9** *Given a graph  $G$ , a simple cycle  $X \subseteq G$  (that may be empty), and a collection  $\mathcal{Z}$  of disjoint subsets of vertices of  $G$ , a strong solution to problem  $\pi(G, X, \mathcal{Z})$  is a drawing  $\psi$  of  $G$ , in which the edges of  $E(X) \cup (\bigcup_{Z \in \mathcal{Z}} E(G[Z]))$  do not participate in any crossings, and  $G$  is embedded inside the bounding box  $X$ . The cost of the solution is the number of edge crossings in  $\psi$ . A weak solution to problem  $\pi(G, X, \mathcal{Z})$  is a subset  $E' \subseteq E(G) \setminus E(X)$  of edges, such that graph  $G \setminus E'$  has a planar drawing inside the bounding box  $X$ , and for all  $Z \in \mathcal{Z}$ ,  $E' \cap E(G[Z]) = \emptyset$ .*

We will sometimes use the above definition for problem  $\pi(G', X, \mathcal{Z})$ , where  $G'$  is a sub-graph of  $G$ . That is, some sets  $Z \in \mathcal{Z}$  may not be contained in  $G'$ , or only partially contained in it. We can then define  $\mathcal{Z}'$  to contain, for each  $Z \in \mathcal{Z}$ , the set  $Z \cap V(G')$ . We will sometimes use the notion of weak or strong solution to problem  $\pi(G', X, \mathcal{Z})$  to mean weak or strong solutions to  $\pi(G', X, \mathcal{Z}')$ , to simplify notation.

## 2.4 Well-linked Decompositions

The next theorem summarizes well-linked decomposition of graphs, which has been used extensively in graph decomposition (e.g., see [7, 26]).

**Theorem 2.4 (Well-linked decomposition)** *Given any graph  $G = (V, E)$ , and any subset  $J \subseteq V$  of vertices, we can efficiently find a partition  $\mathcal{J}$  of  $J$ , such that each set  $J' \in \mathcal{J}$  is  $\alpha^*$ -well-linked for  $\alpha^* = \Omega(1/(\log^{3/2} n \log \log n))$ , and  $\sum_{J' \in \mathcal{J}} |\text{out}(J')| \leq 2|\text{out}(J)|$ .*

We now define some additional properties that set  $J$  may possess, that we use throughout the paper. We will then show that if a set  $J$  has any collection of these properties, then we can find a well-linked decomposition  $\mathcal{J}$  of  $J$ , such that every set  $J' \in \mathcal{J}$  has these properties as well.

**Definition 2.10** *Given a graph  $G$  and any subset  $J \subseteq V(G)$  of its vertices, we say that  $J$  has property (P1) iff the vertices of  $T(J)$  are connected in  $G \setminus J$ . We say that it has*

*property (P2) iff there is a planar drawing of  $J$  in which all interface vertices  $\Gamma(J)$  lie on the boundary of the same face, that we refer to as the outer face. We denote such a planar drawing by  $\pi(J)$ . If there are several such drawing, we select any of them arbitrarily.*

The next theorem is an extension of Theorem 2.4, and its proof is omitted due to lack of space.

**Theorem 2.5** *Suppose we are given any graph  $G = (V, E)$ , a subset  $J \subseteq V$  of vertices, and a collection  $\mathcal{Z}$  of disjoint subsets of vertices of  $V$ , such that each set  $Z \in \mathcal{Z}$  is 1-well-linked. Then we can efficiently find a partition  $\mathcal{J}$  of  $J$ , such that each set  $J' \in \mathcal{J}$  is  $\alpha^*$ -well linked for  $\alpha^* = \Omega(1/(\log^{3/2} n \log \log n))$ , and  $\sum_{J' \in \mathcal{J}} |\text{out}(J')| \leq 2|\text{out}(J)|$ . Moreover, if  $J$  has any combination of the following three properties: (1) property (P1); (2) property (P2); (3) it is a canonical set for  $\mathcal{Z}$ , then each set  $J' \in \mathcal{J}$  will also have the same combination of these properties.*

Throughout the paper, we use  $\alpha^*$  to denote the parameter from Theorem 2.5.

## 3. HIGH LEVEL ALGORITHM OVERVIEW

In this section we provide a high-level overview of the algorithm. We start by defining the notion of nasty vertex sets.

**Definition 3.1** *Given a graph  $G$ , we say that a subset  $S \subseteq V(G)$  of vertices is nasty iff it has properties (P1) and (P2), and  $|S| \geq \frac{2^{16} \cdot d_{\max}^6}{(\alpha^*)^2} \cdot |\Gamma(S)|^2$ , where  $\alpha^*$  is the parameter from Theorem 2.5.*

Note that we do not require that  $G[S]$  is connected.

For the sake of clarity, let us first assume that the input graph  $\mathbf{G}$  contains no nasty sets. Fix any optimal drawing  $\varphi$  of  $\mathbf{G}$ . An edge of  $\mathbf{G}$  is called good iff it does not participate in any crossings in  $\varphi$ . Our algorithm then proceeds as follows. We use a balancing parameter  $\rho = O(\text{OPT}_{\text{cr}}(\mathbf{G}) \cdot \text{poly}(d_{\max} \cdot \log n))$  whose exact value is set later. The algorithm has  $O(\rho \cdot \log n)$  iterations. At the beginning of each iteration  $h$ , we are given a collection  $G_1, \dots, G_{k_h}$  of  $k_h \leq \text{OPT}_{\text{cr}}(\mathbf{G})$  sub-graphs of  $\mathbf{G}$ , together with bounding boxes  $X_i \subseteq G_i$  for all  $i$ . The graphs  $G_i$  are completely disjoint, except that they may share vertices or edges that belong to their bounding boxes. We are also guaranteed that w.h.p., there is a strong solution to each problem  $\pi(G_i, X_i)$ , of total cost at most  $\text{OPT}_{\text{cr}}(\mathbf{G})$ . In the first iteration,  $k_1 = 1$ , and the only graph is  $G_1 = \mathbf{G}$ , whose bounding box is  $X_0 = \emptyset$ .

We now proceed to describe each iteration. The idea is to find a *skeleton*  $K_i$  for each graph  $G_i$ , with  $X_i \subseteq K_i$ , such that  $K_i$  only contains good edges — that is, edges that do not participate in any crossings in the fixed optimal solution  $\varphi$ , and  $K_i$  has a unique planar drawing, in which  $X_i$  serves as the bounding box. Therefore, we can efficiently find the drawing  $\varphi_{K_i}$  of the skeleton  $K_i$ , induced by the optimal drawing  $\varphi$ . We then decompose the remaining graph  $G_i \setminus E(K_i)$  into *clusters*, by removing a small subset of edges from it, so that, on the one hand, for each such cluster  $C$ , we know the face  $F_C$  of  $\varphi_{K_i}$  where we should embed it, while on the other hand, different clusters  $C, C'$  do not interfere with each other, in the sense that we can find an embedding

of each one of these clusters separately, and their embeddings do not affect each other. For each such cluster  $C$ , we then define a new problem  $\pi(C, \gamma(F_C))$ , where  $\gamma(F_C)$  is the boundary of the face  $F_C$ . We will ensure that all resulting sub-problems have strong solutions whose total cost is at most  $\text{OPT}_{\text{cr}}(\mathbf{G})$ . In particular, there are at most  $\text{OPT}_{\text{cr}}(G)$  resulting sub-problems, for which  $\emptyset$  is not a feasible weak solution. Therefore, in the next iteration we will need to solve at most  $\text{OPT}_{\text{cr}}(G)$  new sub-problems. The main challenge is to find, for each graph  $G_i$ , a skeleton  $K_i$ , such that the number of vertices in each resulting cluster  $C$  is bounded by roughly  $(1 - 1/\rho)|V(G_i)|$ , so that the number of iterations is indeed bounded by  $O(\rho \log n)$ . We need this bound on the number of iterations, since the probability of successfully constructing the skeletons in each iteration is only  $(1 - 1/\rho)$ . Roughly speaking, we are able to build the skeleton as required, if we can find a  $\rho$ -balanced  $\alpha$ -well-linked bipartition of the vertices of  $G_i$ , where  $\alpha = 1/\text{poly}(d_{\max} \cdot \log n)$ . We are only able to find such a partition if no nasty sets exist in  $\mathbf{G}$ . More precisely, we show an efficient algorithm, that either finds the desired bi-partition, or returns a nasty vertex set.

In order to obtain the whole algorithm, we therefore need to deal with nasty sets. We do so by performing a graph contraction step, which is formally defined in the next section. Informally, given a nasty set  $S$ , we find a partition  $\mathcal{X}$  of  $S$ , such that for every pair  $X, X' \in \mathcal{X}$ , the graphs  $\mathbf{G}[X], \mathbf{G}[X']$  share at most one vertex, and no edges. The unique vertex  $v \in X \cap X'$  must belong to the interface of both  $X$  and  $X'$ . Each such set  $X$  is also  $\alpha^*$ -well-linked, has properties (P1) and (P2), and  $\sum_{X \in \mathcal{X}} |\Gamma(X)| \leq O(|\Gamma(S)|)$ . We then replace each sub-graph  $\mathbf{G}[X]$  of  $\mathbf{G}$  by a grid  $Z_X$ , whose interface is  $\Gamma(X)$ . After we do so for each  $X \in \mathcal{X}$ , we denote by  $\mathbf{G}_{|S}$  the resulting contracted graph. Notice that we have replaced  $\mathbf{G}[S]$  by a much smaller graph, whose size is bounded by  $O(|\Gamma(S)|^2)$ . Let  $\mathcal{Z}$  denote the collection of sets  $V(Z_X)$  of vertices, for  $X \in \mathcal{X}$ . We then show that the cost of the optimal solution to problem  $\pi(\mathbf{G}_{|S}, \emptyset, \mathcal{Z})$  is at most  $\text{poly}(d_{\max} \cdot \log n) \text{OPT}_{\text{cr}}(G)$ . Therefore, we can restrict our attention to canonical solutions only. We also show that it is enough to find a weak solution to problem  $\pi(\mathbf{G}_{|S}, \emptyset, \mathcal{Z})$ , in order to obtain a weak solution for the whole graph  $\mathbf{G}$ . Unfortunately, we do not know how to find a nasty set  $S$ , such that the corresponding contracted graph  $\mathbf{G}_{|S}$  contains no nasty sets. Instead, we do the following. Let  $H = \mathbf{G}_{|S}$  be the current graph, which is a result of the graph contraction step on some set  $S$  of vertices, and let  $\mathcal{Z}$  be the corresponding collection of sub-sets of vertices representing the grids. Suppose we can find a nasty canonical set  $R$  in the graph  $H$ . We show that this allows us to find a new set  $S'$  of vertices in  $\mathbf{G}$ , such that the contracted graph  $\mathbf{G}_{|S'}$  contains fewer vertices than  $\mathbf{G}_{|S}$ .

Returning to our algorithm, let  $\mathbf{G}_{|S}$  be the current contracted graph. We show that with high probability, the algorithm either returns a weak solution for  $\mathbf{G}_{|S}$  of cost  $O((\text{OPT}_{\text{cr}}(G))^5 \text{poly}(d_{\max} \cdot \log n))$ , or it returns a nasty canonical subset  $S'$  of  $\mathbf{G}_{|S}$ . In the former case, we can recover a good weak solution for the original graph  $\mathbf{G}$ . In the latter case, we find a subset  $S''$  of vertices in the original graph  $\mathbf{G}$ , and perform another contraction step on  $\mathbf{G}$ , obtaining a new graph  $\mathbf{G}_{|S''}$ , whose size is strictly smaller than that of  $\mathbf{G}_{|S}$ . We then apply the algorithm to graph  $\mathbf{G}_{|S''}$ . Since the total number of graph contraction steps is bounded by  $n$ , after  $n$  such iterations, we are guaranteed w.h.p. to obtain a weak

feasible solution of cost  $O((\text{OPT}_{\text{cr}}(G))^5 \text{poly}(d_{\max} \cdot \log n))$  to  $\pi(\mathbf{G}, \emptyset)$ , thus satisfying the requirements of Theorem 1.3. We now turn to formal description of the algorithm. One of the main ingredients is the graph contraction step, summarized in the next section.

## 4. GRAPH CONTRACTION STEP

The input to the graph contraction step consists of the input graph  $\mathbf{G}$ , and a subset  $S \subseteq V(\mathbf{G})$  of vertices, for which properties (P1) and (P2) hold. It will be convenient to think of  $S$  as a nasty set, but we do not require it.

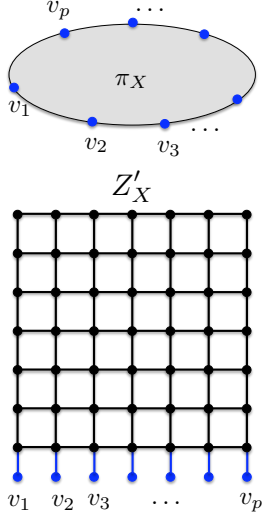
Let  $\mathcal{C} = \{G_1, \dots, G_q\}$  be the set of all connected components of  $\mathbf{G}[S]$ . For each  $1 \leq i \leq q$ , let  $\Gamma_i = V(G_i) \cap \Gamma(S) = \Gamma(V(G_i))$  be the set of the interface vertices of  $G_i$ . The goal of the graph contraction step is to find, for each  $1 \leq i \leq q$ , a partition  $\mathcal{X}_i$  of the set  $V(G_i)$ , that has the following properties. Let  $\mathcal{X} = \bigcup_{i=1}^q \mathcal{X}_i$ .

- C1. Each set  $X \in \mathcal{X}$  is  $\alpha^*$ -well-linked, and has properties (P1) and (P2). Moreover, there is a planar drawing  $\pi'(X)$  of  $\mathbf{G}[X]$ , and a simple closed curve  $\gamma_X$ , such that  $\mathbf{G}[X]$  is embedded inside  $\gamma_X$  in  $\pi'(X)$ , and the vertices of  $\Gamma(X)$  lie on  $\gamma_X$ .
- C2. For each  $X \in \mathcal{X}$ , either  $|\Gamma(X)| = 2$ , or there is a partition  $(C_X^*, R_1, \dots, R_t)$  of  $X$ , such that  $\mathbf{G}[C_X^*]$  is 2-connected and  $\Gamma(X) \subseteq C_X^*$ . Moreover, for each  $1 \leq t' \leq t$ , there is a vertex  $u_{t'} \in C_X^*$ , whose removal from  $\mathbf{G}[X]$  separates the vertices of  $R_{t'}$  from the remaining vertices of  $X$ .
- C3. For each pair  $X, X' \in \mathcal{X}$ , the two sets of vertices are completely disjoint, except for possibly sharing one interface vertex,  $v \in \Gamma(X) \cap \Gamma(X')$ .
- C4. For each  $1 \leq i \leq q$ , if  $\Gamma'_i = \bigcup_{X \in \mathcal{X}_i} \Gamma(X)$ , then  $|\Gamma'_i| \leq 9|\Gamma_i|$ .
- C5. For each  $X \in \mathcal{X}$ ,  $|X| \geq (\alpha^*|\Gamma(X)|)^2 / 64d_{\max}^2$ .

For each set  $X \in \mathcal{X}$ , we now define a new graph  $Z'_X$ , that will eventually replace the sub-graph  $\mathbf{G}[X]$  in  $\mathbf{G}$ . Intuitively, we need  $Z'_X$  to contain the vertices of  $\Gamma(X)$  and to be 1-well-linked w.r.t. these vertices. We also need it to have a unique planar embedding where the vertices of  $\Gamma(X)$  lie on the boundary of the same face, and finally, we need the size of the graph  $Z'_X$  to be relatively small, since this is a graph contraction step. The simplest graph satisfying these properties is a grid of size  $|\Gamma(X)| \times |\Gamma(X)|$ .

Specifically, we first define a graph  $Z_X$  as follows: if  $|\Gamma_X| = 1$ , then  $Z_X$  consists of a single vertex, and if  $|\Gamma_X| = 2$ , then  $Z_X$  consists of a single edge. Otherwise,  $Z_X$  is a grid of size  $|\Gamma(X)| \times |\Gamma(X)|$ . In order to obtain the graph  $Z'_X$ , we add the set  $\Gamma(X)$  of vertices to  $Z_X$ , and add a matching between the vertices of the first row of the grid and the vertices of  $\Gamma(X)$ . This is done so that the order of the vertices of  $\Gamma(X)$  along the first row of the grid is the same as their order along the curve  $\gamma_X$  in the drawing  $\pi'(X)$ . We refer to these new edges as the *matching edges*. For the cases where  $|\Gamma_X| = 1$  and  $|\Gamma_X| = 2$ , we obtain  $Z'_X$  by adding the vertices of  $\Gamma(X)$  to  $Z_X$ , and adding an arbitrary matching between  $\Gamma_X$  and the vertices of  $Z_X$ . (See Figure 4.1).

The contracted graph  $\mathbf{G}_{|S}$  is obtained from  $\mathbf{G}$ , by replacing, for each  $X \in \mathcal{X}$ , the subgraph  $\mathbf{G}[X]$  of  $\mathbf{G}$ , with the



(a) General case

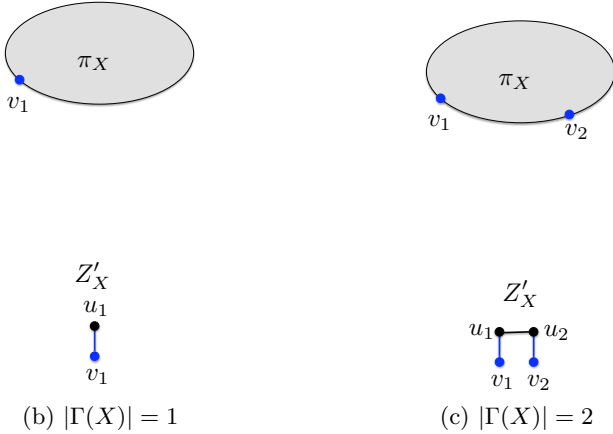


Figure 4.1: Graph  $Z'_X$ .

graph  $Z'_X$ . This is done as follows: first, delete all vertices and edges of  $\mathbf{G}[X]$ , except for the vertices of  $\Gamma(X)$ , from  $\mathbf{G}$ , and add the edges and the vertices of  $Z'_X$  instead. Next, identify the copies of the interface vertices  $\Gamma[X]$  in the two graphs. Let  $H = \mathbf{G}_{|S}$  denote the resulting contracted graph. Notice that

$$\begin{aligned}
 \sum_{i=1}^q \sum_{X \in \mathcal{X}_i} |V(Z'_X)| &\leq \sum_{i=1}^q \sum_{X \in \mathcal{X}_i} 2|\Gamma(X)|^2 \\
 &\leq \sum_{i=1}^q 2|\Gamma'_i|^2 d_{\max}^2 \\
 &\leq 162d_{\max}^2 |\Gamma|^2
 \end{aligned} \tag{4.1}$$

(we have used the fact that a vertex may belong to the interface of at most  $d_{\max}$  sets  $X \in \mathcal{X}_i$ , and Property (C4)).

Therefore, if the initial vertex set  $S$  is nasty, then we have indeed reduced the graph size, as  $|V(H)| < |V(\mathbf{G})|$ .

We now define a collection  $\mathcal{Z}$  of subsets of vertices of  $H$ , as follows:  $\mathcal{Z} = \{V(Z_X) \mid X \in \mathcal{X}\}$ . Notice that these sets are completely disjoint, as  $Z_X$  does not contain the interface vertices  $\Gamma(X)$ . Moreover, for each  $Z \in \mathcal{Z}$ ,  $H[Z]$  is a grid,  $\Gamma_H(Z)$  consists of the vertices in the first row of the grid, and  $\text{out}_H(Z)$  consists of the set of the matching edges, each of which connects a vertex in the first row of the grid  $Z$  to a distinct vertex in  $T_H(Z)$ . Using Definitions 2.7 and 2.8, we can now define canonical subsets of vertices, canonical drawings and canonical solutions to the Minimum Planarization problem on  $H$ , with respect to  $\mathcal{Z}$ . Our main result for graph contraction is summarized in the next theorem, whose proof is omitted due to lack of space.

**Theorem 4.1** *Let  $S \subseteq V(\mathbf{G})$  be any subset of vertices with properties (P1) and (P2), and let  $\{G_1, \dots, G_q\}$  be the set of all connected components of graph  $\mathbf{G}[S]$ . Then for each  $1 \leq i \leq q$ , we can efficiently find a partition  $\mathcal{X}_i$  of  $V(G_i)$ , such that the resulting partition  $\mathcal{X} = \bigcup_{i=1}^q \mathcal{X}_i$  of  $S$  has properties (C1)–(C5). Moreover, there is a canonical drawing of the resulting contracted graph  $H = \mathbf{G}_{|S}$  with  $O(d_{\max}^9 \cdot \log^{10} n \cdot (\log \log n)^4 \cdot \text{OPT}_{\text{cr}}(\mathbf{G}))$  crossings.*

The next claim shows, that in order to find a good solution to the Minimum Planarization problem on  $\mathbf{G}$ , it is enough to solve it on  $\mathbf{G}_{|S}$ .

**Claim 4.2** *Let  $S$  be any subset of vertices of  $\mathbf{G}$ ,  $\mathcal{X}$  any partition of  $S$  with properties (C1)–(C5),  $H = \mathbf{G}_{|S}$  the corresponding contracted graph and  $\mathcal{Z}$  the collection of grids  $Z_X$  for  $X \in \mathcal{X}$ . Then given any canonical solution  $E^*$  to the Minimum Planarization problem on  $H$ , we can efficiently find a solution of cost  $O(d_{\max})|E^*|$  to Minimum Planarization on  $\mathbf{G}$ .*

**PROOF.** Partition set  $E^*$  of edges into two subsets:  $E_1^*$  contains all edges that belong to sub-graphs  $Z'_X$  for  $X \in \mathcal{X}$ , and  $E_2^*$  contains all remaining edges. Notice that since  $E^*$  is a canonical solution, each edge  $e \in E_1^*$  must be a matching edge for some graph  $Z'_X$ . Also from the construction of the contracted graph  $H$ , all edges in  $E_2^*$  belong to  $E(\mathbf{G})$ .

Consider some set  $X \in \mathcal{X}$ , and let  $\Gamma'(X) \subseteq \Gamma(X)$  denote the subset of the interface vertices of  $Z'_X$ , whose matching edges belong to  $E_1^*$ . Let  $\Gamma' = \bigcup_{X \in \mathcal{X}} \Gamma'(X)$ . We now define a subset  $E_1^{**}$  of edges of  $\mathbf{G}$  as follows: for each vertex  $v \in \Gamma'$ , add all edges incident to  $v$  in  $\mathbf{G}$  to  $E_1^{**}$ . Finally, we set  $E^{**} = E_1^{**} \cup E_2^*$ . Notice that  $E^{**}$  is a subset of edges of  $\mathbf{G}$ , and  $|E^{**}| = |E_1^{**}| + |E_2^*| \leq d_{\max}|E_1^*| + |E_2^*| \leq d_{\max}|E^*|$ . In order to complete the proof of the claim, it is enough to show that  $E^{**}$  is a feasible solution to the Minimum Planarization problem on  $\mathbf{G}$ .

Let  $\mathbf{G}' = \mathbf{G} \setminus E^{**}$ , let  $H' = H \setminus E^*$ , and let  $\psi$  be a planar drawing of  $H'$ . It is now enough to construct a planar drawing  $\psi'$  of  $\mathbf{G}'$ . In order to do so, we start from the planar drawing  $\psi$  of  $H'$ . We then consider the sets  $X \in \mathcal{X}$  one-by-one. For each such set, we replace the drawing of  $Z'_X \setminus \Gamma'(X)$  with a drawing of  $\mathbf{G}[X] \setminus \Gamma'(X)$ . The drawings of the vertices in  $\Gamma(X)$  are not changed by this procedure. After all sets  $X \in \mathcal{X}$  are processed, we will obtain a planar drawing of graph  $\mathbf{G}'$  (that may also contain drawings of some edges in  $E^{**}$ , that we can simply erase).

Consider some such set  $X \in \mathcal{X}$ . Let  $G$  be the current graph (obtained from  $H'$  after a number of such replacement steps), and let  $\psi$  be the current planar drawing of  $G$ . Observe that the grid  $Z_X$  has a unique planar drawing. We say that a planar drawing of graph  $Z'_X \setminus \Gamma'(X)$  is *standard* in  $\psi$ , iff we can draw a simple closed curve  $\gamma'_X$ , such that  $Z_X$  is embedded completely inside  $\gamma'_X$ ; no other vertices or edges of  $G$  are embedded inside  $\gamma'_X$ ; the only edges that  $\gamma'_X$  intersects are the matching edges of  $Z'_X \setminus \Gamma'(X)$ , and each such matching edge is intersected exactly once by  $\gamma'_X$ .

It is possible that the drawing of  $Z'_X \setminus \Gamma'(X)$  in  $\psi$  is not standard. However, since  $\psi$  is planar, this can only happen for the following three reasons: (1) some connected component  $C$  of the current graph  $G$  is embedded inside some face of the grid  $Z_X$ : in this case we can simply move the drawing of  $C$  elsewhere; (2) there is some subset  $C$  of  $V(G)$ , and a vertex  $v \in \Gamma(X) \setminus \Gamma'(X)$ , such that  $\Gamma_G(C) = v$ , and  $G[C]$  is embedded inside one of the faces of the grid  $Z_X$  incident to the other endpoint of the matching edge of  $v$ ; and (3) there is some subset  $C$  of  $V(G)$ , and two consecutive vertices  $u, v \in \Gamma(X) \setminus \Gamma'(X)$ , such that  $\Gamma_G(C) = \{u, v\}$ , and  $G[C]$  is embedded inside the unique face of the grid  $Z_X$  incident to the other endpoints of the matching edges of  $u$  and  $v$ . In the latter two cases, we simply move the drawing of  $C$  right outside the grid, so that the corresponding matching edges now cross the curve  $\gamma'(X)$ .

To summarize, we can transform the current planar drawing  $\psi$  of the graph  $G$  into another planar drawing  $\tilde{\psi}$ , such that the induced drawing of  $Z'_X \setminus \Gamma'(X)$  is standard. We can now draw a simple closed curve  $\gamma''(X)$ , such that  $Z'_X \setminus \Gamma'(X)$  is embedded inside  $\gamma''(X)$ , no other vertices or edges are embedded inside  $\gamma''(X)$ , and the set of vertices whose drawings lie on  $\gamma''(X)$  is precisely  $\Gamma(X) \setminus \Gamma'(X)$ . Notice that the ordering of the vertices of  $\Gamma(X) \setminus \Gamma'(X)$  along this curve is exactly the same as their ordering along the curve  $\gamma(X)$  in the planar embedding  $\pi'(X)$  of  $\mathbf{G}[X]$ , guaranteed by Property (C1). Let  $\pi''(X)$  be the drawing of  $\mathbf{G}[X] \setminus \Gamma'(X)$  induced by  $\pi'(X)$ . We can now simply replace the drawing of  $Z'_X \setminus \Gamma'(X)$  with the drawing  $\pi''(X)$  of  $\mathbf{G}[X] \setminus \Gamma'(X)$ , identifying the curves  $\gamma_X$  and  $\gamma'_X$ , and the drawings of the vertices in  $\Gamma(X) \setminus \Gamma'(X)$  on them. The resulting drawing remains planar, and the drawings of the vertices in  $\Gamma(X)$  do not change.  $\square$

Finally, we show that if we find a nasty canonical set in  $\mathbf{G}_{|S}$ , then we can contract  $\mathbf{G}$  even further. The proof of the following theorem is omitted due to lack of space.

**Theorem 4.3** *Let  $S$  be any subset of vertices of  $\mathbf{G}$ ,  $\mathcal{X}$  any partition of  $S$  with properties (C1)–(C5),  $H = \mathbf{G}_{|S}$  the corresponding contracted graph, and  $\mathcal{Z}$  the corresponding collection of grids  $Z_X$  for  $X \in \mathcal{X}$ . Then given any nasty canonical vertex set  $R \subseteq V(H)$ , we can efficiently find a subset  $S' \subseteq V(\mathbf{G})$  of vertices, and a partition  $\mathcal{X}'$  of  $S'$ , such that properties (C1)–(C5) hold for  $\mathcal{X}'$ , and if  $H' = \mathbf{G}_{|S'}$  is the corresponding contracted graph, then  $|V(H')| < |V(H)|$ . Moreover, there is a canonical drawing  $\varphi'$  of  $H'$  with  $\text{cr}_{\varphi'}(H') = O(d_{\max}^2 \cdot \log^{10} n \cdot (\log \log n)^4 \cdot \text{OPT}_{\text{cr}}(\mathbf{G}))$ .*

Notice that Claim 4.2 applies to the new contracted graph as well.

## 5. THE ALGORITHM

The algorithm consists of a number of stages. In each stage  $j$ , we are given as input a subset  $S$  of vertices of  $\mathbf{G}$ , the contracted graph  $\mathbf{H} = \mathbf{G}_{|S}$ , and the collection  $\mathcal{Z}$  of disjoint sub-sets of vertices of  $\mathbf{H}$ , corresponding to the grids  $Z_X$  obtained during the contraction step. The goal of stage  $j$  is to either produce a nasty canonical set  $R$  in  $\mathbf{H}$ , or to find a weak feasible solution to problem  $\pi(\mathbf{H}, \emptyset, \mathcal{Z})$ . We prove the following theorem.

**Theorem 5.1** *There is an efficient randomized algorithm, that, given a contracted graph  $\mathbf{H}$ , a corresponding collection  $\mathcal{Z}$  of disjoint subsets of vertices of  $\mathbf{H}$ , and a bound  $\text{OPT}'$  on the cost of the strong optimal solution to problem  $\pi(\mathbf{H}, \emptyset, \mathcal{Z})$ , with probability at least  $1/\text{poly}(n)$ , produces either a nasty canonical subset  $R$  of vertices of  $\mathbf{H}$ , or a weak feasible solution  $E^*$ ,  $|E^*| \leq O((\text{OPT}')^5 \text{poly}(d_{\max} \cdot \log n))$  for problem  $\pi(\mathbf{H}, \emptyset, \mathcal{Z})$ . (Here,  $n = |V(\mathbf{G})|$ ).*

We provide a high-level sketch of the proof of this theorem in the rest of this section, but we first show that Theorems 1.3, 1.1 and Corollary 1.1 follow from it. We start with proving Theorem 1.3, by showing an efficient randomized algorithm to find a subset  $E^* \subseteq E(\mathbf{G})$  of edges, such that  $\mathbf{G} \setminus E^*$  is planar, and  $|E^*| \leq O((\text{OPT}_{\text{cr}}(\mathbf{G}))^5 \cdot \text{poly}(d_{\max} \cdot \log n))$ . We assume that we know the value  $\text{OPT}_{\text{cr}}(\mathbf{G})$ , by using the standard practice of guessing this value, running the algorithm, and then adjusting the guessed value accordingly. It is enough to ensure that whenever the guessed value  $\text{OPT} \geq \text{OPT}_{\text{cr}}(\mathbf{G})$ , the algorithm indeed returns a subset  $E^*$  of edges,  $|E^*| \leq O(\text{OPT}^5 \text{poly}(d_{\max} \cdot \log n))$ , such that  $\mathbf{G} \setminus E^*$  is a planar graph w.h.p. Therefore, from now on we assume that we are given a value  $\text{OPT} \geq \text{OPT}_{\text{cr}}(\mathbf{G})$ . The algorithm consists of a number of stages. The input to stage  $j$  is a contracted graph  $\mathbf{H}$ , with the corresponding family  $\mathcal{Z}$  of vertex sets. In the input to the first stage,  $\mathbf{H} = \mathbf{G}$ , and  $\mathcal{Z} = \emptyset$ . In each stage  $j$ , we run the algorithm from Theorem 5.1 on the current contracted graph  $\mathbf{H}$ , and the family  $\mathcal{Z}$  of vertex subsets. From Theorem 4.1, there is a strong feasible solution to problem  $\pi(\mathbf{H}, \emptyset, \mathcal{Z})$  of cost  $O(\text{OPT} \cdot \text{poly}(\log n \cdot d_{\max}))$ , and so we can set the parameter  $\text{OPT}'$  to this value. Whenever the algorithm returns a nasty canonical set  $R$  in graph  $\mathbf{H}$ , we terminate the current stage, and compute a new contracted graph  $\mathbf{H}'$ , guaranteed by Theorem 4.3. Graph  $\mathbf{H}'$ , together with the corresponding family  $\mathcal{Z}'$  of vertex subsets, becomes the input to the next stage. Alternatively, if, after  $\text{poly}(n)$  executions of the algorithm from Theorem 5.1, no nasty canonical set is returned, then with high probability, one of the algorithm executions has returned a weak feasible solution  $E^*$ ,  $|E^*| \leq O(\text{OPT}^5 \text{poly}(d_{\max} \cdot \log n))$  for problem  $\pi(\mathbf{H}, \emptyset, \mathcal{Z})$ . From Claim 4.2, we can recover from this solution a planarizing set  $E^{**}$  of edges for graph  $\mathbf{G}$ , with  $|E^{**}| = O(\text{OPT}^5 \text{poly}(d_{\max} \cdot \log n))$ . Since the size of the contracted graph  $\mathbf{H}$  goes down after each contraction step, the number of stages is bounded by  $n$ , thus implying Theorem 1.3. Combining Theorem 1.3 with Theorem 1.2 immediately gives Theorem 1.1. Finally, we obtain Corollary 1.1 as follows. Recall that the algorithm of Even et al. [10] computes a drawing of any  $n$ -vertex bounded degree graph  $\mathbf{G}$  with  $O(\log^2 n) \cdot (n + \text{OPT}_{\text{cr}}(\mathbf{G}))$  crossings. It was shown in [9], that this algorithm can be extended to arbitrary graphs, where the number of crossings becomes  $O(\text{poly}(d_{\max}) \cdot \log^2 n) \cdot (n + \text{OPT}_{\text{cr}}(\mathbf{G}))$ . We run their algorithm, and the algorithm pre-



sented in this section, on graph  $\mathbf{G}$ , and output the better of the two solutions. If  $\text{OPT}_{\text{cr}}(\mathbf{G}) < n^{1/10}$ , then our algorithm gives an  $O(n^{9/10} \text{poly}(d_{\max} \cdot \log n))$ -approximation; otherwise, the algorithm of [10] gives an  $O(n^{9/10} \text{poly}(d_{\max} \cdot \log n))$ -approximation.

The remainder of this section is devoted to proving Theorem 5.1. Recall that we are given the contracted graph  $\mathbf{H}$ , and a collection  $\mathcal{Z}$  of vertex-disjoint subsets of  $V(\mathbf{H})$ . For each  $Z \in \mathcal{Z}$ ,  $\mathbf{H}[Z]$  is a grid, and  $E(Z, V(\mathbf{H}) \setminus Z)$  consists of a set  $M_Z$  of matching edges. Each such edge connects a vertex in the first row of  $Z$  to a distinct vertex in  $T_{\mathbf{H}}(Z)$ , and these edges form a matching between the first row of  $Z$  and  $T_{\mathbf{H}}(Z)$ . Abusing the notation, we denote the bound on the cost of the strong optimal solution to  $\pi(\mathbf{H}, \emptyset, \mathcal{Z})$  by  $\text{OPT}$  from now on, and the number of vertices in  $\mathbf{H}$  by  $n$ . For each  $Z \in \mathcal{Z}$ , we use  $Z$  to denote both the set of vertices itself, and the grid  $\mathbf{H}[Z]$ . We assume throughout the rest of the section that  $\text{OPT} \cdot d_{\max}^6 < \sqrt{n}$ : otherwise, if  $\text{OPT} \cdot d_{\max}^6 \geq \sqrt{n}$ , then the set  $E'$  of all edges of  $\mathbf{H}$  that do not participate in grids  $Z \in \mathcal{Z}$ , is a feasible weak canonical solution for problem  $\pi(\mathbf{H}, \emptyset, \mathcal{Z})$ . It is easy to see that  $|E'| \leq O(\text{OPT}^2 \text{poly}(d_{\max}))$ : this is clearly the case if  $|E'| \leq 4n$ ; otherwise, if  $|E'| > 4n$ , then by Theorem 2.3,  $\text{OPT} = \Omega(n)$ , and so  $|E'| = O(n^2) = O(\text{OPT}^2)$ .

We use two parameters:  $\rho = O(\text{OPT} \text{poly}(d_{\max} \cdot \log n))$  and  $m^* = O(\text{OPT}^3 \cdot \text{poly}(d_{\max} \cdot \log n))$ . The algorithm consists of  $2\rho \log n$  iterations. The input to iteration  $h$  is a collection  $G_1, \dots, G_{k_h}$  of  $k_h \leq \text{OPT}$  sub-graphs of  $\mathbf{H}$ , together with bounding boxes  $X_i \subseteq G_i$  for all  $1 \leq i \leq k_h$ . We denote  $H_i = G_i \setminus V(X_i)$  and  $n(H_i) = |V(H_i)|$ . Additionally, we have collections  $E^{(1)}, \dots, E^{(h-1)}$  of edges of  $\mathbf{H}$ , where for each  $1 \leq h' \leq h-1$ , set  $E^{(h')}$  has been computed in iteration  $h'$ . We say that  $(G_1, X_1), \dots, (G_{k_h}, X_{k_h})$ , and  $E^{(1)}, \dots, E^{(h-1)}$  is a *valid* input to iteration  $h$ , iff the following invariants hold:

- V1. For all  $1 \leq i, j \leq k_h$ , graphs  $H_i$  and  $H_j$  are completely disjoint.
- V2. For all  $1 \leq i \leq k_h$ ,  $G_i \subseteq \mathbf{H} \setminus (E^{(1)}, \dots, E^{(h-1)})$ , and  $H_i$  is the sub-graph of  $\mathbf{H}$  induced by  $V(H_i)$ . In particular, no edges  $e \subseteq V(H_i)$  belong to  $E^{(1)}, \dots, E^{(h-1)}$ . Moreover, every edge  $e \in E(\mathbf{H})$  belongs to either  $\bigcup_{h'=1}^h E^{(h')}$  or to  $\bigcup_{i=1}^{k_h} G_i$ .
- V3. For all  $Z \in \mathcal{Z}$ , for all  $1 \leq i \leq k_h$ , either  $Z \cap V(H_i) = \emptyset$ , or  $Z \subseteq V(H_i)$ . Let  $\mathcal{Z}_i = \{Z \in \mathcal{Z} \mid Z \subseteq V(H_i)\}$ .
- V4. For each  $1 \leq i \leq k_h$ , there is a strong solution  $\varphi_i$  to  $\pi(G_i, X_i, \mathcal{Z}_i)$ , with  $\sum_{i=1}^{k_h} \text{cr}_{\varphi_i}(G_i) \leq \text{OPT}$ .
- V5. If we are given, for each  $1 \leq i \leq k_h$ , **any** weak solution  $E'_i$  to problem  $\pi(G_i, X_i, \mathcal{Z}_i)$ , and denote  $\tilde{E}^{(h)} = \bigcup_{i=1}^{k_h} E'_i$ , then  $E^{(1)} \cup \dots \cup E^{(h-1)} \cup \tilde{E}^{(h)}$  is a feasible weak solution to problem  $\pi(\mathbf{H}, \emptyset, \mathcal{Z})$ .
- V6. For each  $1 \leq h' < h$ , and  $1 \leq i \leq k_h$ , the number of edges in  $E^{(h')}$  incident on vertices of  $H_i$  is at most  $m^*$ , and  $|E^{(h')}| \leq \text{OPT} \cdot m^*$ . Moreover, no edges in grids  $Z \in \mathcal{Z}$  belong to  $\bigcup_{h'=1}^{h-1} E^{(h')}$ .
- V7. Let  $n_h = (1 - 1/\rho)^{(h-1)/2} \cdot n$ . For each  $1 \leq i \leq k_h$ , either  $n(H_i) \leq n_h$ , or  $X_i = \emptyset$  and  $n(H_i) \leq n_{h-1}$ .

The input to the first iteration consists of a single graph,  $G_1 = \mathbf{H}$ , with the bounding box  $X_1 = \emptyset$ . It is easy to see that all invariants hold for this input. We end the algorithm at iteration  $h^*$ , where  $n_{h^*} \leq (m^* \cdot \rho \cdot \log n)^2$ . Clearly,  $h^* \leq 2\rho \log n$ , from Invariant (V7). Let  $\mathcal{G}$  be the set of all instances that serve as input to iteration  $h^*$ . We need the following theorem, whose proof is omitted due to lack of space.

**Theorem 5.2** *There is an efficient algorithm, that, given any problem instance  $\pi(G, X, \mathcal{Z}')$ , where  $V(G \setminus X)$  is canonical for  $\mathcal{Z}'$ , and  $\pi(G, X, \mathcal{Z}')$  has a strong solution of cost  $\text{OPT}$ , finds a weak feasible solution to  $\pi(G, X, \mathcal{Z}')$  of cost  $O(\overline{\text{OPT}} \cdot \sqrt{n'} \cdot \text{poly}(d_{\max} \cdot \log n') + \overline{\text{OPT}}^3)$ , where  $n' = |V(G \setminus X)|$ , and  $d_{\max}$  is the maximum degree in  $G$ .*

For each  $1 \leq i \leq k_{h^*}$ , let  $E_i^{(h^*)}$  be the weak solution from Theorem 5.2, and let  $E^{(h^*)} = \bigcup_{i=1}^{k_{h^*}} E_i^{(h^*)}$ . Let  $\text{OPT}_i$  denote the cost of the strong optimal solution to  $\pi(G_i, X_i, \mathcal{Z}_i)$ . Then  $|E^{(h^*)}| = \sum_{i=1}^{k_{h^*}} O(\text{OPT}_i \cdot \sqrt{n(H_i)} \cdot \text{poly}(d_{\max} \cdot \log n) + \text{OPT}_i^3)$ . Since  $n(H_i) \leq n_{h^*-1} \leq 2n_{h^*}$  for all  $i$ , this is bounded by  $\sum_{i=1}^{k_{h^*}} O(\text{OPT}_i \cdot m^* \cdot \rho \cdot \text{poly}(d_{\max} \log n) + \text{OPT}_i^3) \leq O(\text{OPT} \cdot m^* \cdot \rho \cdot \text{poly}(d_{\max} \log n) + \text{OPT}^3)$ , as  $\sum_{i=1}^{k_{h^*}} \text{OPT}_i \leq \text{OPT}$  from Invariant (V4). The final solution is  $E^* = \bigcup_{h=1}^{h^*} E^{(h)}$ , and

$$\begin{aligned} |E^*| &\leq \sum_{h=1}^{h^*-1} |E^{(h)}| + |E^{(h^*)}| \\ &\leq (2\rho \log n)(\text{OPT} \cdot m^*) \\ &\quad + O(\text{OPT} \cdot m^* \cdot \rho \cdot \text{poly}(d_{\max} \cdot \log n)) \\ &\quad + O(\text{OPT}^3) \\ &= O(\text{OPT}^5 \text{poly}(d_{\max} \cdot \log n)). \end{aligned}$$

We say that the execution of iteration  $h$  is *successful*, iff it either produces a valid input to the next iteration, together with the set  $E^{(h)}$  of edges, or finds a nasty canonical set in  $\mathbf{H}$ . We can execute each iteration, so that it is successful with probability at least  $(1 - 1/\rho)$ , if all previous iterations were successful. If any iteration returns a nasty canonical set, then we stop the algorithm and return this vertex set as an output. Since there are at most  $2\rho \log n$  iterations, the probability that all iterations are successful is at least  $(1 - 1/\rho)^{2\rho \log n} \geq 1/\text{poly}(n)$ . In order to complete the proof of Theorem 5.1, it is now enough to show an algorithm for executing each iteration, such that, given a valid input to the current iteration, the algorithm either finds a nasty canonical set in  $\mathbf{H}$ , or returns a valid input to the next iteration, with probability at least  $\frac{1}{\rho}$ . Due to lack of space, this part is omitted, and can be found in the full version of the paper, available as an arXiv preprint, arXiv:1012.0255v1.

## 6. CONCLUSIONS

We have shown an efficient randomized algorithm to find a drawing of any graph  $\mathbf{G}$  in the plane, with the number of crossings bounded by  $O((\text{OPT}_{\text{cr}}(\mathbf{G}))^{10} \text{poly}(d_{\max} \cdot \log n))$ . We did not make an effort to optimize the powers of  $\text{OPT}$ ,  $d_{\max}$  and  $\log n$  in this guarantee, or the constant hidden in the  $O(\cdot)$  notation, and we believe that they can be improved. We hope that the technical tools developed in this

paper will help obtain better algorithms for the Minimum Crossing Number problem. A specific possible direction is obtaining efficient algorithms for  $\rho$ -balanced  $\alpha$ -well-linked bi-partitions. In particular, an interesting open question is whether there is an efficient algorithm, that, given an  $n$ -vertex graph  $G$  with maximum degree  $d_{\max}$ , finds a  $\rho$ -balanced  $\alpha$ -well-linked bi-partition of  $G$ , for values  $\rho, 1/\alpha = \text{poly}(d_{\max} \cdot \log n)$ . In fact it is not even clear whether such a bi-partition exists in every graph. We note that the dependence of  $\rho$  on  $d_{\max}$  is necessary, for example, in the star graph. This question appears to be interesting in its own right, and its positive resolution (even if the parameters  $\rho$  and  $\alpha$  depend on  $\text{OPT}_{\text{cr}}(G)$ ) would greatly simplify our algorithm and improve its performance guarantee.

### Acknowledgements.

The author thanks Yury Makarychev and Anastasios Sidiropoulos for many fruitful discussions, and for reading earlier drafts of the paper.

## 7. REFERENCES

- [1] M. Ajtai, V. Chvátal, M. Newborn, and E. Szemerédi. Crossing-free subgraphs. *Theory and Practice of Combinatorics*, pages 9–12, 1982.
- [2] C. Ambuhl, M. Mastrolilli, and O. Svensson. Inapproximability results for sparsest cut, optimal linear arrangement, and precedence constrained scheduling. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 329–337, 2007.
- [3] S. Arora, S. Rao, and U. V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2), 2009.
- [4] S. N. Bhatt and F. T. Leighton. A framework for solving VLSI graph layout problems. *J. Comput. Syst. Sci.*, 28(2):300–343, 1984.
- [5] K. J. Börözký, J. Pach, and G. Tóth. Planar crossing numbers of graphs embeddable in another surface. *Int. J. Found. Comput. Sci.*, 17(5):1005–1016, 2006.
- [6] S. Cabello and B. Mohar. Adding one edge to planar graphs makes crossing number hard. In *Proc. ACM Symp. on Computational Geometry*, 2010.
- [7] C. Chekuri, S. Khanna, and F. B. Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 183–192, New York, NY, USA, 2005. ACM.
- [8] M. Chimani, P. Hliněný, and P. Mutzel. Approximating the crossing number of apex graphs. In *Graph Drawing*, pages 432–434, 2008.
- [9] J. Chuzhoy, Y. Makarychev, and A. Sidiropoulos. On graph crossing number and edge planarization. *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1050–1069, 2011. Full version: arXiv:1010.3976v1, Oct. 2010.
- [10] G. Even, S. Guha, and B. Schieber. Improved approximations of crossings in graph drawings and VLSI layout areas. *SIAM J. Comput.*, 32(1):231–252, 2002.
- [11] M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM J. Algebraic Discrete Methods*, 4(3):312–316, 1983.
- [12] I. Gitler, P. Hliněný, J. Leaños, and G. Salazar. The crossing number of a projective graph is quadratic in the face-width. *Electronic Notes in Discrete Mathematics*, 29:219–223, 2007.
- [13] M. Grohe. Computing crossing numbers in quadratic time. *J. Comput. Syst. Sci.*, 68(2):285–302, 2004.
- [14] R. Guy. A combinatorial problem. *Nabla (Bulletin of the Malayan Mathematical Society)*, 7:68–72, 1960.
- [15] P. Hliněný. Crossing number is hard for cubic graphs. *J. Comb. Theory, Ser. B*, 96(4):455–471, 2006.
- [16] P. Hliněný and M. Chimani. Approximating the crossing number of graphs embeddable in any orientable surface. In *Proc. 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010.
- [17] P. Hliněný and G. Salazar. Approximating the crossing number of toroidal graphs. *Lecture Notes in Computer Science*, 4835/2007:148–159, 2007.
- [18] J. Hopcroft and R. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.
- [19] K. Kawarabayashi and B. A. Reed. Computing crossing number in linear time. In *STOC*, pages 382–390, 2007.
- [20] F. T. Leighton. *Complexity issues in VLSI: optimal layouts for the shuffle-exchange graph and other networks*. MIT Press, 1983.
- [21] F. T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46:787–832, 1999.
- [22] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.
- [23] J. Matoušek. *Lectures on discrete geometry*. Springer-Verlag, 2002.
- [24] J. Pach and G. Tóth. Thirteen problems on crossing numbers. *Geombinatorics*, 9(4):194–207, 2000.
- [25] S. Pan and R. B. Richter. The crossing number of  $K_{11}$  is 100. *J. Graph Theory*, 56(2):128–134, 2007.
- [26] H. Räcke. Minimizing congestion in general networks. In *In Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 43–52, 2002.
- [27] R. B. Richter and G. Salazar. Crossing numbers. In L. W. Beineke and R. J. Wilson, editors, *Topics in Topological Graph Theory*, chapter 7, pages 133–150. Cambridge University Press, 2009.
- [28] L. A. Szekeley. Progress on crossing number problems. In *SOFSEM*, pages 53–61, 2005.
- [29] P. Turán. A note of welcome. *J. Graph Theory*, 1:1–5, 1977.
- [30] I. Vrto. Crossing numbers of graphs: A bibliography. <http://www.ifi.savba.sk/~imrich>.
- [31] H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54(1):150–168, 1932.
- [32] D. R. Wood and J. A. Telle. Planar decompositions and the crossing number of graphs with an excluded minor. *New York J. Mathematics*, 13:117–146, 2007.