THE UNIVERSITY OF CHICAGO

FROM STRUCTURAL GRAPH THEORY TO GRAPH ALGORITHMS, AND BACK

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCE
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
ZIHAN TAN

CHICAGO, ILLINOIS
MAY 2022

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Julia Chuzhoy, for her guidance, care, encouragement, and support. She taught me how to do research step by step, and never let me away from sloppy proofs and insufficient explanation in the writing. I also learned and benefit a lot from her advice on career and life. I consider myself extremely lucky to be her student. I cannot thank you enough, Julia!

I would like to thank my co-advisor Laci Babai, for his enduring support and encouragement. Research can be frustrating, but every time after talking to Laci, I felt fully recharged, excited, and enthusiastic. He inspired me not only by his outstanding contribution to theoretical computer science but also by his devotion to teaching and his love for students. I would also like to thank Lorenzo Orecchia, for the helpful and inspiring discussions. I am honored to have you on my thesis committee.

I am also grateful to Sanjeev Khanna and Robert Krauthgamer, my collaborators and mentors. Working with you is an enjoyable and rewarding experience. Your copious advice also had a great impact on me, and many fruitful discussions with you open my eyes to a broader world of research.

I gratefully acknowledge all my collaborators during my PhD studies. Special thanks to Yu Chen, for our countless productive discussions. I would like to also thank Hsien-Chih Chang, Gramoz Goranci, Sepideh Mahabadi, Merav Parter, Harald Räcke, Yifeng Teng, Liwei Zeng, and Mingfei Zhao. Working with you has been a pleasure for me.

I would like to extend my gratitude to other theoretical computer science faculties at The University of Chicago and Toyota Technological Institute at Chicago - especially to Aaron Potechin, Andrew Drucker, Yury Makarychev, Alexander Razborov, Janos Simon, and Madhur Tulsiani, for creating a warm and inspiring atmosphere for academic pursuits. I would also like to thank my friends in both institutes, Christopher Jones, Goutham Rajendran, Rachit Nimavat, Leonardo Coregliano, Fernando Granha Jeronimo, and Mrinalkanti Ghosh, David Kim, with whom I enjoyed learning and working and had a lot of enlightening discussions.

Lastly, my love and thanks go to my family, whose constant encouragement and support have been invaluable to me. Most importantly, I would like to thank my beloved wife, Anran Li. Your unconditional love, company, and support helped me through the past six years, and I promise to be there for you forever.

ABSTRACT

In this thesis we study two algorithmic problems on graphs: Graph Crossing Number and the Packing of Low-Diameter Spanning Trees.

In the first part of the thesis, we consider the *Graph Crossing Number* problem, in which we are given a graph and are asked to find a drawing of it in the plane that minimizes the number of crossings. For bounded-degree graphs, almost all previous algorithms followed the same framework, and the best of them achieved an $\tilde{O}(\sqrt{n})$-approximation, which was also proved to be optimal in this framework. We propose a new framework that overcomes this barrier. This allows us to reduce the problem to another related problem called the *Crossing Number with Rotation System* problem, and eventually obtain an $n^{o(1)}$-approximation for the Crossing Number problem on low-degree graphs.

In the second part of this thesis, we study the problem of packing low-diameter spanning trees. The celebrated tree-packing theorem due to Tutte and Nash-Williams states that every $2k$-edge-connected graph contains k edge-disjoint spanning trees. We use the techniques of randomized algorithms to show that every $2k$-edge-connected graph with diameter D contains k spanning trees with diameter $k^{O(D)}$ each, that cause edge-congestion 2. We then show that the same techniques can be applied to improve the Karger's edge-sampling theorem. Finally, we use these graph-theoretic results to improve the round complexities of various distributed graph algorithms.

CHAPTER 1

A SUBPOLYNOMIAL APPROXIMATION ALGORITHM FOR GRAPH CROSSING NUMBER IN LOW-DEGREE GRAPHS

1.1 Introduction

Given a graph G, a *drawing* of G is an embedding of the graph into the plane, that maps every vertex to a point in the plane, and every edge to a continuous curve connecting the images of its endpoints. We require that the image of an edge may not contain images of vertices of G other than its two endpoints, and no three curves representing drawings of edges of G may intersect at a single point, unless that point is the image of their shared endpoint. A *crossing* in a drawing of G is a point that belongs to the images of two edges of G, and is not their common endpoint. In the Minimum Crossing Number problem, the goal is to compute a drawing of the input n-vertex graph G with minimum number of crossings. We denote the value of the optimal solution to this problem, also called the *crossing number* of G, by $\mathsf{OPT}_{\mathsf{cr}}(G)$.

The Minimum Crossing Number problem naturally arises in several areas of Computer Science and Mathematics. The problem was initially introduced by Turán [35], who considered the question of computing the crossing number of complete bipartite graphs, and since then it has been extensively studied (see, e.g., [35, 9, 12, 8, 7, 23, 24], and also [33, 31, 29, 37] for surveys on this topic). The problem is known to be APX-hard, and it remains APX-hard on cubic graphs [17, 20, 5]. The Minimum Crossing Number problem appears to be notoriously difficult from the approximation perspective. All currently known algorithms achieve approximation factors that depend polynomially on Δ – the maximum vertex degree of the input graph, and, to the best of our knowlgedge, no non-trivial approximation algorithms are known for graphs with arbitrary vertex degrees. We note that the famous Crossing Number Inequality [1, 26] shows that, for every graph G with $|E(G)| \geq 4n$, the crossing number of G is $\Omega(|E(G)|^3/n^2)$.

Since the problem is most interesting when the crossing number of the input graph is low, and since our understanding of the problem is still extremely poor, it is reasonable to focus on designing algorithms for low-degree graphs. Throughout, we denote by Δ the maximum vertex degree of the input graph. While we do not make this assumption explicitly, it may be convenient to think of Δ as being bounded by a constant or by some slowly-growing function of n like $\operatorname{poly}\log n$.

The first non-trivial algorithm for the Minimum Crossing Number problem, due to Leighton and Rao [27], combined their algorithm for balanced separators with the framework of [4], to compute a drawing of input graph G with $O\big((n + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \Delta^{O(1)} \cdot \log^4 n\big)$ crossings. This bound was later improved to $O\big((n + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \Delta^{O(1)} \cdot \log^3 n\big)$ by [16], and then to $O\big((n + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \Delta^{O(1)} \cdot \log^2 n\big)$ as a consequence of the improved algorithm of [3] for Balanced Cut. All these results provide an $O(n \cdot \operatorname{poly}(\Delta \log n))$-approximation for Minimum Crossing Number (but perform much better when $\mathsf{OPT}_{\mathsf{cr}}(G)$ is high). For a long time, this remained the best approximation algorithm for Minimum Crossing Number, while the best inapproximability result, to this day, only rules out the existence of a PTAS, unless NP has randomized subexponential time algorithms [17, 2].

A sequence of two papers [12, 9] was the first to break the barrier of $\tilde{\Theta}(n)$-approximation, providing a $\tilde{O}\left(n^{9/10} \cdot \Delta^{O(1)}\right)$-approximation algorithm. Recently, a breakthrough sequence of works [23, 24] has led to an improved $\tilde{O}\left(\sqrt{n} \cdot \Delta^{O(1)}\right)$-approximation for Minimum Crossing Number. All the above-mentioned algorithms exploit the same algorithmic paradigm, that builds on the connection of the Minimum Crossing Number problem to a related problem called Minimum Planarization. However, it was shown by Chuzhoy, Madan and Mahabadi [11] that this paradigm is unlikely to yield a better than $O(\sqrt{n})$-approximation for Minimum Crossing Number.

The main result in the first part of this thesis is a randomized $2^{O((\log n)^{7/8} \log\log n)} \cdot \Delta^{O(1)}$-approximation algorithm for Minimum Crossing Number, which is the first approximation algorithm that achieves a subpolynomial in n approximation factor in graphs whose maximum vertex degree is subpolynomial in n. In order to achieve this result, we reduce the $\mathsf{Minimum\ Crossing\ Number}$ problem to a related problem, called $\mathsf{Minimun\ Crossing\ Number\ with\ Rotation\ System}$ (MCNwRS), and then design a new algorithm for MCNwRS. In MCNwRS, the input consists of a multigraph G, and, for every vertex $v \in V(G)$, a circular ordering $\mathcal{O}_v$ of edges that are incident to v, that we call a *rotation* for vertex v. The set $\Sigma = \{\mathcal{O}_v\}_{v\in V(G)}$ of all such rotations is called a *rotation system* for graph G. We say that a drawing φ of G *obeys* the rotation system Σ, if, for every vertex $v \in V(G)$, the images of the edges in $\delta_G(v)$ enter the image of v in the order $\mathcal{O}_v$ (but the *orientation* of the ordering can be either clock-wise or counter-clock-wise). In the MCNwRS problem, given a graph G and a rotation system Σ for G, the goal is to compute a drawing φ of G that obeys the rotation system Σ and minimizes the number of edge crossings. We construct a randomized $2^{O((\log m)^{7/8}\log\log m)}$-approximation algorithm for MCNwRS (where m is the number of the edges in the input graph to MCNwRS), and then use it to obtain a randomized $2^{O((\log n)^{7/8}\log\log n)}\cdot\Delta^{O(1)}$-approximation algorithm for $\mathsf{Minimum\ Crossing\ Number}$.

1.1.1 Our Results

Given an instance (G, Σ) of MCNwRS, we denote by $\mathsf{OPT}_{\mathsf{cnwrs}}(G, \Sigma)$ the number of crossings in an optimal solution to instance (G, Σ). Our first result is a reduction from $\mathsf{Minimum\ Crossing\ Number}$ to MCNwRS, which is summarized in the following theorem.

Theorem 1.1.1. *There is an efficient algorithm, that, given an n-vertex graph G with maximum vertex degree Δ, computes an instance (G', Σ) of the MCNwRS problem, such that the number of edges in G' is at most $O\left(\mathsf{OPT}_{\mathsf{cr}}(G)\cdot \mathrm{poly}(\Delta \log n)\right)$, and $\mathsf{OPT}_{\mathsf{cnwrs}}(G', \Sigma) \le O\left(\mathsf{OPT}_{\mathsf{cr}}(G)\cdot \mathrm{poly}(\Delta \log n)\right)$. Moreover, there is an efficient algorithm that, given any solution to instance (G', Σ) of MCNwRS of value X, computes a drawing of G with at most $O\left((X + \mathsf{OPT}_{\mathsf{cr}}(G))\cdot \mathrm{poly}(\Delta \log n)\right)$ crossings.*

Our second result is an approximation algorithm for the MCNwRS problem, that is summarized in the following theorem.

Theorem 1.1.2. *There is an efficient randomized algorithm, that, given an instance $I = (G, \Sigma)$ of MCNwRS with $|E(G)| = m$, computes a drawing of G that obeys the rotation system Σ. The number of crossings in the drawing is w.h.p. bounded by $2^{O((\log m)^{7/8} \log\log m)} \cdot (\mathsf{OPT}_{\mathsf{cnwrs}}(I) + m)$.*

Combining Theorem 1.1.2 and Theorem 1.1.1, we obtain the following corollary, whose proof is deferred to Section 1.9.1.

Corollary 1.1.3. *There is an efficient randomized algorithm, that, given a simple n-vertex graph G with maximum vertex degree Δ, computes a drawing of G, such that, w.h.p., the number of crossings in the drawing is at most $2^{O((\log n)^{7/8} \log\log n)} \cdot \mathrm{poly}(\Delta) \cdot \mathsf{OPT}_{\mathsf{cr}}(G)$.*

1.1.2 Organization

We provide the proof of Theorem 1.1.1 in the remaining sections of the first part of the thesis, and the proof of Theorem 1.1.2 can be found in [14]. We start with some basic definitions and notations in Section 1.2. We then provide a high-level overview of the proof of Theorem 1.1.1 and state the main theorems used in the proof of Section 1.3. In Section 1.4 we present some definitions and general results regarding block decompositions of graphs (mostly from previous work). The proof of Theorem 1.1.1 is completed in Sections 1.5–1.8.

1.2 Preliminaries

All logarithms are to the base of 2. All graphs are finite, simple and undirected. Graphs with parallel edges are explicitly referred to as multi-graphs.

We follow standard graph-theoretic notation. Assume that we are given a graph $G = (V, E)$. For a vertex $v \in V$, we denote by $\delta_G(v)$ the set of all edges of G that are incident to v.

For two disjoint subsets A, B of vertices of G, we denote by $E_G(A, B)$ the set of all edges with one endpoint in A and another in B. For a subset $S \subseteq V$ of vertices, we denote by $E_G(S)$ the set of all edges with both endpoints in S, and we denote by $\mathrm{out}_G(S)$ the subset of edges of E with exactly one endpoint in S, namely $\mathrm{out}_G(S) = E_G(S, V \setminus S)$. We denote by $G[S]$ the subgraph of G induced by S. We sometimes omit the subscript G if it is clear from the context. We say that a graph G is ℓ-connected for some integer $\ell > 0$, if there are ℓ vertex-disjoint paths between every pair of vertices in G.

Given a graph $G = (V, E)$, a *drawing* φ of G is an embedding of the graph into the plane, that maps every vertex to a point and every edge to a continuous curve that connects the images of its endpoints. We require that the interiors of the curves representing the edges do not contain the images of any of the vertices. We say that two edges *cross* at a point p, if the images of both edges contain p, and p is not the image of a shared endpoint of these edges. We require that no three edges cross at the same point in a drawing of φ. We say that φ is a *planar drawing* of G iff no pair of edges of G crosses in φ. For a vertex $v \in V(G)$, we denote by $\varphi(v)$ the image of v, and for an edge $e \in E(G)$, we denote by $\varphi(e)$ the image of e in φ. For any subgraph C of G, we denote by $\varphi(C)$ the union of images of all vertices and edges of C in φ. For a path $P \subseteq G$, we sometimes refer to $\varphi(P)$ as the *image of P in φ*. Note that a drawing of G in the plane naturally defines a drawing of G on the sphere and vice versa; we use both types of drawings. Given a graph G and a drawing φ of G in the plane, we use $\mathrm{cr}(\varphi)$ to denote the number of crossings in φ. Let φ' be the drawing of G that is a mirror image of φ. Whitney [38] proved that every 3-connected planar graph has a unique planar drawing. Throughout, for a 3-connected planar graph G, we denote by ρ_G the unique planar drawing of G.

Faces and Face Boundaries. Suppose we are given a planar graph G and a drawing φ of G in the plane. The set of faces of φ is the set of all connected regions of $\mathbb{R}^2 \setminus \varphi(G)$. We designate a single face of φ as the "outer", or the "infinite" face. The *boundary* $\delta(F)$ of a face F is a subgraph of G consisting of all vertices and edges of G whose image is incident to F. Notice that, if graph G is not connected, then boundary of a face may also be not connected. Unless φ has a single face, the boundary $\delta(F)$ of every face F of φ must contain a simple cycle $\delta'(F)$ that separates F from the outer face. We sometimes refer to graph $\delta(F) \setminus \delta'(F)$ as the *inner boundary* of F. Lastly, observe that, if G is 2-connected, then the boundary of every face of φ is a simple cycle.

Bridges and Extensions of Subgraphs. Let G be a graph, and let $C \subseteq G$ be a subgraph of G. A *bridge* for C in graph G is either (i) an edge $e = (u, v) \in E(G)$ with $u, v \in V(C)$ and $e \notin E(C)$; or (ii) a connected component of $G \setminus V(C)$. We denote by $\mathcal{R}_G(C)$ the set of all bridges for C in graph G. For each bridge $R \in \mathcal{R}_G(C)$, we define the set of vertices $L(R) \subseteq V(C)$, called the *legs of* R, as follows. If R consists of a single edge e, then $L(R)$ contains the endpoints of e. Otherwise, $L(R)$ contains all vertices $v \in V(C)$, such that v has a neighbor that belongs to R.

Next, we define an *extension* of the subgraph $C \subseteq G$, denoted by $X_G(C)$. The extension contains, for every bridge $R \in \mathcal{R}_G(C)$, a tree T_R, that is defined as follows. If R is a bridge consisting of a single edge e, then the corresponding tree T_R only contains the edge e. Otherwise, let R' be the subgraph of G consisting of the graph R, the vertices of $L(R)$, and all edges of G connecting vertices of R to vertices of $L(R)$. We let $T_R \subseteq R'$ be a tree, whose leaves are precisely the vertices of $L(R)$. Note that such a tree exists because graph R is connected, and it can be found efficiently. We let the extension of C in G be $X_G(C) = \{T_R \mid R \in \mathcal{R}_G(C)\}$.

Sparsest Cut and Well-Linkedness. Suppose we are given a graph $G = (V, E)$, and a subset $\Gamma \subseteq V$ of its vertices. We say that a cut (X, Y) in G is a valid Γ-cut iff $X \cap \Gamma, Y \cap \Gamma \neq \emptyset$. The *sparsity* of a valid Γ-cut (X, Y) is $\frac{|E(X,Y)|}{\min\{|X \cap \Gamma|, |Y \cap \Gamma|\}}$. In the Sparsest Cut problem, given a graph G and a subset Γ of its vertices, the goal is to compute a valid Γ-cut of minimum sparsity. Arora, Rao and Vazirani [3] have shown an $O(\sqrt{\log n})$-approximation algorithm for the sparsest cut problem. We denote this algorithm by $\mathcal{A}_{\text{ARV}}$, and its approximation factor by $\alpha_{\text{ARV}}(n) = O(\sqrt{\log n})$. We say that a set Γ of vertices of G is *α-well-linked* in G, iff the value of the sparsest cut in G with respect to Γ is at least α.

1.3 High-Level Overview

In this section we provide a high-level overview of the proof of Theorem 1.1.1, and state the main theorems from which the proof is derived. As in previous work, we start by considering a special case of Minimum Crossing Number, where the input graph G is 3-connected. This special case captures the main technical challenges of the whole problem, and the extension to non-3-connected graphs is relatively easy and follows the same framework as in previous work [12]. We start by defining several central notions that our proof uses.

1.3.1 Acceptable Clusters and Decomposition into Acceptable Clusters

In this section we define acceptable clusters and decomposition into acceptable clusters. These definitions are central to all our results. Let G be an input graph on n vertices of maximum degree at most Δ; we assume that G is 3-connected. Let $\hat{E}$ be any planarizing set of edges for G, and let $H = G \setminus \hat{E}$. Let $\Gamma \subseteq V(G)$ be the set of all vertices that serve as endpoints of edges in $\hat{E}$; we call the vertices of Γ *terminals*. We will define a set A of *fake edges*; for every fake edge $e \in A$, both endpoints of e must lie in Γ. We emphasize that the edges of A do not necessarily lie in H or in G; in fact we will use these edges in order to augment the graph H.

We denote by $\mathcal{C}$ the set of all connected components of graph $H \cup A$, and we call elements of $\mathcal{C}$ *clusters*. For every cluster $C \in \mathcal{C}$, we denote by $\Gamma(C) = \Gamma \cap V(C)$ the set of all terminals that lie in C. We also denote by $A_C = A \cap C$ the set of all fake edges that lie in C.

Definition 1. *We say that a cluster $C \in \mathcal{C}$ is a* type-1 acceptable cluster *iff:*

- $A_C = \emptyset$; *and*
- $|\Gamma(C)| \leq \mu$ *for* $\mu = 512\Delta\alpha_{\text{ARV}}(n)\log_{3/2} n = O(\Delta \log^{1.5} n)$ *(recall that* $\alpha_{\text{ARV}}(n) = O(\sqrt{\log n})$ *is the approximation factor of the algorithm* $\mathcal{A}_{\text{ARV}}$ *for the sparsest cut problem).*

Consider now some cluster $C \in \mathcal{C}$, and assume that it is 2-connected. For a pair (u, v) of vertices of C, we say that (u, v) is a 2-separator for C iff the graph $C \setminus \{u, v\}$ is not connected. We denote by $S_2(C)$ the set of all vertices of C that participate in 2-separators, that is, a vertex $v \in C$ belongs to $S_2(C)$ iff there is another vertex $u \in C$ such that (v, u) is a 2-separator for C. Next, we define type-2 acceptable clusters.

Definition 2. *We say that a cluster $C \in \mathcal{C}$ is a* type-2 acceptable cluster *with respect to its drawing ψ'_C on the sphere if the following conditions hold:*

- **(Connectivity):** *C is a simple 2-connected graph, and $|S_2(C)| \leq O(\Delta|\Gamma(C)|)$. Additionally, graph $C \setminus A_C$ is a 2-connected graph.*
- **(Planarity):** *C is a planar graph, and the drawing ψ'_C is planar. We denote by $\psi_{C \setminus A_C}$ the drawing of $C \setminus A_C$ is induced by ψ'_C.*
- **(Bridge Consistency Property):** *for every bridge $R \in \mathcal{R}_G(C \setminus A_C)$, there is a face F in the drawing $\psi_{C \setminus A_C}$ of $C \setminus A_C$, such that all vertices of $L(R)$ lie on the boundary of F; and*
- **(Well-Linkedness of Terminals):** *the set $\Gamma(C)$ of terminals is α-well-linked in $C \setminus A_C$, for $\alpha = \frac{1}{128\Delta\alpha_{\text{ARV}}(n)\log_{3/2} n} = \Theta\left(\frac{1}{\Delta \log^{1.5} n}\right)$.*

Let $\mathcal{C}_1 \subseteq \mathcal{C}$ denote the set of all type-1 acceptable clusters. For a fake edge $e = (x, y) \in A$, an *embedding* of e is a path $P(e) \subseteq G$ connecting x to y. We will compute an embedding of all fake edges in A that has additional useful properties summarized below.

Definition 3. *A* legal embedding *of the set A of fake edges is a collection $\mathcal{P}(A) = \{P(e) \mid e \in A\}$ of paths in G, such that the following hold.*

- *For every edge $e = (x, y) \in A$, path $P(e)$ has endpoints x and y, and moreover, there is a type-1 acceptable cluster $C(e) \in \mathcal{C}_1$ such that $P(e) \setminus \{x, y\}$ is contained in $C(e)$; and*
- *For any pair $e, e' \in A$ of distinct edges, $C(e) \neq C(e')$;*

Note that from the definition of the legal embedding, all paths in $\mathcal{P}(A)$ must be mutually internally disjoint. Finally, we define a decomposition of a graph G into acceptable clusters; this definition is central for the proof of our main result.

Definition 4. *A* decomposition of a graph G into acceptable clusters *consists of:*

- *a planarizing set $\hat{E} \subseteq E(G)$ of edges of G;*
- *a set A of fake edges (where the endpoints of each fake edge are terminals with respect to $\hat{E}$);*
- *a partition $(\mathcal{C}_1, \mathcal{C}_2)$ of all connected components (called clusters) of the resulting graph $(G \setminus \hat{E}) \cup A$ into two subsets, such that every cluster $C \in \mathcal{C}_1$ is a type-1 acceptable cluster;*
- *for every cluster $C \in \mathcal{C}_2$, a planar drawing ψ'_C of C on the sphere, such that C is a type-2 acceptable cluster with respect to ψ'_C; and*
- *a legal embedding $\mathcal{P}(A)$ of all fake edges.*

We denote such a decomposition by $\mathcal{D} = \left(\hat{E}, A, \mathcal{C}_1, \mathcal{C}_2, \{\psi'_C\}_{C \in \mathcal{C}_2}, \mathcal{P}(A)\right)$.

Our first result is the following theorem, whose proof appears in Section 1.5, that allows us to compute a decomposition of the input graph G into acceptable clusters. This result is one of the main technical contributions of our work.

Theorem 1.3.1. *There is an efficient algorithm, that, given a 3-connected n-vertex graph G with maximum vertex degree at most Δ and a planarizing set E' of edges for G, computes a decomposition $\mathcal{D} = \left(E'', A, \mathcal{C}_1, \mathcal{C}_2, \{\psi'_C\}_{C \in \mathcal{C}_2}, \mathcal{P}(A)\right)$ of G into acceptable clusters, such that $E' \subseteq E''$ and $|E''| \leq O((|E'| + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \mathrm{poly}(\Delta \log n))$.*

1.3.2 Canonical Drawings

In this subsection, we assume that we are given a 3-connected n-vertex graph G with maximum vertex degree at most Δ, and a decomposition $\mathcal{D} = \left(E'', A, \mathcal{C}_1, \mathcal{C}_2, \{\psi'_C\}_{C \in \mathcal{C}_2}, \mathcal{P}(A)\right)$ of G into acceptable clusters. Next, we define drawings of G that are "canonical" with respect to the clusters in the decomposition. For brevity of notation, we refer to type-1 and type-2 acceptable clusters as type-1 and type-2 clusters, respectively.

Intuitively, in each such canonical drawing, we require that, for every type-2 cluster $C \in \mathcal{C}_2$, the edges of $C \setminus A_C$ do not participate in any crossings, and for every type-1 acceptable cluster $C \in \mathcal{C}_1$, the edges of C only participate in a small number of crossings (more specifically, we will define a subset $E^{**}(C)$ of edges for each cluster $C \in \mathcal{C}_1$ that are allowed to participate in crossings).

We will define, for every type-1 cluster $C \in \mathcal{C}_1$, a fixed drawing ψ_C, and we will require that, in the final drawing of G, the induced drawing of each such cluster C is precisely ψ_C. For every type-2 cluster $C \in \mathcal{C}_2$, we have already defined a drawing $\psi_{C \setminus A_C}$ of $C \setminus A_C$ – the drawing of $C \setminus A_C$ that is induced by the drawing ψ'_C of C. We will require that the drawing of $C \setminus A_C$ that is induced by the final drawing of G is precisely $\psi_{C \setminus A_C}$. Additionally, for each cluster $C \in \mathcal{C}_1$, and for each bridge $R \in \mathcal{R}_G(C)$, we will define a disc $D(R)$ in the drawing ψ_C of C, and we will require that all vertices and edges of R are drawn inside $D(R)$ in the final drawing of G. Similarly, for each type-2 acceptable cluster $C \in \mathcal{C}_2$, for every bridge $R \in \mathcal{R}_G(C \setminus A_C)$, we define a disc $D(R)$ in the drawing $\psi_{C \setminus A_C}$ of $C \setminus A_C$, and we will require that all vertices and edges of R are drawn inside $D(R)$ in the final drawing of G.

This will allow us to fix the locations of the components of $\mathcal{C}_1 \cup \mathcal{C}_2$ with respect to each other (that is, for each pair $C, C' \in \mathcal{C}_1 \cup \mathcal{C}_2$ of clusters, we will identify a face F in the drawing $\psi_{C \setminus A_C}$ of $C \setminus A_C$, and a face F' in the drawing $\psi_{C' \setminus A_{C'}}$ of $C' \setminus A_{C'}$, such that, in the final drawing φ of the graph G, graph $C' \setminus A_{C'}$ is drawn inside the face F (of the drawing of $C \setminus A$ induced by φ, which is identical to $\psi_{C \setminus A_C}$), and similarly graph $C \setminus A_C$ is drawn inside the face F').

Before we continue, it would be convenient for us to ensure that, for every type-1 cluster $C \in \mathcal{C}_1$, the vertices of $\Gamma(C)$ have degree 1 in C, and degree 2 in G; it would also be convenient for us to ensure that no edge of E'' connects two vertices that lie in the same cluster. In order to ensure these properties, we subdivide some edges of G. Specifically, if $e = (u, v) \in E''$ is an edge with $u, v \in C$, for some cluster $C \in \mathcal{C}_1 \cup \mathcal{C}_2$, then we subdivide the edge (u, v) with two vertices, replacing it with a path (u, u', v', v). The edges (u, u') and (v', v) are then added to set E'' instead of the edge (u, v), and we add a new cluster to $\mathcal{C}_1$, that consists of the vertices u', v', and the edge (u', v'). This transformation ensures that no edge of E'' connects two vertices that lie in the same cluster. Consider now any type-1 cluster $C \in \mathcal{C}_1$. For every edge $e = (u, v) \in E''$ with $u \in V(C)$ and $v \notin V(C)$, we subdivide the edge with a new vertex u', thereby replacing the edge with the path (u, u', v). Vertex u' and edge (u, u') are added to the cluster C, while edge (u', v) replaces the edge (u, v) in set E''. Note that u' now becomes a terminal, and, once all edges of E'' that are incident to the vertices of C are processed, u will no longer be a terminal. Abusing the notation, the final cluster that is obtained after processing all edges of E'' incident to $V(C)$ is still denoted by C. Notice that now the number of terminals that lie in C may have grown by at most a factor Δ, and so $|\Gamma(C)| \leq \mu\Delta$ must hold. Abusing the notation, we will still refer to C as a type-1 acceptable cluster, and we will continue to denote by $\mathcal{C}_1$ the set of all such clusters in the decomposition. Observe that this transformation ensures that every vertex of $\Gamma(C)$ has degree 1 in C and degree 2 in G. Once every cluster $C \in \mathcal{C}_1$ is processed in this manner, we obtain the final graph G'. Observe that $|E''|$ may have increased by at most a constant factor. Notice also that, for every fake edge $e = (x, y) \in A$, the endpoints of e remain terminals in Γ, and the path $P(e) \in \mathcal{P}$ that was used as a legal embedding of the edge e can be converted into a path $P'(e)$ embedding e in the new graph G', by possibly subdividing the first and the last edge of $P(e)$ if needed. If $C(e) \in \mathcal{C}_1$ is the type-1 cluster with $P(e) \setminus \{x, y\} \subseteq C(e)$, then the new vertices that (possibly) subdivide the first and the last edge of $P(e)$ lie in the new cluster $C'(e)$ corresponding to $C(e)$, so $P'(e) \setminus \{x, y\} \subseteq C'(e)$ continues to hold. The resulting path set $\mathcal{P}' = \left\{P'(e) \mid e \in A\right\}$ is a legal embedding of the set of fake edges into G'. Lastly, observe that any drawing of G' on the sphere immediately gives a drawing of G with the same number of crossings. Therefore, to simplify the notation, we will denote the graph G' by G and $\mathcal{P}'$ by $\mathcal{P}$ and we will assume that the decomposition $\mathcal{D}$ of G into acceptable clusters has the following two additional properties:

P1. For every edge $e \in E''$, the endpoints of e lie in different clusters of $\mathcal{C}_1 \cup \mathcal{C}_2$; and

P2. For every type-1 cluster $C \in \mathcal{C}_1$, for every terminal $t \in \Gamma(C)$, the degree of t in C is 1, and its degree in G is 2.

We now proceed to define canonical drawings of the graph G with respect to the clusters of $\mathcal{C}_1 \cup \mathcal{C}_2$.

Canonical Drawings for Type-2 Acceptable Clusters

Consider any type-2 cluster $C \in \mathcal{C}_2$. Recall that the decomposition $\mathcal{D}$ into acceptable clusters defines a planar drawing ψ'_C of C on the sphere, that induces a planar drawing $\psi_{C \setminus A_C}$ of $C \setminus A_C$ on the sphere. Recall that the Bridge Consistency Property of type-2 acceptable clusters ensures that, for every bridge $R \in \mathcal{R}_G(C \setminus A_C)$, there is a face F of the drawing $\psi_{C \setminus A_C}$, such that the vertices of $L(R)$ lie on the boundary of F (we note that face F is not uniquely defined; we break ties arbitrarily). Since graph $C \setminus A_C$ is 2-connected, the boundary of face F is a simple cycle, whose image is a simple closed curve. We denote by $D(R)$ the disc corresponding to the face F, so the boundary of $D(R)$ is the simple closed curve that serves as the boundary of the face F. Notice that the resulting set $\{D(R)\}_{R \in \mathcal{R}_G(C \setminus A_C)}$ of discs has the following properties:

D1. If $R \neq R'$ are two distinct bridges in $\mathcal{R}_G(C \setminus A_C)$, then either $D(R) = D(R')$, or $D(R) \cap D(R')$ only contains points on the boundaries of the two discs; and

D2. For every bridge $R \in \mathcal{R}_G(C \setminus A_C)$, the vertices of $L(R)$ lie on the boundary of $D(R)$ in the drawing $\psi_{C \setminus A_C}$.

We are now ready to define canonical drawings with respect to type-2 clusters.

Definition 5. *Let φ be any drawing of the graph G on the sphere. We say that the drawing φ is* canonical *with respect to a type-*2 *cluster $C \in \mathcal{C}_2$ iff:*

- *the drawing of $C \setminus A_C$ induced by φ is identical to $\psi_{C \setminus A_C}$ (but its orientation may be different);*
- *the edges of $C \setminus A_C$ do not participate in any crossings in φ; and*
- *for every bridge $R \in \mathcal{R}_G(C \setminus A_C)$, all vertices and edges of R are drawn in the interior of the disc $D(R)$ (that is defined with respect to the drawing $\psi_{C \setminus A_C}$ of $C \setminus A_C$).*

Canonical Drawings for Type-1 Acceptable Clusters

For convenience, we denote $\mathcal{C}_1 = \left\{C_1, \ldots, C_q\right\}$. We fix an arbitrary optimal drawing φ^* of the graph G. For each $1 \leq i \leq q$, we denote by χ_i the set of all crossings (e, e') such that either e or e' (or both) are edges of $E(C_i)$. The following observation is immediate.

Observation 1.3.2. $\sum_{i=1}^{q} |\chi_i| \leq 2 \cdot \mathrm{cr}(\varphi^*) = 2 \cdot \mathsf{OPT}_{\mathsf{cr}}(G)$.

We use the following theorem in order to fix a drawing of each type-1 acceptable cluster C_i; the proof appears in Section 1.6.

Theorem 1.3.3. *There is an efficient algorithm that, given a type-1 cluster $C_i \in \mathcal{C}_1$, computes a drawing ψ_{C_i} of C_i on the sphere with $O\left((|\chi_i|+1) \cdot \operatorname{poly}(\Delta \log n)\right)$ crossings, together with a set $E^*(C_i) \subseteq E(C_i)$ of at most $O\left((|\chi_i|+1) \cdot \operatorname{poly}(\Delta \log n)\right)$ edges, such that graph $C_i \setminus E^*(C_i)$ is connected, and the drawing of $C_i \setminus E^*(C_i)$ induced by ψ_{C_i} is planar. Additionally, the algorithm computes, for every bridge $R \in \mathcal{R}_G(C_i)$, a closed disc $D(R)$, such that:*

- *the vertices of $L(R)$ are drawn on the boundary of $D(R)$ in ψ_{C_i};*
- *the interior of $D(R)$ is disjoint from the drawing ψ_{C_i}; and*
- *for every pair $R, R' \in \mathcal{R}_G(C_i)$ of bridges, either $D(R) = D(R')$, or $D(R) \cap D(R') = \emptyset$.*

Note that in particular, Properties (D1) and (D2) also hold for the discs in $\{D(R)\}_{R \in \mathcal{R}_G(C)}$. For each type-1 cluster $C_i \in \mathcal{C}_1$, let $E^{**}(C_i) \subseteq E(C_i)$ be the set of all edges of C_i that participate in crossings in ψ_{C_i}. Clearly, $|E^{**}(C_i)| \le O(\operatorname{cr}(\psi_{C_i})) \le O((|\chi_i|+1) \operatorname{poly}(\Delta \log n))$. Let $E^* = \bigcup_{C_i \in \mathcal{C}_1} E^{**}(C_i)$. Then, from Observation 1.3.2 and Theorem 1.3.1:

$$\begin{aligned} |E^*| &\le \sum_{C_i \in \mathcal{C}_1} O((|\chi_i|+1) \cdot \operatorname{poly}(\Delta \log n)) \\ &\le O\left((\mathsf{OPT}_{\mathsf{cr}}(G) + |E''|) \operatorname{poly}(\Delta \log n)\right) \\ &\le O\left((\mathsf{OPT}_{\mathsf{cr}}(G) + |E'|) \operatorname{poly}(\Delta \log n)\right). \end{aligned}$$

We now define canonical drawings with respect to type-1 clusters.

Definition 6. *Let φ be any drawing of the graph G on the sphere, and let $C_i \in \mathcal{C}_1$ be a type-1 cluster. We say that φ is a* canonical drawing *with respect to C_i, iff:*

- *the drawing of C_i induced by φ is identical to ψ_{C_i} (but orientation of the two drawings may be different); and*

- *for every bridge $R \in \mathcal{R}_G(C_i)$, all vertices and edges of R are drawn in the interior of the disc $D(R)$ (that is defined with respect to the drawing ψ_{C_i} of C_i).*

Notice that the definition implies that the only edges of C_i that participate in crossings of φ are the edges of $E^{**}(C_i)$.

Obtaining a Canonical Drawing

Our next result shows that there exists a near-optimal drawing of the graph G that is canonical with respect to all clusters. The proof of the following theorem appears in Section 1.7.

Theorem 1.3.4. *There is an efficient algorithm, that, given, as input:*

- *an n-vertex graph G of maximum vertex degree at most Δ;*
- *an arbitrary drawing φ of G;*
- *a decomposition $\mathcal{D} = \left(E'', A, \mathcal{C}_1, \mathcal{C}_2, \{\psi_C\}_{C \in \mathcal{C}_2}, \mathcal{P}(A)\right)$ of G into acceptable clusters for which Properties (P1) and (P2) hold;*
- *a drawing ψ_{C_i} and an edge set $E^*(C_i)$ for each cluster $C_i \in \mathcal{C}_1$ as defined above; and*
- *for each cluster $C \in \mathcal{C}_1 \cup \mathcal{C}_2$, a collection $\{D(R)\}_{R \in \mathcal{R}_G(C \setminus A_C)}$ of discs on the sphere with Properties (D1) and (D2);*

computes a drawing φ' of G on the sphere with $O\left((|E''| + \mathrm{cr}(\varphi)) \cdot \mathrm{poly}(\Delta \log n)\right)$ crossings, such that φ' is canonical with respect to every cluster $C \in \mathcal{C}_1 \cup \mathcal{C}_2$.

We note that for our purposes, an existential variant of the above theorem, that shows that a drawing φ' with the required properties exists, is sufficient. We provide the proof of the stronger constructive result in case it may be useful for future work on the problem.

1.3.3 Reduction to Crossing Number with Rotation System – Proof of Theorem 1.1.1 for 3-Connected Graphs

In this section we provide a reduction from Minimum Crossing Number in 3-connected graphs to MCNwRS, completing the proof of Theorem 1.1.1 in the case where the input graph G is 3-connected. We extend this proof to general graphs in Section 1.8. Recall that Kawarabayashi and Sidiropoulos [24] provide an efficient $O(\text{poly}(\Delta \log n))$-approximation algorithm for the Minimum Planarization problem. Since, for every graph G, there is a planarizing set E^* containing at most $\mathsf{OPT}_{\mathsf{cr}}(G)$ edges, we can use their algorithm in order to compute, for the input graph G, a planarizing edge set E' of cardinality $O(\mathsf{OPT}_{\mathsf{cr}}(G) \cdot \text{poly}(\Delta \log n))$. We then use Theorems 1.3.1 and 1.3.3 to compute another planarizing edge set E'' of cardinality $O(\mathsf{OPT}_{\mathsf{cr}}(G) \cdot \text{poly}(\Delta \log n))$ for G, the collection $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$ of clusters, together with their drawings ψ_C, and we use families $\{D(R)\}_{R \in \mathcal{R}_G(C)}$ of discs for all $C \in \mathcal{C}$ that we have computed. We will not need fake edges anymore, so for every type-2 cluster $C \in \mathcal{C}_2$, we let $C' = C \setminus A_C$, and we let $\psi_{C'}$ be the planar drawing of C' that was used in the definition of the canonical drawing. For a type-1 cluster $C \in \mathcal{C}_1$, consider the drawing ψ_C of C given by Theorem 1.3.3. We let C' be a graph obtained from C, by placing a vertex on every crossing of a pair of edges in ψ_C. Therefore, graph C' is planar, and we denote by $\psi_{C'}$ its planar drawing that is induced by ψ_C. We still denote by $\Gamma(C')$ the set of all vertices of C' that serve as endpoints of the edges of E''. Consider the graph G', that is obtained by taking the union of all clusters in $\left\{C' \mid C \in \mathcal{C}\right\}$ and the edges in E''. Suppose we compute a drawing φ of G' with z crossings, such that the only edges that participate in the crossings are the edges of E'', and for every cluster $C \in \mathcal{C}$, the drawing of C' induced by φ is identical to $\psi_{C'}$. Then we can immediately obtain a drawing φ' of G with $O(z + \mathsf{OPT}_{\mathsf{cr}}(G)\,\text{poly}(\Delta \log n))$ crossings, where the additional crossings arise because we replace, for every type-1 cluster $C \in \mathcal{C}_1$, the planar drawing $\psi_{C'}$ of C' with the (possibly non-planar) drawing ψ_C of C. Let $\mathcal{C}'_1 = \left\{C' \mid C \in \mathcal{C}_1\right\}$, $\mathcal{C}'_2 = \left\{C' \mid C \in \mathcal{C}_2\right\}$, and $\mathcal{C}' = \mathcal{C}'_1 \cup \mathcal{C}'_2$. Since we do not use the original clusters in $\mathcal{C}$ in the remainder of this subsection, for simplicity of notation, we denote $\mathcal{C}'_1, \mathcal{C}'_2$ and $\mathcal{C}'$ by $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}$, and we will use notation $C \in \mathcal{C}$ instead of C'. Recall that for every cluster $C \in \mathcal{C}$ we are now given a fixed planar drawing ψ_C. In order to reduce the problem to MCNwRS, we use the Cluster Placement problem, that we define next, as an intermediate problem.

Cluster Placement Problem. In the Cluster Placement problem, we are given a collection $\hat{\mathcal{C}}$ of disjoint connected planar graphs (that we call clusters). For every cluster $\hat{C} \in \hat{\mathcal{C}}$, we are also given a planar drawing $\psi_{\hat{C}}$ of $\hat{C}$ on the sphere. We denote by $\mathcal{F}_{\hat{C}}$ the set of all faces of this drawing. Additionally, for every ordered pair $(\hat{C}_1, \hat{C}_2) \in \hat{\mathcal{C}}$ of clusters, we are given a face $F_{\hat{C}_1}(\hat{C}_2) \in \mathcal{F}_{\hat{C}_1}$. The goal is to compute a planar drawing φ of $\bigcup_{\hat{C} \in \hat{\mathcal{C}}} \hat{C}$ on the sphere such that, for every cluster $\hat{C}_1 \in \hat{\mathcal{C}}$, the drawing of $\hat{C}_1$ induced by φ is identical to $\psi_{\hat{C}_1}$, and moreover, for every cluster $\hat{C}_2 \in \hat{\mathcal{C}} \setminus \left\{\hat{C}_1\right\}$, the drawing of $\hat{C}_2$ in φ is contained in the face $F_{\hat{C}_1}(\hat{C}_2)$ of the drawing of $\hat{C}_1$ in φ.

The proof of the following simple theorem is deferred to Section 1.9.2.

Theorem 1.3.5. *There is an efficient algorithm, that, given an instance of the* Cluster Placement *problem, finds a feasible solution for the problem, if such a solution exists.*

We note that the current collection $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$ of clusters that we obtained for the instance G of the Minimum Crossing Number problem naturally defines an instance of the Cluster Placement problem. For every cluster $C \in \mathcal{C}$, we have already defined a fixed planar drawing ψ_C of C on the sphere. Consider now some ordered pair $(C_1, C_2) \in \mathcal{C}$ of clusters. Then there must be some bridge $R \in \mathcal{R}_{G'}(C_1)$, such that $C_2 \subseteq R$. Recall that we have defined a disc $D(R)$ corresponding to the bridge R in the drawing ψ_{C_1} of C_1, and that the images of the edges and vertices of C_1 in ψ_{C_1} are disjoint from the interior of the disc $D(R)$. Therefore, there is some face F in the drawing ψ_{C_1} of C_1 with $D(R) \subseteq F$. We then set $F_{C_1}(C_2)$ to be this face F. This defines a valid instance of the Cluster Placement problem. Moreover, since

Theorem 1.3.4 guarantees the existence of a canonical drawing of the graph G, this problem has a feasible solution. In fact Theorem 1.3.4 provides the following stronger guarantees:

Observation 1.3.6. *There is a drawing φ of graph G' with $O(\mathsf{OPT}_{\mathsf{cr}}(G)\,\mathrm{poly}(\Delta \log n))$ crossings, such that for every cluster $C \in \mathcal{C}$, the edges of C do not participate in any crossings, and the drawing of C induced by φ is identical to ψ_C (but the orientation may be arbitrary). Moreover, for any ordered pair $(C_1, C_2) \in \mathcal{C}$ of clusters, the image of C_2 in φ is contained in the interior of the face $F_{C_1}(C_2)$ of the drawing of C_1 in φ.*

We use the algorithm from Theorem 1.3.5 in order to compute a feasible solution to this instance of the $\mathsf{Cluster\ Placement}$ problem, obtaining a drawing $\tilde{\varphi}$ of $\bigcup_{C\in\mathcal{C}} C$ on the sphere. In order to compute a final drawing of G' (and hence of G), it is enough to add the drawings of the edges of E'' into $\tilde{\varphi}$. We do so by defining an instance of the MCNwRS problem.

Defining Instances of MCNwRS. Let $\mathcal{F}$ be the set of all faces in the drawing $\tilde{\varphi}$ of the graph $\bigcup_{C\in\mathcal{C}} C$. For every face $F \in \mathcal{F}$, let $\mathcal{H}(F) \subseteq \mathcal{C}$ be the set of all clusters $C \in \mathcal{C}$, such that at least one terminal of $\Gamma(C)$ (that is, endpoint of an edge of E'' that lies in C) lies on the boundary of the face F.

We associate, with each face $F \in \mathcal{F}$, an instance (G^F, Σ^F) of the MCNwRS problem, as follows. Let $E^F \subseteq E''$ be the set of all edges whose both endpoints lie on the boundary of the face F in $\tilde{\varphi}$. In order to construct the graph G^F, we start with the union of the clusters $C \in \mathcal{H}(F)$, and add the edges of E^F to the resulting graph. Then we contract every cluster $C \in \mathcal{H}(F)$ into a vertex $v(C)$, keeping parallel edges and deleting self-loops. This concludes the definition of the graph G^F. We now define a rotation system for G^F. Consider any vertex $v = v(C) \in V(G^F)$, and let $\delta(v)$ be the set of all edges that are incident to $v(C)$ in G^F. If $C \in \mathcal{C}_1$, then $|\delta(v)| \leq |\Gamma(C)| \leq \mathrm{poly}(\Delta \log n)$. We define $\mathcal{O}_v$ to be an arbitrary ordering of the edges in $\delta(v)$. Assume now that $C \in \mathcal{C}_2$. Recall that graph C must be 2-connected, so the intersection of the boundary of F with C is a simple cycle, that we denote by $K^F(C)$. We denote by $\Gamma^F(C)$ the set of all vertices of $\Gamma(C)$ that lie on the cycle $K^F(C)$. From Observation 1.3.6, the vertices of $\Gamma(C)$ that serve as endpoints of the edges of $\delta(v)$ must belong to $\Gamma^F(C)$. We let $\tilde{\mathcal{O}}^F(C)$ denote the circular ordering of the vertices of $\Gamma^F(C)$ along the cycle $K^F(C)$. The ordering $\mathcal{O}_v$ of the edges of $\delta(v)$ is determined by the ordering of their endpoints in $\tilde{\mathcal{O}}^F(C)$: edges that are incident to the same vertex of $K^F(C)$ appear consecutively in $\mathcal{O}_v$ in an arbitrary order. The ordering of the edges that are incident to different vertices of $\Gamma^F(C)$ follows the ordering $\tilde{\mathcal{O}}^F(C)$. We then let $\Sigma^F = \{\mathcal{O}_v\}_{v \in V(G^F)}$. This completes the definition of instance (G^F, Σ^F) of MCNwRS. Since Theorem 1.1.1 calls for a single instance of MCNwRS, we let G'' be the disjoint union of the graphs in $\left\{G^F\right\}_{F\in\mathcal{F}}$, and we let $\Sigma = \bigcup_{F\in\mathcal{F}} \Sigma^F$. This defines the final instance (G'', Σ) of the MCNwRS problem. Notice that $E(G'') \subseteq E''$ and so $|E(G'')| \leq O(\mathsf{OPT}_{\mathsf{cr}}(G)\,\mathrm{poly}(\Delta \log n))$. We need the following observation.

Observation 1.3.7. $E'' = \bigcup_{F\in\mathcal{F}} E^F$.

Proof. From the definition of edge sets E^F, $\bigcup_{F\in\mathcal{F}} E^F \subseteq E''$, so it is enough to show that $E'' \subseteq \bigcup_{F\in\mathcal{F}} E^F$. Assume for contradiction that this is not the case. Then there is some edge $e = (u, v) \in E''$, such that no face of $\mathcal{F}$ contains both u and v on its boundary. Let C' be the cluster containing u and C'' the cluster containing v; recall that $C' \neq C''$ must hold. Then there must be some cluster C and a cycle K in C, such that $u, v \notin K$, and the image of K in $\tilde{\varphi}$ separates the images of u and v. If $C = C'$, then we let F_1 be any face in the drawing of C that is incident to u; otherwise, we let $F_1 = F_C(C')$. Similarly, if $C = C''$, then we let F_2 be any face in the drawing of C that is incident to v; otherwise, we let $F_2 = F_C(C')$. Notice that $F_1 \neq F_2$, and moreover, since K separates u from v, if $C = C'$ then u does not lie on the boundary of F_2 (and similarly, if $C = C''$, then v does not lie on the boundary of F_1). But that means that, in the drawing φ of graph G' that is given by Observation 1.3.6, there is some cycle $K' \subseteq C$ that separates the image of u from the image of v in φ. But then the image of edge e must cross the image of some edge of C in φ, which is impossible. $\square$

Next, we show that the final instance (G', Σ) of MCNwRS has a sufficiently cheap solution. The proof of the next lemma appears in Section 1.9.3.

Lemma 1.3.8. *There is a solution to instance (G'', Σ) of* MCNwRS *of value $O(\mathsf{OPT}_{\mathsf{cr}}(G) \cdot \text{poly}(\Delta \log n))$.*

Lastly, in order to complete the proof of Theorem 1.1.1, it is enough to show an efficient algorithm, that, given any solution to instance (G'', Σ) of MCNwRS of value X, computes a drawing of G' with $O\left((X + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \text{poly}(\Delta \log n)\right)$ crossings.

Obtaining the Final Drawing of G'. We assume that we are given a solution $\hat{\varphi}$ to instance (G'', Σ) of MCNwRS, whose value is denoted by X. Since graph G'' is the disjoint union of graphs $\left\{G^F\right\}_{F \in \mathcal{F}}$, we can use $\hat{\varphi}$ to obtain, for each face $F \in \mathcal{F}$, a solution $\hat{\varphi}^F$ to instance (G^F, Σ^F) of MCNwRS, of value X^F, such that $\sum_{F \in \mathcal{F}} X^F \leq X$. Intuitively, ideally, we would like to start with the drawing $\tilde{\varphi}$ of $\bigcup_{C \in \mathcal{C}} C$ given by the solution to the $\mathsf{Cluster\ Placement}$ problem, and then consider the faces $F \in \mathcal{F}$ one-by-one. For each such face, we would like to use the drawing $\hat{\varphi}^F$ of graph G^F in order to insert the images of the edges of E^F into the face F. Since, from Observation 1.3.7, $E'' = \bigcup_{F \in \mathcal{F}} E^F$, once every face of $\mathcal{F}$ is processed, all the edges of E'' are inserted into the drawing, and we obtain a valid drawing of graph G'. There is one difficulty with using this approach. Recall that, a solution $\hat{\varphi}^F$ to instance (G^F, Σ^F) guarantees that for every vertex $v \in V(G^F)$, the images of the edges of $\delta(v)$ enter v in an ordering identical to $\mathcal{O}_v$. However, the *orientation* of this ordering may be arbitrary. In other words, in order to insert the edges of E^F into the face F of the drawing of φ', by copying their drawings in $\hat{\varphi}^F$, we may need to *flip* the drawings of some clusters $C \in \mathcal{H}(F)$. Since each cluster may belong to several sets $\mathcal{H}(F)$, we need to do this carefully.

Consider the drawing $\tilde{\varphi}$ of $\bigcup_{C \in \mathcal{C}} C$. Consider any cluster $C \in \mathcal{C}$ and any face $F \in \mathcal{F}$, such that $C \in \mathcal{H}(F)$. As before, we let $\Gamma^F(C) \subseteq \Gamma(C)$ be the set of terminals that belong to C and lie on the boundary of the face F. Next, we define a disc $D^F(C)$, as follows. If

$C \in \mathcal{C}_1$, then let $\gamma^F(C)$ be a simple closed curve that contains every terminal in $\Gamma^F(C)$, and separates the drawing of $C \setminus \Gamma^F(C)$ from the drawing of every cluster $C' \in \mathcal{H}(F) \setminus \{C\}$. If $C \in \mathcal{C}_2$, then we let $\gamma^F(C)$ be the image of the cycle $K^F(C)$ in $\tilde{\varphi}$. We then let $D^F(C)$ be a disc, whose boundary is $\gamma^F(C)$, that contains the drawing of C in $\tilde{\varphi}$. Notice that for every cluster $C' \in \mathcal{H}(F) \setminus \{C\}$, the drawing of C' in $\tilde{\varphi}$ is disjoint from $D^F(C)$. Notice also that, if $C \in \mathcal{C}_2$, then the ordering of the terminals in $\Gamma^F(C)$ on the boundary of $D^F(C)$ is identical to $\tilde{\mathcal{O}}^F(C)$.

We now proceed as follows. First, we describe a procedure ProcessFace, that, intuitively, will allow us to insert, for a given face $F \in \mathcal{F}$, all edges of E^F into the drawing; this may require flipping drawings in some discs $D^F(C)$, for $C \in \mathcal{H}(F)$. We then show an algorithm that builds on this procedure in order to compute a drawing of G'.

Procedure ProcessFace. The input to procedure ProcessFace is a face $F \in \mathcal{F}$, and a collection $\left\{D^F(C)\right\}_{C \in \mathcal{H}(F)}$ of disjoint discs (intuitively, disc $D^F(C)$ already contains a drawing of some parts of the graph G, as defined above, but in this procedure we do not modify parts of the graph drawn inside the disc, and consider these parts as being fixed; we may however flip the drawing that is contained in $D(C)$). Additionally, for every cluster $C \in \mathcal{H}(F)$, we are given a drawing of the terminals in $\Gamma^F(C)$ on the boundary of the disc $D^F(C)$. We require that, if $C \in \mathcal{C}_2$, then the circular ordering of the terminals in $\Gamma^F(C)$ on the boundary of the disc $D^F(C)$ is identical to $\tilde{\mathcal{O}}^F(C)$. We are also given a cluster $C^* \in \mathcal{H}(F)$, whose orientation is fixed (that is, we are not allowed to flip the disc $D^F(C^*)$). The procedure inserts the edges of E^F into this drawing, while possibly flipping some discs $D^F(C)$.

In order to execute the procedure, we start with the solution $\hat{\varphi}^F$ to instance (G^F, Σ^F). For every vertex $v \in V(G^F)$, we consider a small disc $\eta(v)$ around the drawing of v in $\hat{\varphi}^F$. We also define a smaller disc $\eta'(v) \subseteq \eta(v)$ that is contained in the interior of $\eta(v)$ and contains the image of v. For every edge $e = (v, v')$, we truncate the image of e, so that it originates at some point $p_v(e)$ on the boundary of $\eta(v)$ and terminates at some point $p_{v'}(e)$ on the boundary of $\eta(v')$.

Consider now some vertex $v(C) \in V(G^F)$, whose corresponding cluster C lies in $\mathcal{C}_2$. Recall that for every terminal $t \in \Gamma^F(C)$, there must be a contiguous segment $\sigma(t)$ on the boundary of $\eta(v)$ that contains all points $\left\{p_{v(C)}(e)\right\}$, where $e \in \delta(v)$ is an edge that is incident to t, so that all resulting segments in $\left\{\sigma_t \mid t \in \Gamma^F(C)\right\}$ are disjoint. The circular ordering of these segments along the boundary of $\eta(v)$ is identical to $\tilde{\mathcal{O}}^F(C)$. We place images of the terminals in $\Gamma^F(C)$ on the boundary of $\eta'(v)$, in the circular order that is identical to $\tilde{\mathcal{O}}^F(C)$, whose orientation is the same as the orientation of the ordering of the segments in $\left\{\sigma_t \mid t \in \Gamma^F(C)\right\}$. If the orientation of this ordering is identical to the orientation of the ordering of terminals of $\Gamma^F(C)$ on the boundary of the disc $D^F(C)$, then we say that cluster C *agrees* with the orientation of $\hat{\varphi}^F$, and otherwise we say that it disagrees with it. We can assume without loss of generality that, if cluster $C^* \in \mathcal{C}_2$, then it agrees with the orientation of $\hat{\varphi}^F$, since otherwise we can flip the drawing $\hat{\varphi}^F$.

In order to insert the edges of E^F into the current drawing, we will do the opposite: we will "insert" the discs $D^F(C)$ into the discs $\eta'(v(C))$ in the drawing $\hat{\varphi}^F$. Specifically, for every cluster $C \in \mathcal{C}$, if C agrees with the orientation of $\hat{\varphi}^F$, then we insert the disc $D^F(C)$ into the disc $\eta'(v(C))$ in the current drawing $\hat{\varphi}^F$, so that the images of the terminals of $\Gamma^F(C)$ coincide (it may be convenient to think of the disc $D^F(C)$ as containing a drawing of C and maybe some additional subgraphs of G'). If C disagrees with the orientation of $\hat{\varphi}^F$, then we first create a mirror image of the disc $D^F(C)$ (that will result in flipping whatever drawing currently appears in $D^F(C)$), and then insert the resulting disc into the disc $\eta(v(C))$ in the current drawing $\hat{\varphi}^F$, so that the images of the terminals of $\Gamma^F(C)$ coincide. In either case, we can extend the drawings of the edges of $\delta(v(C))$ inside $\eta(v(C)) \setminus \eta'(v(C))$, so that for every terminal $t \in \Gamma^F(C)$, the drawing of every edge e that is incident to t terminates at the image of t, without introducing any crossings. Lastly, if $C \in \mathcal{C}_1$, then we simply insert the disc $D^F(C)$ into the disc $\eta'(C)$. We extend the drawings of the edges of $\delta(v(C))$ inside

$\eta(v(C)) \setminus \eta'(v(C))$, so that for every terminal $t \in \Gamma^F(C)$, the drawing of every edge e that is incident to t terminates at the image of t, while introducing at most $|\Gamma^F(C)|^2$ new crossings. This completes the description of Procedure $\mathsf{ProcessFace}$.

We are now ready to complete the drawing of the graph G'. Our algorithm performs a number of iterations. In each iteration i we will fix an orientation of some subset $\mathcal{C}^i \subseteq \mathcal{C}$ of clusters. We maintain the invariant that for every cluster $C \in \mathcal{C}^i$, if $F \in \mathcal{F}$ is any face with $C \in \mathcal{H}(F)$ that has not been processed yet, then no cluster of $(\mathcal{C}^1 \cup \mathcal{C}^2 \cup \cdots \cup \mathcal{C}^i) \setminus \{C\}$ lies in $\mathcal{H}(F)$. We let $\mathcal{C}^0$ consist of a single arbitrary cluster $C_0 \in \mathcal{C}$.

In order to execute the first iteration, we let $F \in \mathcal{F}$ be any face with $C_0 \in \mathcal{H}(F)$. We run Procedure $\mathsf{ProcessFace}$ on face F, with cluster $C^* = C_0$. Notice that the outcome of this procedure can be used in order to insert the edges of E^F into the current drawing $\tilde{\varphi}$ of $\bigcup_{C \in \mathcal{C}} C$, after possibly flipping the images inside some discs in $\{D(C)\}_{C \in \mathcal{H}(F) \setminus \{C_0\}}$. We then let $\mathcal{C}^1$ contain all clusters in $\mathcal{H}(F)$. Notice that the invariant holds for this definition of set $\mathcal{C}^1$. For each cluster $C \in \mathcal{C}^1$, its orientation is fixed from now on, and the drawing of C will never be modified again.

In order to execute the ith iteration, we start with $\mathcal{C}^i = \emptyset$. We consider each cluster $C \in \mathcal{C}_{i-1}$, one-by-one. For each such cluster C, for every face $F \in \mathcal{F}$ with $C \in \mathcal{H}(F)$, that has not been processed yet, we apply Procedure $\mathsf{ProcessFace}$ to face F, with $C = C^*$. As before, this procedure can be used in order to insert all edges of E^F into the current drawing, after, possibly, flipping the images contained in some discs $D^F(C')$, for $C' \in \mathcal{H}(F) \setminus \{C\}$. Notice, however, that from our invariant, none of the clusters corresponding to these discs may belong to $\mathcal{C}^1 \cup \cdots \cup \mathcal{C}^{i-1}$. We then add, to set $\mathcal{C}^i$, all clusters of $\mathcal{H}(F) \setminus \{C\}$. It is easy to verify that the invariant continues to hold. Once every face in $\mathcal{F}$ is processed, we have inserted all edges of E'' into φ', and obtain a final drawing φ'' of the graph G'.

We now bound the number of crossings in G'. In addition to the crossings that were present in the drawings $\hat{\varphi}^F$, for $F \in \mathcal{F}$, we may have added, for every cluster $C \in \mathcal{C}_1$, at most

$O(\Delta^2|\Gamma(C)|^2)$ new crossings of edges that are incident to the terminals of C (this bound follows the same reasonings as those in the proof of Lemma 1.3.8). Since, for every cluster $C \in \mathcal{C}_1$, $|\Gamma(C)| \leq O(\text{poly}(\Delta \log n))$, we get that the total number of crossings in the drawing φ'' of G' is at most:

$$
\begin{aligned}
X + O\Big(\sum_{C \in \mathcal{C}_1} \Delta |\Gamma(C)|^2 \Big) &\leq X + O(|\Gamma| \,\text{poly}(\Delta \log n)) \\
&\leq X + O(|E''| \,\text{poly}(\Delta \log n)) \\
&\leq X + O(\mathsf{OPT}_{\mathsf{cr}}(G) \,\text{poly}(\Delta \log n)).
\end{aligned}
$$

Note that drawing φ'' of G' immediately induces a drawing of the original graph G, where the number of crossings is bounded by $\text{cr}(\varphi'')$ plus the sum, over all original type-1 clusters $C \in \mathcal{C}_1$, of the number of crossings in the original drawing ψ_C of C (recall that we have replaced all such crossings with vertices in G'). However, the total number of all such additional crossings, as shown already, is bounded by $O(\mathsf{OPT}_{\mathsf{cr}}(G) \,\text{poly}(\Delta \log n))$, and so overall, the total number of crossings in the final drawing of G is bounded by $X + O(\mathsf{OPT}_{\mathsf{cr}}(G) \,\text{poly}(\Delta \log n))$. This completes the proof of Theorem 1.1.1 for the special case where the input graph G is 3-connected. We extend the proof to general graphs in Section 1.8.

1.4 Block Decompositions and Embedding of Fake Edges

In this section we provide some definitions and results on Block Decompositions (mostly from previous work), that will be later in Sections 1.5–1.8. Let G be a 2-connected graph. A 2-*separator* for G is a pair (u, v) of vertices, such that graph $G \setminus \{u, v\}$ is not connected. Most definitions in this section are from [12].

Definition 7. *Let $G = (V, E)$ be a 2-connected graph. A subgraph $B = (V', E')$ of G is called a* block *iff:*

- *$V \setminus V' \neq \emptyset$ and $|V'| \geq 3$;*

- *There are two distinct vertices* $u, v \in V'$, *called* block end-points *and denoted by* $I(B) = (u, v)$, *such that there are no edges from* $V \setminus V'$ *to* $V' \setminus \{u, v\}$ *in* G. *All other vertices of* B *are called* inner vertices*;*

- B *is the subgraph of* G *induced by* V', *except that it* **does not** *contain the edge* (u, v) *even if it is present in* G.

Notice that, if G is a 2-connected graph, then every 2-separator (u, v) of G defines at least two internally disjoint blocks B', B'' with $I(B'), I(B'') = (u, v)$. If B is a block with endpoints u and v, then the *complement* of B, denoted by B^c, is the sub-graph of G induced by the vertices of $(V(G) \setminus V(B)) \cup \{u, v\}$. Notice that B^c is itself a block, unless edge $e = (u, v)$ belongs to B^c; in the latter case, $B^c \setminus \{e\}$ is a block.

1.4.1 Block Decomposition of 2-Connected Graphs

Let $\mathcal{L}$ be a collection of sub-graphs of G. We say that $\mathcal{L}$ is *laminar* iff for every pair $H, H' \in \mathcal{L}$ of subgraphs, either $H \cap H' = \emptyset$, or $H \subseteq H'$, or $H' \subseteq H$ hold. Given a laminar collection $\mathcal{L}$ of subgraphs of G with $G \in \mathcal{L}$, we can associate a tree $\tau = \tau(\mathcal{L})$ with it, called a *decomposition tree*, as follows. For every graph $H \in \mathcal{L}$, there is a vertex $v(H)$ in τ. The tree is rooted at the vertex $v(G)$. For every pair $H, H' \in \mathcal{L}$ of subgraphs, such that $H \subsetneq H'$, and there is no other graph $H'' \in \mathcal{L} \setminus \left\{H, H'\right\}$ with $H \subsetneq H'' \subsetneq H'$, there is an edge $(v(H), v(H'))$ in the tree, and $v(H')$ is the parent of $v(H)$ in τ.

Let G be a 2-connected graph, and assume that we are given a laminar family of sub-graphs of G with $G \in \mathcal{L}$. Let $\tau = \tau(\mathcal{L})$ be the decomposition tree associated with $\mathcal{L}$. Assume further that every graph $B \in \mathcal{L} \setminus \{G\}$ is a block. For each such graph $B \in \mathcal{L}$, we define a new graph, $\tilde{B}$; this definition is used throughout the paper. The edges of $\tilde{B}$ will be classified into "fake" edges and "real" edges, where every real edge of $\tilde{B}$ is an edge of B. In order to obtain graph $\tilde{B}$, we start with the graph B. We then consider every child vertex $v(B')$ of

$v(B)$ in $\tau(\mathcal{L})$ one-by-one. For each such child vertex $v(B')$, we delete all edges and vertices of B' from B, except for the endpoints $I(B')$. If the current graph $\tilde{B}$ does not contain an edge connecting the endpoints of B', then we add such an edge as a fake edge. Consider now the graph $\tilde{B}$ obtained after processing every child vertex of $v(B)$ in tree $\tau(\mathcal{L})$. If $B \neq G$, then we add a fake edge connecting endpoints of B to $\tilde{B}$ (notice that, from the definition of a block, $\tilde{B}$ may not contain a real edge connecting the endpoints of B). This completes the definition of the graph $\tilde{B}$. Observe that by our construction, $\tilde{B}$ has no parallel edges. The definition of $\tilde{B}$ depends on the family $\mathcal{L}$, so when using it we will always fix some such family. We denote by $A_{\tilde{B}}$ the set of all fake edges in $\tilde{B}$. We also denote by $e^*_{\tilde{B}}$ the unique fake edge connecting the endpoints of B; if no such edge exists, then $e^*_{\tilde{B}}$ is undefined. We refer to $e^*_{\tilde{B}}$ as the *fake parent-edge of* $\tilde{B}$.

We let $\mathcal{N}(B)$ be a collection of sub-graphs of G that contains the graph B^c – the complement of the block B, and additionally, for every child vertex $v(B')$ of $v(B)$ in the tree τ, the block B'. Observe that every fake edge $e = (u, v)$ of $\tilde{B}$ is associated with a distinct graph $B_e \in \mathcal{N}(B)$, where $I(B_e) = (u, v)$. We are now ready to define a block decomposition.

Definition 8. *Let G be a 2-connected graph, let $\mathcal{L}$ be a laminar family of sub-graphs of G with $G \in \mathcal{L}$, and let $\tau = \tau(\mathcal{L})$ be the decomposition tree associated with $\mathcal{L}$. We say that $\mathcal{L}$ is a* block decomposition *of G, iff:*

- *every graph $B \in \mathcal{L} \setminus \{G\}$ is a block;*
- *for each graph $B \in \mathcal{L}$, either $\tilde{B}$ is 3-connected, or $\tilde{B}$ is isomorphic to K_3 – a clique graph on 3 vertices; and*
- *if a vertex $v(B) \in V(\tau)$ has exactly one child vertex $v(B')$, then $I(B) \neq I(B')$.*

For convenience, if $\mathcal{L}$ is a block decomposition of G, then we call the elements of $\mathcal{L}$ *pseudo-blocks*. Note that each pseudo-block is either a block of G, or it is G itself. The following theorem was proved in [12].

Theorem 1.4.1. *[Theorem 12 in the arxiv version of [12]] There is an efficient algorithm, that, given a 2-connected graph $G = (V, E)$ with $|V| \geq 3$, computes a block decomposition $\mathcal{L}$ of G, such that, for each vertex $v \in V$ that participates in some 2-separator (u, v) of G, either (i) v is an endpoint of a block $B \in \mathcal{L}$, or (ii) v has exactly two neighbors in G, and there is an edge $(u', v) \in E$, such that u' is an endpoint of a block $B \in \mathcal{L}$.*

We need the following two simple observations.

Observation 1.4.2. *Let G be a 2-connected graph, let $\mathcal{L}$ be a block decomposition of G, and let $B \in \mathcal{L}$ be a pseudo-block in the decomposition. Consider the corresponding graph $\tilde{B}$. Then, for every fake edge $e \in A_{\tilde{B}}$, there is a path $P(e)$ in G, connecting its endpoints, that is internally disjoint from $V(\tilde{B})$ and is completely contained in some graph $B_e \in \mathcal{N}(B)$. Moreover, if $e \neq e'$ are two distinct fake edges in $A_{\tilde{B}}$, then $B_e \neq B_{e'}$, and so paths $P(e)$ and $P(e')$ are internally disjoint.*

Observation 1.4.3. *Let G be a 2-connected planar graph, let $\mathcal{L}$ be a block decomposition of G, and let $B \in \mathcal{L}$ be a pseudo-block. Then $\tilde{B}$ is a planar graph, and it has a unique planar drawing.*

1.4.2 Block Decomposition of General Graphs

So far we have defined block decompositions for 2-connected graphs. We now extend this notion to general graphs, that may not even be connected, and introduce some useful notation. Let G be any graph. We denote by $\mathcal{C}(G)$ the set of all connected components of G. Consider now some connected component $C \in \mathcal{C}(G)$. Let $\mathcal{Z}(C)$ denote the collection of all maximal 2-connected sub-graphs of C (that is, $Z \subseteq C$ belongs to $\mathcal{Z}(C)$ iff Z is 2-connected, and it is not strictly contained in any other 2-connected subgraph of C). It is easy to verify that $\mathcal{Z}(C)$ is uniquely defined and can be computed efficiently. For convenience, we call the elements in $\mathcal{Z}(C)$ *super-blocks*. We also denote by $\mathcal{Z}(G) = \bigcup_{C \in \mathcal{C}(G)} \mathcal{Z}(C)$ the collection of all resulting super-blocks.

Finally, for every super-block $Z \in \mathcal{Z}(G)$, we let $\mathcal{L}(Z)$ be the block decomposition of Z given by Theorem 1.4.1. Recall that $\mathcal{L}(Z)$ contains the graph Z, and all other graphs in $\mathcal{L}(Z)$ are blocks of Z. We denote by $\mathcal{B}(C) = \bigcup_{Z \in \mathcal{Z}(C)} \mathcal{L}(Z)$ the collection of all pseudo-blocks in the block decompositions of the subgraphs $Z \in \mathcal{Z}(C)$, and we denote $\mathcal{B}(G) = \bigcup_{C \in \mathcal{C}(G)} \mathcal{B}(C)$. We will refer to the collection $\mathcal{B}(G)$ of pseudo-blocks a *block decomposition of* G. Observe that this generalizes the definition of block decompositions of 2-connected graphs to general graphs.

Consider now some super-block $Z \in \mathcal{Z}(G)$, and some pseudo-block $B \in \mathcal{L}(Z)$. If $B = Z$, then the complement block B^c is empty. Otherwise, the complement block B^c is defined exactly as before, with respect to the graph Z. In other words, B^c is the sub-graph of Z induced by the set $(V(Z) \setminus V(B)) \cup I(B)$ of vertices. We define the set $\mathcal{N}(B)$ of graphs as before: we add B^c to $\mathcal{N}(B)$, and additionally, for every child vertex $v(B')$ of $v(B)$, we add the block B' to $\mathcal{N}(B)$.

Embedding of Fake Edges

We will repeatedly use the following lemma, whose proof appears in Section 1.9.4.

Lemma 1.4.4. *Let G be a graph, and let $\mathcal{B}(G)$ be its block decomposition. Denote $\tilde{\mathcal{B}}(G) = \left\{\tilde{B} \mid B \in \mathcal{B}(G)\right\}$, and let $\tilde{\mathcal{B}}^*(G) \subseteq \tilde{\mathcal{B}}(G)$ contain all graphs $\tilde{B}$ that are not isomorphic to K_3. Then we can efficiently compute, for each graph $\tilde{B} \in \tilde{\mathcal{B}}^*(G)$, a collection $\mathcal{P}_{\tilde{B}} = \left\{P_{\tilde{B}}(e) \mid e \in A_{\tilde{B}}\right\}$ of paths in G, such that:*

- *for each fake edge $e = (u, v) \in A_{\tilde{B}}$, the path $P_{\tilde{B}}(e)$ connects u to v in G and it is internally disjoint from $\tilde{B}$;*
- *all paths in $\mathcal{P}_{\tilde{B}}$ are mutually internally disjoint; and*
- *if we denote $\mathcal{P} = \bigcup_{\tilde{B} \in \tilde{\mathcal{B}}^*(G)} \left(\mathcal{P}_{\tilde{B}} \setminus \left\{P_{\tilde{B}}(e^*_{\tilde{B}})\right\}\right)$, then every edge of G participates in at most 6 paths in $\mathcal{P}$.*

1.5 Computing a Decomposition into Acceptable Clusters: Proof of Theorem 1.3.1

This section is dedicated to the proof of Theorem 1.3.1. We start by introducing some notation and defining the notions of good pseudo-blocks.

1.5.1 Good Pseudo-Blocks

Throughout this subsection, we assume that we are given some planarizing set $\hat{E}$ of edges for the input graph G, that is, $G \setminus \hat{E}$ is planar. We note that $\hat{E}$ is not necessarily the same as the original planarizing set E', since, as the algorithm progresses, we may add edges to the planarizing set. The definitions in this subsection refer to any planarizing set $\hat{E}$ that may arise over the course of the algorithm.

Given a planarizing set $\hat{E}$ of edges for G, let $H = G \setminus \hat{E}$. Recall that we say that a vertex v of H is a *terminal* iff it is incident to some edge in $\hat{E}$, and we denote the set of terminals by Γ.

Recall that we have defined a block decomposition of a graph H as follows. We denoted by $\mathcal{C} = \mathcal{C}(H)$ the set of all connected components of H, and we refer to the elements of $\mathcal{C}$ as *clusters*. For each cluster $C \in \mathcal{C}$, we have defined a collection $\mathcal{Z}(C)$ of super-blocks (maximal 2-connected subgraphs) of C, and we denoted $\mathcal{Z}(H) = \bigcup_{C \in \mathcal{C}} \mathcal{Z}(C)$. Lastly, for each superblock $Z \in \mathcal{Z}(H)$, we let $\mathcal{L}(Z)$ be the block decomposition of Z given by Theorem 1.4.1, and we denoted by $\mathcal{B}(C) = \bigcup_{Z \in \mathcal{Z}(C)} \mathcal{L}(Z)$ the resulting collection of pseudo-blocks for cluster C. The final block decomposition of H is defined to be $\mathcal{B}(H) = \bigcup_{C \in \mathcal{C}} \mathcal{B}(C)$.

Consider some pseudo-block $B \in \mathcal{B}(H)$, and let $\tilde{B}$ be the corresponding graph that is either a 3-connected graph or isomorphic to K_3. Recall that $\tilde{B}$ has a unique planar drawing, that we denote by $\rho_{\tilde{B}}$. Throughout this section, for any pseudo-block B, we denote by $\tilde{B}' \subseteq \tilde{B}$ the graph obtained from $\tilde{B}$ by deleting all its fake edges. Note that $\tilde{B}'$ may be not 3-connected, and it may not even be connected. The drawing $\rho_{\tilde{B}}$ of $\tilde{B}$ naturally induces a drawing of $\tilde{B}'$, that we denote by $\rho'_{\tilde{B}'}$.

Definition 9. *We say that a pseudo-block $B \in \mathcal{B}(H)$ is a* good pseudo-block *iff there is a planar drawing $\hat{\psi}_B$ of B, that we call the* associated drawing*, such that, for each bridge $R \in \mathcal{R}_G(B)$, there is a face F in $\hat{\psi}_B$, whose boundary contains all vertices of $L(R)$. If B is not a good pseudo-block, then it is called a* bad pseudo-block.

Note that, if B is a bad pseudo-block, then for every planar drawing $\hat{\psi}_B$ of B, there is some bridge $R \in \mathcal{R}_G(B)$, such that no face of $\hat{\psi}_B$ contains all vertices of $L(R)$. We call such a bridge R a *witness* for B and $\hat{\psi}_B$. We note that for each bridge $R \in \mathcal{R}_G(B)$ and for every vertex $v \in L(R)$, either v is a terminal of Γ, or it is a separator vertex for the connected component C of H that contains B.

The remainder of the proof of Theorem 1.3.1 consists of two stages. In the first stage, we augment the initial planarizing set E' of edges to a new edge set E_1, such that for every connected component C of $G \setminus E_1$, either C is a type-1 acceptable cluster, or every pseudo-block in the block decomposition of C is good. In the second stage, we further augment E_1 in order to obtain the final edge set E'' and a decomposition of G into acceptable clusters. We now describe each of the two stages in turn.

1.5.2 Stage 1: Obtaining Type-1 Acceptable Clusters and Good Pseudo-Blocks

The main result of this stage is summarized in the following theorem.

Theorem 1.5.1. *There is an efficient algorithm, that, given a 3-connected n-vertex graph G with maximum vertex degree Δ and a planarizing set E' of edges for G, computes a planarizing edge set E_1 with $E' \subseteq E_1$, such that $|E_1| \leq O\left((|E'| + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \mathrm{poly}(\Delta \log n)\right)$, and, if we denote $H_1 = G \setminus E_1$, and let Γ_1 be the set of all endpoints of edges in E_1, then for* *every connected component C of H_1, either $|V(C) \cap \Gamma_1| \leq \mu$, or every pseudo-block in the block decomposition $\mathcal{B}(C)$ of C is a good pseudo-block. Moreover, for each connected component C of the latter type, for each pseudo-block B in $\mathcal{B}(C)$, the algorithm also computes its associated planar drawing $\hat{\psi}_B$.*

The remainder of this subsection is dedicated to proving Theorem 1.5.1. We start with a high-level intuition in order to motivate the next steps in this stage. Consider any pseudo-block $B \in \mathcal{B}(H)$ in the block decomposition of the graph H. We would like to construct a small set $E^*(B)$ of edges of $\tilde{B}'$, such that, for every bridge $R \in \mathcal{R}_G(\tilde{B}')$, all vertices of $L(R)$ lie on the boundary of a single face in the drawing of $\tilde{B}' \setminus E^*(B)$ induced by $\rho_{\tilde{B}}$. In general, we are able to find such an edge set $E^*(B)$, while ensuring that its size is small, compared with the following two quantities. The first quantity is the size of the vertex set $\Gamma'(B)$, that is defined to be the union of (i) all terminals (that is, endpoints of edges in E') lying in $\tilde{B}$; (ii) all endpoints of the fake edges of $\tilde{B}$; and (iii) all separator vertices of C that lie in $\tilde{B}$, where $C \in \mathcal{C}$ is the component of H that contains B. The second quantity is, informally, the number of crossings in which the edges of $\tilde{B}'$ participate in the optimal drawing φ^* of G (we need a slightly more involved definition of the second quantity that we provide later). We would like to augment E' with the edges of $\bigcup_{B \in \mathcal{B}(H)} E^*(B)$ to obtain the desired set E_1. Unfortunately, we cannot easily bound the size of $|E_1|$, as we cannot directly bound $\sum_{B \in \mathcal{B}(H)} |\Gamma'(B)|$. For example, consider a situation where the decomposition tree τ_Z associated with some maximal 2-connected subgraph Z of H contains a long induced path P. Then for every vertex $v(B)$ on path P, graph $\tilde{B}$ contains exactly two fake edges, one corresponding to its parent, and the other corresponding to its unique child. Note that, it is possible that many of the graphs $\tilde{B}$ with $v(B) \in P$ do not contain any terminals or separator vertices of the component of H that contains B, so $\sum_{v(B) \in P} |\Gamma'(B)|$ is very large, and it may be much larger than $\mathsf{OPT}_{\mathsf{cr}}(G) + |E'|$. To vercome this difficulty, we carefully decompose all such paths P, such that, after we delete a small number of edges from the graph, we obtain a collection of type-1 acceptable clusters, and for each block B with $v(B) \in P$, graph

$\tilde{B}$ is contained in one of these clusters. Our proof proceeds as follows. First, we bound the cardinality of the set U of the separator vertices of H by comparing it to the size of Γ. Then we mark some blocks B in the block decomposition $\mathcal{B}(H)$ of H. We will ensure that, on the one hand, we can suitably bound the total cardinality of the vertex sets $\Gamma'(B)$ for all marked blocks B, while on the other hand, in the forest associated with the block decomposition $\mathcal{B}(H)$ of H, if we delete all vertices corresponding to the marked blocks in $\mathcal{B}(H)$, we obtain a collection of paths, each of which can be partitioned into subpaths, which we use in order to define type-1 acceptable clusters. Let $\mathcal{B}'$ denote the set of all marked blocks. For each block $B \in \mathcal{B}'$, we define a collection $\chi(B)$ of crossings in the optimal drawing φ^* of G, such that $\sum_{B\in\mathcal{B}'} |\chi(B)|$ can be suitably bounded. We then process each such block $B \in \mathcal{B}'$ one by one, computing the edge set $E^*(B)$ of $\tilde{B}$, such that, for every bridge $R \in \mathcal{R}_G(\tilde{B}')$, all vertices of $L(R)$ lie on the boundary of a single face in the drawing of $\tilde{B}' \setminus E^*(B)$ induced by $\rho_{\tilde{B}}$. The cardinality of $E^*(B)$ will be suitably bounded by comparing it to $|\Gamma'(B)| + |\chi(B)|$.

We now proceed with the formal proof. Throughout the proof, we denote $H = G \setminus E'$ and denote by $\mathcal{C}$ the set of connected components of H that we call clusters. The set Γ of terminals contains all vertices that are endpoints of the edges of E'. For every cluster $C \in \mathcal{C}$, we denote by $U(C)$ the set of all separator vertices of C; that is, vertex $v \in U(C)$ iff graph $C \setminus \{v\}$ is not connected. Let $U = \bigcup_{C\in\mathcal{C}} U(C)$. We start by proving the following observation.

Observation 1.5.2. $|U| \leq O(|\Gamma|)$.

Proof. It suffices to show that, for every cluster $C \in \mathcal{C}$, $|U(C)| \leq O(|\Gamma \cap V(C)|)$. From now on we fix a cluster $C \in \mathcal{C}$. Let $\mathcal{Z}(C)$ be the decomposition of C into super-blocks. We can associate a graph T with this decomposition as follows. The set $V(T)$ of vertices is defined to be the union of (i) the set $U(C)$ of separator vertices, that we also refer to as regular vertices; and (ii) the set $U' = \{v_Z \mid Z \in \mathcal{Z}(C)\}$ of vertices called supernodes, representing the super-blocks of the decomposition. For every super-block $Z \in \mathcal{Z}(C)$ and every separator vertex $u \in U(C)$ such that $u \in V(Z)$, we add the edge (u, v_Z) to the graph. For every pair of distinct separator vertices $u, u' \in U(C)$, such that $(u, u') \in E(C)$ and there is no super-block $Z \in \mathcal{Z}(C)$ that contains both u and u', we add the edge (u, u') to the graph. It is easy to verify that graph T is a tree.

We partition the set $V(T)$ of vertices into the following three subsets: (i) the set V_1 contains all vertices that have degree 1 in T, namely V_1 is the set of all leaf vertices of T; (ii) the set V_2 contains all vertices that have degree 2 in T; and (iii) the set $V_{\geq 3}$ contains all vertices that have degree at least 3 in T. We further partition the set V_2 into three subsets: the set $U'_2 = U' \cap V_2$ containing all supernodes of V_2; the set $\hat{U}_2$ containing all regular vertices $u \in V_2$, such that both neighbors of u are regular vertices; and the set $\tilde{U}_2 = (U(C) \cap V_2) \setminus \hat{U}_2$ containing all remaining vertices. We use the following observation.

Observation 1.5.3. *$|V_1| \leq |\Gamma \cap V(C)|$, $|U'_2| \leq |\Gamma \cap V(C)|$, and $|\hat{U}_2| \leq |\Gamma \cap V(C)|$.*

Proof. Observe that $V_1 \subseteq U'$, and moreover, if some node $v_Z \in V_1$ corresponds to the super-block $Z \in \mathcal{Z}(C)$, then there must be at least one terminal vertex in $\Gamma \cap V(Z)$ that does not belong to $U(C)$. This follows from the fact that G is 3-connected, and $|V(Z) \cap U(C)| = 1$. Therefore, $|V_1| \leq |\Gamma \cap V(C)|$. Similarly, we can deduce that, for each vertex $v_Z \in U'_2$, its corresponding block Z must contain a terminal in $\Gamma \cap V(Z)$ that does not belong to $U(C)$. Therefore, $|U'_2| \leq |\Gamma \cap V(C)|$. From the definition of $\hat{U}_2$ and the fact that G is 3-connected, we get that every node in $\hat{U}_2$ has to be a terminal in $\Gamma \cap V(C)$. Therefore, $|\hat{U}_2| \leq |\Gamma \cap V(C)|$. $\square$

From the definition of the sets V_1 and $V_{\geq 3}$, $|V_{\geq 3}| \leq |V_1| \leq |\Gamma \cap V(C)|$. Moreover, if we denote by E^* the set of all edges of the tree T that are incident to a vertex of $V_{\geq 3}$, then $|E^*| \leq O(|V_1|)$.

Consider a vertex $u \in \tilde{U}_2$. Recall that u has exactly two neighbors in T, that we denote by x and y, and x and y are not both regular vertices. If $x \in V_{\geq 3}$ or $y \in V_{\geq 3}$, then u is an endpoint of an edge in E^*. If either of the vertices x or y lies in V_1, then u is the unique neighbor of that vertex in T. Assume now that neither of the two vertices lies in $V_1 \cup V_{\geq 3}$, that is, both vertices lie in V_2. Assume w.l.o.g. that $x \notin U$, so x is a supernode. Then u is one of the two neighbors of a supernode $x \in U_2'$. To summarize, if $u \in \tilde{U}_2$, then either (i) u is an endpoint of an edge of E^*; or (ii) u is a unique neighbor of a vertex of V_1; or (iii) one of the two neighbors of a vertex of U_2'. Therefore, $|\tilde{U}_2| \leq O(|E^*| + |V_1| + |U_2'|) \leq O(|V_1| + |U_2'|) \leq O(|\Gamma \cap V(C)|)$. Altogether:

$$|U(C)| = |U(C) \cap V_1| + |U(C) \cap V_2| + |U(C) \cap V_{\geq 3}| \leq 0 + |\hat{U}_2| + |\tilde{U}_2| + |V_{\geq 3}| \leq O(|\Gamma \cap V(C)|).$$

Summing over all clusters $C \in \mathcal{C}$, we get that $|U| \leq O(|\Gamma|)$. $\square$

Let $C \in \mathcal{C}$ be any cluster of H, and let $\mathcal{Z}(C)$ be the set of all super-blocks of C. For each super-block $Z \in \mathcal{Z}(C)$, we let $\mathcal{L}(Z)$ be the block decomposition of Z, given by Theorem 1.4.1. Recall that this block decomposition is associated with a decomposition tree, that we denote for brevity by τ_Z. As before, we denote by $\mathcal{B}(C) = \bigcup_{Z \in \mathcal{Z}(C)} \mathcal{L}(Z)$ and by $\mathcal{B}(H) = \bigcup_{C \in \mathcal{C}} \mathcal{B}(C)$ the block decompositions of C and H, respectively. Let $\mathcal{T}$ be the forest that consists of the trees τ_Z for all $Z \in \bigcup_{C \in \mathcal{C}} \mathcal{Z}(C)$. Recall that every vertex $v(B) \in \mathcal{T}$ corresponds to a pseudo-block $B \in \mathcal{B}(H)$ and vice versa.

Consider now some tree $\tau_Z \in \mathcal{T}$. We *mark* a vertex $v(B)$ of τ_Z iff, either (i) vertex $v(B)$ has at least two children in the tree τ_Z, or (ii) graph $\tilde{B}$ contains at least one vertex of $\Gamma \cup U$ that is not an endpoint of a fake edge of $\tilde{B}$. We denote by $\mathcal{B}' \subseteq \mathcal{B}(H)$ the set of all pseudo-blocks B whose corresponding vertex $v(B)$ was marked. For each pseudo-block $B \in \mathcal{B}'$, let $\Gamma'(B)$ be the set of vertices of $\tilde{B}$ that contains all terminals of Γ that lie in $\tilde{B}$, the vertices of U that lie in $\tilde{B}$, and all endpoints of the fake edges that belong to $\tilde{B}$. We need the following simple observation.

Observation 1.5.4. $\sum_{B \in \mathcal{B}'} |\Gamma'(B)| \leq O(\Delta \cdot |\Gamma|)$.

Proof. Consider a pseudo-block $B \in \mathcal{B}'$. We partition the set $\Gamma'(B)$ of vertices into three subsets: set $\Gamma'_1(B)$ contains all endpoints of fake edges of $\tilde{B}$; set $\Gamma'_2(B)$ contains all vertices of $\Gamma \setminus U$ that lie in $\tilde{B}$ and do not serve as endpoints of fake edges, and set $\Gamma'_3(B)$ contains all vertices of U that lie in $\tilde{B}$ and do not serve as endpoints of fake edges.

Notice that the sets $\left\{\Gamma'_2(B)\right\}_{B\in\mathcal{B}'}$ are mutually disjoint, since for any pair $B_1, B_2 \in \mathcal{B}(H)$ of pseudo-blocks in the decomposition, the only vertices of $\tilde{B}_1$ that may possibly lie in $\tilde{B}_2$ are vertices of U and vertices that serve as endpoints of fake edges in $\tilde{B}_1$. Therefore, $\sum_{B\in\mathcal{B}'} |\Gamma'_2(B)| \leq |\Gamma|$.

Notice that a vertex $u \in U$ may belong to at most Δ super-blocks in $\mathcal{Z}(H)$. For each super-block $Z \in \mathcal{Z}(H)$, there is at most one pseudo-block $B \in \mathcal{L}(Z)$, such that u belongs to $\tilde{B}$ but is not an endpoint of a fake edge of $\tilde{B}$. This is because for any pair B_1, B_2 of blocks of $\mathcal{L}(Z)$, the only vertices of $\tilde{B}_1$ that may possibly lie in $\tilde{B}_2$ are endpoints of fake edges of $\tilde{B}_1$. Therefore, $\sum_{B\in\mathcal{B}'} |\Gamma'_3(B)| \leq \Delta \cdot |U| \leq O(\Delta \cdot |\Gamma|)$, from Observation 1.5.2.

In order to bound $\sum_{B\in\mathcal{B}'} |\Gamma'_1(B)|$, we partition the set $\mathcal{B}'$ of pseudo-blocks into two subsets: set $\mathcal{B}'_1$ contains all pseudo-blocks B such that $v(B)$ has at least two children in the forest $\mathcal{T}$, and $\mathcal{B}'_2$ contains all remaining pseudo-blocks. Let $\mathcal{B}^* \subseteq \mathcal{B}(H)$ be the set of all pseudo-blocks whose corresponding vertex $v(B)$ has degree 1 in $\mathcal{T}$. Since the original graph G is 3-connected, and since for each pseudo-block $B \in \mathcal{B}^*$, $\tilde{B}$ contains at most one fake edge, for each pseudo-block $B \in \mathcal{B}^*$, the corresponding graph $\tilde{B}$ must contain a terminal $t \in \Gamma$ that is not one of its endpoints. If $t \notin U$, then B is the only pseudo-block in $\mathcal{B}^*$ such that $t \in \tilde{B}$ and t is not one of the endpoints of B. Otherwise, there are at most Δ such pseudo-blocks in $\mathcal{B}^*$. Therefore, $|\mathcal{B}^*| \leq |\Gamma| + \Delta \cdot |U| \leq O(\Delta \cdot |\Gamma|)$. From the definition of the set $\mathcal{B}'_1$, it is immediate to see that the total number of fake edges in all pseudo-blocks of $\mathcal{B}'_1$ is at most $O(|\mathcal{B}^*|) \leq O(\Delta \cdot |\Gamma|)$. Therefore, $\sum_{B\in\mathcal{B}'_1} |\Gamma'_1(B)| \leq O(\Delta \cdot |\Gamma|)$. Consider now a pseudo-block $B \in \mathcal{B}'_2$. Then $\tilde{B}$ contains at most two fake edges and at least one vertex of $\Gamma \cup U$, that is not an endpoint of a fake edge. Therefore, for each pseudo-block $B \in \mathcal{B}'_2$, $|\Gamma'_1(B)| \leq 4$, and $\sum_{B\in\mathcal{B}'_2} |\Gamma'_1(B)| \leq 4\sum_{B\in\mathcal{B}'_2}(|\Gamma'_2(B)| + |\Gamma'_3(B)|) \leq O(\Delta \cdot |\Gamma|)$. Altogether, we conclude that

$\sum_{B\in\mathcal{B}'} |\Gamma'(B)| \le O(\Delta \cdot |\Gamma|)$. $\square$

The final set E_1 of edges that is the outcome of Theorem 1.5.1 is the union of the input set E' of edges and four other edge sets, $\tilde{E}_1, \tilde{E}_2, \tilde{E}_3$ and $\tilde{E}_4$, that we define next.

Sets $\tilde{E}_1$ and $\tilde{E}_2$. We let set $\tilde{E}_1$ contain, for every pseudo-block $B \in \mathcal{B}'$ and for every fake edge $e = (x, y)$ of $\tilde{B}$, all edges of G that are incident to x or y. From Observation 1.5.4 and the definition of the set $\Gamma'(B)$, it is immediate that $|\tilde{E}_1| \le \sum_{B\in\mathcal{B}'} \Delta\cdot|\Gamma'(B)| \le O(\Delta^2\cdot|\Gamma|)$. We let set $\tilde{E}_2$ contain all edges incident to vertices of U. From Observation 1.5.2, $|\tilde{E}_2| \le O(\Delta \cdot |\Gamma|)$.

Set $\tilde{E}_3$. We now define the set $\tilde{E}_3$ of edges and identify a set $\mathcal{C}_1'$ of connected components of $H \setminus (\tilde{E}_1 \cup \tilde{E}_2 \cup \tilde{E}_3)$, each of which contains at most μ vertices that serve as endpoints of edges in $E' \cup \tilde{E}_1 \cup \tilde{E}_2 \cup \tilde{E}_3$.

Consider the graph obtained from the forest $\mathcal{T}$ by deleting all marked vertices in it. It is easy to verify that the resulting graph is a collection of disjoint paths, that we denote by $\mathcal{Q}$. Observe that the total number of paths in $\mathcal{Q}$ is bounded by the total number of fake edges in graphs of $\left\{\tilde{B} \mid B \in \mathcal{B}'\right\}$, so $|\mathcal{Q}| \le O(\Delta \cdot |\Gamma|)$ from Observation 1.5.4. Next, we will process the paths in $\mathcal{Q}$ one-by-one.

Consider now a path $Q \in \mathcal{Q}$. Notice that, from the definition of marked vertices, for every vertex $v(B) \in Q$, graph $\tilde{B}$ may contain at most two fake edges and at most four vertices of $\Gamma \cup U$, and all such vertices must be endpoints of the fake edges of $\tilde{B}$. For any sub-path $Q' \subseteq Q$, we define the graph $H(Q')$ as the union of all graphs $\tilde{B}'$, for all pseudo-blocks $B \in \mathcal{B}(H)$ with $v(B) \in V(Q')$ (recall that graph $\tilde{B}'$ is obtained from graph $\tilde{B}$ by removing all fake edges from it). The *weight* $w(Q')$ of the path Q' is defined to be the total number of vertices of $H(Q')$ that belong to $\Gamma \cup U$. We need the following simple observation.

Observation 1.5.5. *There is an efficient algorithm that computes, for every path $Q \in \mathcal{Q}$, a partition $\Sigma(Q)$ of Q into disjoint sub-paths $Q_1, \ldots, Q_z$, such that for all $Q_i \in \Sigma(Q)$, $w(Q_i) \le$*

$\mu/(2\Delta)$, and all but at most one path of $\Sigma(Q)$ have weight at least $\mu/(4\Delta)$. Moreover, every vertex of Q lies on exactly least one path of $\Sigma(Q)$.

Proof. We start with $\Sigma(Q) = \emptyset$ and then iterate as long as $w(Q) > \mu/(2\Delta)$. In an iteration, we let Q' be the shortest sub-path of Q that contains one endpoint of Q and has weight at least $\mu/(4\Delta)$. Since for every pseudo-block B with $v(B) \in Q$, $|(\Gamma \cup U) \cap V(\tilde{B})| \leq 4$ and $\mu > 16\Delta$, we get that $w(Q') \leq \mu/(2\Delta)$. We add Q' to $\Sigma(Q)$, delete all vertices of Q' from Q, and terminate the iteration. Once $w(Q) \leq \mu/(2\Delta)$ holds, we add the current path Q to $\Sigma(Q)$ and terminate the algorithm. □

Consider a sub-path $Q_i \in \Sigma(Q)$ and let $Q_i = (v(B_1), \ldots, v(B_x))$. We assume that $v(B_1)$ is an ancestor of $v(B_x)$ in the forest $\mathcal{T}$, and we denote by $v(B_{x+1})$ the unique child of $v(B_x)$ in the forest, if it exists. We denote by $\eta(Q_i)$ the set that consists of the endpoints of B_1 and the endpoints of B_{x+1} (if B_{x+1} exists). Note that the only vertices that $H(Q_i)$ may share with other graphs $H(Q_j)$ (for $i \neq j$) are the vertices of $\eta(Q_i)$. Moreover, for any path $Q' \neq Q$ in $\mathcal{Q}$ and any sub-path $Q'_j \in \Sigma(Q')$, the only vertices of $H(Q_i)$ that may possibly belong to $H(Q'_j)$ are the vertices of $\eta(Q_i)$ and $U \cap H(Q_i)$. On the other hand, note that a vertex $u \in U$ may belong to at most Δ super-blocks of $\mathcal{Z}(H)$. Therefore, for each $u \in U$, the number of graphs in $\{H(Q_i) \mid Q \in \mathcal{Q}, Q_i \in \Sigma(Q)\}$ such that $u \in V(H(Q_i)) \setminus \eta(Q_i)$ is at most Δ. Denote $\Sigma = \bigcup_{Q \in \mathcal{Q}} \Sigma(Q)$ and $\eta = \bigcup_{Q_i \in \Sigma} \eta(Q_i)$. We let $\tilde{E}_3$ contain all edges of G that are incident to the vertices of η. The following observation bounds the cardinality of $\tilde{E}_3$.

Observation 1.5.6. $|\tilde{E}_3| \leq O(\Delta^2 \cdot |\Gamma|)$.

Proof. Consider any path $Q \in \mathcal{Q}$ and the corresponding subset $\Sigma(Q)$ of its sub-paths. From Observation 1.5.5, there is at most one path $Q_i \in \Sigma(Q)$ with $w(Q_i) < \mu/(4\Delta)$. Denote $\Sigma'(Q) = \Sigma(Q) \setminus \{Q_i\}$ and $\Sigma' = \bigcup_{Q \in \mathcal{Q}} \Sigma'(Q)$. Observe that $|\Sigma \setminus \Sigma'| \leq |\mathcal{Q}| \leq O(\Delta \cdot |\Gamma|)$. We claim that $|\Sigma'| \leq O(\Delta \cdot |\Gamma|)$. Note that this implies Observation 1.5.6, since every path in Σ

contributes at most four vertices to set η, and the maximum vertex degree of a vertex in η is at most Δ.

It remains to show $|\Sigma'| \le O(\Delta \cdot |\Gamma|)$. Consider again some path $Q \in \mathcal{Q}$. Each sub-path $Q_i \in \Sigma'(Q)$ has weight $w(Q_i) \ge \mu/(4\Delta)$. Since $\mu > 16\Delta$ and $|\eta(Q_i)| \le 4$, there are at least $\mu/(8\Delta)$ vertices of $H(Q_i) \setminus \eta(Q_i)$ that belong to $\Gamma \cup U$. Let $S(Q_i)$ be the set of all these vertices. Note that every terminal vertex $t \in \Gamma$ may belong to at most one set $S(Q_i)$ for all paths $Q_i \in \Sigma'$, and every vertex $u \in U$ may belong to at most Δ such sets. Therefore, $|\Sigma'| \le \frac{|\Gamma|+\Delta|U|}{\mu/(8\Delta)} \le O(\Delta^2 \cdot |\Gamma|)$, since $\mu = \Theta(\Delta \cdot \log^{1.5} n)$. □

Consider now the graph $H' = H \setminus (\tilde{E}_1 \cup \tilde{E}_2 \cup \tilde{E}_3)$. From the definition of $\tilde{E}_1, \tilde{E}_2, \tilde{E}_3$, it is immediate that, for every path $Q_i \in \Sigma$, $\mathrm{out}_G(V(H(Q_i))) \subseteq E' \cup \tilde{E}_1 \cup \tilde{E}_2 \cup \tilde{E}_3$. Therefore, for every connected component C of H', either $V(C) \subseteq V(H(Q_i))$, or $V(C) \cap V(H(Q_i)) = \emptyset$. We let $\mathcal{C}_1'$ contain all connected components C of H', such that $V(C) \subseteq V(H(Q_i))$ for some path $Q_i \in \Sigma$.

Observation 1.5.7. *Each connected component $\mathcal{C}_1'$ contains at most μ vertices that are endpoints of edges in $E' \cup \tilde{E}_1 \cup \tilde{E}_2 \cup \tilde{E}_3$.*

Proof. Let $C \in \mathcal{C}_1'$ be any component, and let $Q_i \in \Sigma$ be the path such that $V(C) \subseteq V(H(Q_i))$. Let $S(Q_i)$ be the set of all vertices $v \in V(Q_i)$, such that $v \in U \cup \Gamma \cup \eta(Q_i)$. From the construction of the path Q_i and the definition of $w(Q_i)$, $|S(Q_i)| \le w(Q_i)+4 \le \mu/(2\Delta)+4$. Notice that a vertex v of $H(Q_i)$ may be an endpoint of an edge in $E' \cup \tilde{E}_1 \cup \tilde{E}_2 \cup \tilde{E}_3$ iff $v \in S(Q_i)$, or v has a neighbor that lies in $S(Q_i)$. Therefore, the total number of vertices of C that may be incident to edges of $E' \cup \tilde{E}_1 \cup \tilde{E}_2 \cup \tilde{E}_3$ is at most $(\Delta+1)|S(Q_i)| \le (\Delta+1) \cdot (\mu/(2\Delta)+4) \le \mu$. □

Set $\tilde{E}_4$. We now define the set $\tilde{E}_4$ of edges. Let $\mathcal{B}'' \subseteq \mathcal{B}'$ be the set of pseudo-blocks with $|V(\tilde{B})| > 3$. Recall that, for every pseudo-block $B \in \mathcal{B}''$, $\tilde{B}'$ is the graph obtained from $\tilde{B}$ by deleting all its fake edges. We will define, for each pseudo-block $B \in \mathcal{B}''$, a set $E^*(B)$ of edges of $\tilde{B}'$, that have the following useful property: for every bridge $R \in \mathcal{R}_G(\tilde{B}')$, all vertices of $L(R)$ lie on the boundary of a single face in the drawing of $\tilde{B}' \setminus E^*(B)$ induced by $\rho_{\tilde{B}}$. (Notice that this property already holds for all pseudo-blocks in $\mathcal{B}' \setminus \mathcal{B}''$, since for each pseudo-block $B \in \mathcal{B}' \setminus \mathcal{B}''$, graph $\tilde{B}$ is isomorphic to K_3). We will then set $\tilde{E}_4 = \bigcup_{B \in \mathcal{B}''} E^*(B)$.

In order to be able to bound $|\tilde{E}_4|$, we start by setting up an accounting scheme. We use Lemma 1.4.4 to compute, for each pseudo-block $B \in \mathcal{B}''$, an embedding $\mathcal{P}_{\tilde{B}} = \left\{ P_{\tilde{B}}(e) \mid e \in A_{\tilde{B}} \right\}$ of the set $A_{\tilde{B}}$ of fake edges of $\tilde{B}$ into paths that are internally disjoint from $\tilde{B}$ and are mutually internally disjoint. Recall that all paths in $\mathcal{P} = \bigcup_{B \in \mathcal{B}''} \left(\mathcal{P}_{\tilde{B}} \setminus \left\{ P_{\tilde{B}}(e^*_{\tilde{B}}) \right\} \right)$ cause edge-congestion at most 6 in G, where $e^*_{\tilde{B}}$ is the fake parent-edge for B, that connects the endpoints of B (it is possible that $e^*_{\tilde{B}}$ is undefined).

Consider now some pseudo-block $B \in \mathcal{B}''$. Recall that φ^* is some fixed optimal drawing of G. We define a set $\chi(B)$ of crossings to be the union of (i) all crossings (e, e') in φ^*, such that either e or e' (or both) are real edges of $\tilde{B}$; and (ii) all crossings (e, e') in φ^*, such that e, e' lie on two distinct paths of $\mathcal{P}_{\tilde{B}}$ (that is, $e \in \mathcal{P}_{\tilde{B}}(e_1)$, $e' \in \mathcal{P}_{\tilde{B}}(e_2)$ and $e_1 \neq e_2$ are two distinct fake edges of $\tilde{B}$). We need the following simple observation.

Observation 1.5.8. $\sum_{B \in \mathcal{B}''} |\chi(B)| \leq O(\mathsf{OPT}_{\mathsf{cr}}(G))$.

Proof. Consider any crossing (e, e') in the optimal drawing φ^* of G. Recall that e, e' may belong to $\chi(B)$ in one of two cases: either at least one of e, e' is a real edge of $\tilde{B}$; or e, e' lie on two distinct paths in $\mathcal{P}_{\tilde{B}}$. In particular, in the latter case, one of the two edges e, e' must lie on a path $P_{\tilde{B}}(\hat{e})$, where $\hat{e} \neq e^*_{\tilde{B}}$ (that is, $\hat{e}$ is not the fake parent-edge of B). Note that there may be at most one pseudo-block $B \in \mathcal{B}''$ for which e is a real edge, and the same is true for e'. Moreover, there are at most $O(1)$ pseudo-blocks $B \in \mathcal{B}''$, such that edge e lies on a path of $\mathcal{P}_{\tilde{B}} \setminus \left\{ P_{\tilde{B}}(e^*_{\tilde{B}}) \right\}$, and the same holds for e'. Therefore, there are at most $O(1)$ pseudo-blocks $B \in \mathcal{B}''$ with $(e, e') \in \chi(B)$. $\square$

In order to construct the sets $\{E^*(B)\}_{B \in \mathcal{B}''}$, we process the pseudo-blocks in $\mathcal{B}''$ one by one, using the following lemma.

Lemma 1.5.9. *There is an efficient algorithm, that, given a pseudo-block $B \in \mathcal{B}''$, computes a subset $E^*(B) \subseteq E(\tilde{B}')$ of edges, such that, if ρ' is the drawing of the graph $\tilde{B}' \setminus E^*(B)$ induced by the unique planar drawing $\rho_{\tilde{B}}$ of graph $\tilde{B}$, then for every bridge $R \in \mathcal{R}_G(\tilde{B}')$, all vertices of $L(R)$ lie on the boundary of a single face in ρ'. Moreover, $|E^*(B)| \leq O((|\chi(B)| + |\Gamma'(B)|) \cdot \text{poly}(\Delta \log n))$.*

Notice that Lemma 1.5.9 only considers bridges that are defined with respect to graph $\tilde{B}'$, namely bridges in $\mathcal{R}_G(\tilde{B}')$. Graph $\tilde{B}' \setminus E^*(B)$ may not even be connected, and its bridges in G can be completely different. However, this weaker property turns out to be sufficient for us. We defer the proof of Lemma 1.5.9 to Section 1.5.3. Now we complete the proof of Theorem 1.5.1 using it.

We let $\tilde{E}_4 = \bigcup_{B \in \mathcal{B}''} E^*(B)$. From Observations 1.5.4 and 1.5.8,

$$\begin{aligned} |\tilde{E}_4| = \sum_{B \in \mathcal{B}''} |E^*(B)| &\leq \sum_{B \in \mathcal{B}''} O\left((|\chi(B)| + |\Gamma'(B)|) \cdot \text{poly}(\Delta \log n)\right) \\ &\leq O((|\Gamma| + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \text{poly}(\Delta \log n)) \leq O((|E'| + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \text{poly}(\Delta \log n)). \end{aligned}$$

Lastly, we define the edge set E_1 to be the union of edge sets E', $\tilde{E}_1$, $\tilde{E}_2$, $\tilde{E}_3$ and $\tilde{E}_4$. From the above discussion, $|E_1| \leq O((|E'| + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \text{poly}(\Delta \log n))$.

Recall that, in graph $G \setminus (E' \cup \tilde{E}_1 \cup \tilde{E}_2 \cup \tilde{E}_3)$, all vertices in the set $\eta \cup U$ are isolated. From the definition of $\mathcal{C}_1'$, it is immediate that for any cluster $C \in \mathcal{C}_1'$, if a vertex of C is incident to an edge in $\tilde{E}_4$, then this vertex must lie in $\eta \cup U$, and therefore C contains a single vertex. Therefore, every cluster $C \in \mathcal{C}_1'$ remains a connected component of $G \setminus E_1$, and it contains at most μ vertices that are endpoints of edges in E_1.

Denote $H_1 = G \setminus E_1$. Consider now a connected component C of H_1 with $C \notin \mathcal{C}_1'$, and a block B in the block decomposition $\mathcal{B}(C)$ of C. Clearly, there is a block $B_0 \in \mathcal{B}'$ such that $\tilde{B}_0'$ contains B as a subgraph (this is because we have deleted all edges incident to vertices that serve as endoints of fake edges of every graph $\tilde{B}_a$, for $B_a \in \mathcal{B}'$). We then let $\hat{\psi}_B$ be the drawing of B induced by $\rho_{\tilde{B}_0}$, the unique planar drawing of $\tilde{B}_0$. It is now enough to prove the following claim.

Claim 1.5.10. *For every connected component C of H_1 with $C \notin \mathcal{C}'_1$, every pseudo-block B in the block decomposition $\mathcal{B}(C)$ of C is good, with $\hat{\psi}_B$ being its associated drawing.*

Proof. Assume for contradiction that the claim is false. Let $B \in \mathcal{B}(C)$ be a bad pseudo-block, and let $R \in \mathcal{R}_G(B)$ be a bridge that is a witness for B and drawing $\hat{\psi}_B$. Recall that there is a block $B_0 \in \mathcal{B}$ with $B \subseteq \tilde{B}'_0$. For brevity, we denote $\rho = \rho_{\tilde{B}_0}$, $\rho' = \rho'_{\tilde{B}'_0}$, and denote by $\psi = \hat{\psi}_B$ the drawing of B induced by ρ'.

Let $\mathcal{F}$ be the set of faces in the drawing ψ of B. From Lemma 1.5.9, for every bridge $R' \in \mathcal{R}_G(\tilde{B}'_0)$, if $L(R') \cap V(B) \neq \emptyset$, then there is some face $F \in \mathcal{F}$, such that all vertices of $L(R') \cap V(B)$ lie on the boundary of F in the drawing ψ. (If $B_0 \in \mathcal{B}' \setminus \mathcal{B}''$, so $\tilde{B}_0$ is isomorphic to K_3, this property must also hold). On the other hand, for every vertex $v \in V(\tilde{B}_0) \setminus V(B)$, there is a unique face $F(v) \in \mathcal{F}$, such that the image of vertex v in ρ lies in the interior of the face $F(v)$.

Consider now the witness bridge R for B. Recall that $T_R \subseteq R$ is a tree whose leaves are precisely the vertices of $L(R)$. Assume first that $V(T_R) \cap V(\tilde{B}_0) = L(R)$. In this case, there is some bridge $R' \in \mathcal{R}_G(\tilde{B}'_0)$ that contains the tree T_R, so $L(R) \subseteq L(R')$. However, from Lemma 1.5.9, all vertices of $L(R')$ lie on the boundary of the same face in the drawing ρ', and therefore they also lie on the boundary of the same face in the drawing ψ. This leads to a contradiction to R being a witness bridge for B and ψ.

Assume now that there is some vertex $v \in V(T_R) \cap V(\tilde{B}_0)$ that does not lie in $L(R)$. We will show that all vertices of $L(R)$ must lie on the boundary of $F(v)$, again leading to a contradiction. Let u be an arbitrary vertex of $L(R)$. Let $P \subseteq T_R$ be the unique path connecting v to u in T_R. Since the leaves of tree T_R are precisely the vertices of $L(R)$, except for v, every vertex x of P lies outside $V(B)$, and, if $x \in V(\tilde{B}_0)$, then the image of x

in ρ lies in the interior of some face in $\mathcal{F}$. Let $v = v_1, v_2, \ldots, v_r = u$ be all vertices of P that belong to $V(\tilde{B}_0)$, and assume that they appear on P in this order. It remains to prove the following observation.

Observation 1.5.11. *For all $1 \leq i < r-1$, $F(v_i) = F(v_{i+1})$. Moreover, vertex v_r lies on the boundary of face $F(v_{r-1})$.*

Proof. Fix some index $1 \leq i \leq r-1$. Assume for contradiction that the observation is false. Then there is some face $F' \in \mathcal{F}$, such that v_i lies in the interior of F', but v_{i+1} does not lie in the interior or on the boundary of F' (the latter case is only relevant for $i = r-1$). Since the boundary of F' separates v_i from v_{i+1}, they cannot lie on the boundary of the same face in the drawing ρ' of $\tilde{B}'_0$. Let σ_i be the subpath of P between v_i and v_{i+1}. If σ_i consists of a single edge (v_i, v_{i+1}), then it is either a bridge in $\mathcal{R}_G(\tilde{B}'_0)$, or it is an edge of $\tilde{B}'_0$. In the former case, Lemma 1.5.9 ensures that the endpoints of σ_i may not be separated by the boundary of a face of drawing ψ. In the latter case, the same holds since the drawing ψ is planar. In either case, we reach a contradiction. Otherwise, there exists a bridge $R' \in \mathcal{R}_G(\tilde{B}'_2)$ containing σ'_i, with $v_i, v_{i+1} \in L(R')$. However, from Lemma 1.5.9, it is impossible that the boundary of a face in the drawing ψ separates the two vertices, a contradiction. □

□

1.5.3 Proof of Lemma 1.5.9

In this section we provide the proof of Lemma 1.5.9. We fix a pseudo-block $B \in \mathcal{B}''$ throughout the proof. For brevity, we denote $\Gamma' = \Gamma'(B)$ and we denote by $\rho = \rho_{\tilde{B}}$ the unique planar drawing of $\tilde{B}$. Recall that vertex set Γ' contains all terminals of Γ (that is, endpoints of edges of E') lying in $\tilde{B}$, all endpoints of all fake edges of $\tilde{B}$, and all separator vertices in $U \cap V(\tilde{B})$. The set Γ' of vertices remains fixed throughout the algorithm. Abusing the notation, we call the vertices of Γ' *terminals* throughout the proof.

Throughout the algorithm, we maintain a subgraph J of $\tilde{B}$ and gradually construct the set $E^*(B)$. Initially, we set $E^*(B) = \emptyset$ and $J = \tilde{B}'$, the graph obtained from $\tilde{B}$ by deleting all its fake edges. Over the course of the algorithm, we will remove some edges from J and add them to the set $E^*(B)$. We will always use ρ' to denote the drawing of the current graph J induced by the drawing ρ of $\tilde{B}$.

Notice that a terminal of Γ' may belong to the boundaries of several faces in the drawing ρ', which is somewhat inconvenient for us. As our first step, we remove all edges that are incident to the terminals in Γ' from J, and add them to $E^*(B)$. Notice that now each terminal of Γ' becomes an isolated vertex and lies on the (inner) boundary of exactly one face of the current drawing ρ'. Clearly, $|E^*(B)| \leq \Delta \cdot |\Gamma'|$.

From now on, we denote by $\mathcal{F}$ the set of all faces in the drawing ρ' of the current graph J. For every terminal $t \in \Gamma'$, there is a unique face $F(t) \in \mathcal{F}$, such that t lies on the (inner) boundary of $F(t)$. For a face $F \in \mathcal{F}$, we denote by $\Gamma(F) \subseteq \Gamma'$ the set of all terminals t with $F(t) = F$.

Bad Faces. We denote by $\mathcal{R} = \mathcal{R}_G(\tilde{B}')$ the set of bridges for the graph $\tilde{B}'$ in G. For each bridge $R \in \mathcal{R}$, all vertices in $L(R)$ must be terminals. Let $\mathcal{F}(R) = \{F \in \mathcal{F} \mid L(R) \cap \Gamma(F) \neq \emptyset\}$ be the set of all faces in the drawing ρ' of the current graph J, whose inner boundaries contain terminals of $L(R)$. We say that a bridge $R \in \mathcal{R}$ is *bad* for J iff $|\mathcal{F}(R)| > 1$, namely, not all vertices of $L(R)$ lie on the boundary of the same face of $\mathcal{F}$. In such a case, we say that every face in $\mathcal{F}(R)$ is a *bad face*, and for each face $F \in \mathcal{F}(R)$, we say that bridge R is *responsible for F being bad*. As the algorithm progresses and the graph J changes, so does the set $\mathcal{F}$. The set $\mathcal{R}$ of bridges does not change over the course of the algorithm, and the definitions of the sets $\mathcal{F}(R)$ of faces for $R \in \mathcal{R}$ and of bad faces are always with respect to the current graph J and its drawing. The main subroutine that we use in our algorithm is summarized in the following lemma.

Lemma 1.5.12. *There is an efficient algorithm, that, given the current graph J and its drawing ρ', computes a subset $\hat{E}$ of at most $O\left((|\chi(B)| + |\Gamma'|) \cdot \mathrm{poly}(\Delta \log n)\right)$ edges, such that, if n_1 is the number of bad faces in the drawing ρ' of J, and n_2 is the number of bad faces in the drawing of $J \setminus \hat{E}$ induced by ρ', then $n_2 \leq n_1/2$.*

It is easy to complete the proof of Lemma 1.5.9 using Lemma 1.5.12. As long as the drawing ρ' of the current graph J contains bad faces (note that the number of bad faces is always either 0 or at least 2), we apply the algorithm from Lemma 1.5.12 to graph J to compute a set $\hat{E}$ of edges, then delete the edges of $\hat{E}$ from J and add them to $E^*(B)$, and continue to the next iteration. Once the drawing ρ' of the current graph J contains no bad faces, the algorithm terminates. It is easy to see that the number of iterations in the algorithm is $O(\log n)$. Therefore, at the end, $|E^*(B)| \leq O\left((|\chi(B)| + |\Gamma'|) \cdot \mathrm{poly}(\Delta \log n)\right)$. Consider now the graph J obtained at the end of the algorithm. Since the drawing ρ' of J contains no bad faces, for every bridge $R \in \mathcal{R}$, there is a single face of ρ' whose boundary contains all vertices of $L(R)$ (we emphasize that the graph J is not connected, and the vertices of $L(R)$ are isolated since they are terminals; but the drawing of each such vertex and the face to whose boundary it belongs are fixed by the original drawing of $\tilde{B}'$ induced by ρ). In order to complete the proof of Lemma 1.5.9, it suffices to prove Lemma 1.5.12.

From now on we focus on the proof of Lemma 1.5.12. Throughout the proof, we fix the drawing ρ' of the current graph J. Consider a pair F, F' of faces in $\mathcal{F}$. Let P be the shortest path connecting F to F' in the dual graph of J with respect to ρ'. This path defines a curve $\gamma(F, F')$, that starts at the interior of F, terminates at the interior of F', and intersects the image of J only at edges. Let $E(\gamma(F, F'))$ be the set of all edges whose image intersects $\gamma(F, F')$. Equivalently, $\gamma(F, F')$ can viewed as the curve that, among all curves γ connecting a point in the interior of F to a point in the interior of F' that only intersects the image of J at its edges, minimizes $|E(\gamma)|$. We define the distance between F and F' to be $\mathsf{dist}(F,F') = |E(\gamma(F,F'))|$. Equivalently, $\mathsf{dist}(F,F')$ is the minimum cardinality of a set $\tilde{E} \subseteq E(J)$ of edges, such that, in the drawing of $J \setminus \tilde{E}$ induced by ρ', the faces F and F' are merged into a single face.

Let $\mathcal{F}' \subseteq \mathcal{F}$ be the set of bad faces. For each $F \in \mathcal{F}'$, denote $\hat{F} = \arg\min_{F' \in \mathcal{F}'} \left\{\mathsf{dist}(F,F')\right\}$. We also denote $\Pi = \left\{(F,\hat{F}) \mid F \in \mathcal{F}'\right\}$. We define $\hat{E} = \bigcup_{F \in \mathcal{F}'} E(\gamma(F,\hat{F}))$. In other words, set $\hat{E}$ contains, for every pair $(F,\hat{F}) \in \Pi$, the set $E(\gamma(F,\hat{F}))$ of edges. Notice that, since a bad face may participate in several pairs in Π, it is possible that more than two faces may be merged into a single face. We remove the edges of $\hat{E}$ from J. Note that no new bad faces may be created, since the bad faces are only defined with respect to the original set $\mathcal{R}_G(\tilde{B}')$ of bridges. Therefore, the number of bad faces decreases by at least a factor of 2. It now remains to show that $|\hat{E}| \leq O\left((|\Gamma'| + |\chi(B)|) \cdot \mathrm{poly}(\Delta \log n)\right)$. This is done in the next claim, whose proof completes the proof of Lemma 1.5.12.

Claim 1.5.13. *$|\hat{E}| \leq O\left((|\Gamma'| + |\chi(B)|) \cdot \Delta \log n\right)$.*

Proof. For each $F \in \mathcal{F}'$, we denote $c(F) = \mathsf{dist}(F,\hat{F})$. To prove Claim 1.5.13, it is sufficient to show that $\sum_{F \in \mathcal{F}'} c(F) \leq O\left((|\Gamma'| + |\chi(B)|) \cdot \Delta \log n\right)$. We partition the set $\mathcal{F}'$ of bad faces into $O(\log n)$ classes $\mathcal{F}'_1, \ldots, \mathcal{F}'_z$, for $z \leq O(\log n)$ as follows. For each $1 \leq i \leq z$, face $F \in \mathcal{F}'$ lies in class $\mathcal{F}'_i$ iff $2^i \leq c(F) < 2^{i+1}$. Clearly, there must be an index i^*, such that $\sum_{F \in \mathcal{F}'_{i^*}} c(F) \geq \sum_{F' \in \mathcal{F}'} c(F')/O(\log n)$. We denote $\mathcal{F}'_{i^*} = \mathcal{F}^*$ and $c^* = 2^{i^*}$. Therefore, for every face $F \in \mathcal{F}^*$, $c^* \leq c(F) < 2c^*$. Since each bad face contains at least one terminal of Γ' on its inner boundary, $|F^*| \leq |\Gamma'|$. Therefore, if c^* is upper bounded by a constant, then $\sum_{F' \in \mathcal{F}'} c(F') \leq O(c^* \cdot |\mathcal{F}^*| \cdot \log n) \leq O(|\Gamma'| \log n)$. We will assume from now on that c^* is greater than some large constant. We use the following claim.

Claim 1.5.14. *$|\mathcal{F}^*| \leq O(|\chi(B)| \cdot \Delta/c^*)$.*

Note that Claim 1.5.14 completes the proof of Claim 1.5.13, since

$$\sum_{F' \in \mathcal{F}'} c(F') \leq O(\log n) \cdot \sum_{F \in \mathcal{F}^*} c(F) \leq O(|\mathcal{F}^*| \cdot c^* \cdot \log n) \leq O(|\chi(B)|\Delta \log n).$$

From now on we focus on the proof of Claim 1.5.14. The main idea of the proof is to associate, with each face $F \in \mathcal{F}^*$, a collection $\chi^F \subseteq \chi(B)$ of $\Omega(c^*/\Delta)$ crossings of $\chi(B)$, such that each crossing in $\chi(B)$ appears in at most $O(1)$ sets of $\{\chi_F\}_{F \in \mathcal{F}^*}$. Clearly, this implies that $|\mathcal{F}^*| \leq O(|\chi(B)|\Delta/c^*)$. In order to define the sets χ^F of crossings, we carefully construct a witness graph $W(F) \subseteq \tilde{B}$ for each face $F \in \mathcal{F}^*$, such that, for every pair $F, F' \in \mathcal{F}^*$ of distinct faces, graphs $W(F)$ and $W(F')$ are disjoint. We define the set χ^F of crossings for the face F by carefully considering the crossings in which the edges of $W(F)$ participate in the optimal drawing φ^* of G. The remainder of the proof consists of three steps. In the first step, we define a "shell" around each face F. In the second step, we use the shells in order to define the witness graphs $\{W(F)\}_{F \in \mathcal{F}^*}$. In the third and the last step, we use the witness graphs in order to define the collections $\chi^F \subseteq \chi(B)$ of crossings associated with each face $F \in \mathcal{F}^*$.

Step 1: Defining the Shells. We denote $z = \lfloor c^*/(16\Delta) \rfloor$. In this step we define, for every face $F \in \mathcal{F}^*$, a shell $\mathcal{H}(F) = \{L_1(F), \ldots, L_z(F)\}$, which is a collection of z disjoint subgraphs $L_1(F), \ldots, L_z(F)$ of J, that we refer to as *layers*.

We now fix a face $F \in \mathcal{F}^*$ and define its shell $\mathcal{H}(F)$ inductively, as follows. We consider the drawing ρ' of the graph J, and we view the face F as the outer face of the drawing. We let $L_1(F)$ be the boundary of the face F (note that this boundary may not be connected). Assume now that we have defined layers $L_1(F), \ldots, L_{i-1}(F)$. In order to define $L_i(F)$, we again consider the drawing ρ' of J, with F being its outer face, and we delete from this drawing the images of all vertices of $L_1(F), \ldots, L_{i-1}(F)$, and of all edges that are incident to these vertices. We then let $L_i(F)$ be the boundary of the outer face in the resulting plane graph. This completes the definition of the shell $\mathcal{H}(F) = \{L_1(F), \ldots, L_z(F)\}$. We denote $\Lambda(F) = \bigcup_{i=1}^{z} L_i(F)$.

In order to analyze the properties of the shells, we need the following notion of J-normal curves.

Definition 10. *Given a plane graph $\hat{J}$ and a curve γ, we say that γ is a $\hat{J}$-normal curve iff it intersects the image of $\hat{J}$ only at the images of its vertices.*

We state some simple properties of the shells in the next observation and its two corollaries.

Observation 1.5.15. *Let $\mathcal{H}(F) = \{L_1(F), \ldots, L_z(F)\}$ be a shell for some face $F \in \mathcal{F}^*$. Then for each $1 \leq i \leq z$, for every vertex $v \in L_i(F)$, there is a J-normal curve γ connecting v to a point in the interior of F, such that γ intersects exactly i vertices of J – one vertex from each graph $L_1(F), \ldots, L_i(F)$.*

Proof. It suffices to show that, for each $1 < j \leq z$ and for every vertex $v' \in L_j(F)$, there is a vertex $v'' \in L_{j-1}(F)$ and a curve γ_j connecting v' to v'', intersecting J only at its endpoints. The existence of the curve γ_j follows immediately from the definition of $L_j(F)$. Indeed, consider the drawing obtained from ρ' after we delete the images all vertices of $L_1(F), \ldots, L_{j-2}(F)$ together with all incident edges from it. Then there must be a face F' in the resulting drawing, that contains v' on its boundary, and also contains, on its boundary, another vertex $v'' \in L_{j-1}(F)$. This is because v' does not lie on the boundary of the outer face in the current drawing, but it lies on the boundary of the outer face in the drawing obtained from the current one by deleting all vertices of $L_{j-1}(F)$ and all incident edges from it. Therefore, there must be a curve γ_j connecting v' to v'', that is contained in F' and intersects J only at its endpoints. $\square$

Corollary 1.5.16. *Let $F \in \mathcal{F}^*$ and let $F' \neq F$ be any other bad face in $\mathcal{F}'$. Then graph $\Lambda(F)$ contains no vertex that lies on the boundary of F'.*

Proof. Assume for contradiction that the claim is false, and let $v \in \Lambda(F)$ be a vertex that lies on the boundary of F'. Then from Observation 1.5.15, there is a J-normal curve γ in the drawing ρ' of J, connecting v to a point in the interior of F, such that γ intersects at most $z \leq c^*/(16\Delta)$ vertices of J. Let $U \subseteq V(J)$ be the set of these vertices. By slightly adjusting γ we can ensure that it originates in the interior of F', terminates in the interior of F, does not intersect any vertices of J, and only intersects those edges of J that are incident to the vertices of U. But then $\mathsf{dist}(F, F') \leq |U| \cdot \Delta < c^* \leq c(F)$, a contradiction to the definition of $c(F)$. $\square$

Notice that, since $|\mathcal{F}'| \geq 2$, Corollary 1.5.16 implies, that for every face $F \in \mathcal{F}^*$, $L_z(F)$ is non-empty.

Corollary 1.5.17. *Let $F, F' \in \mathcal{F}^*$ be two distinct faces. Then $\Lambda(F) \cap \Lambda(F') = \emptyset$.*

Proof. Assume for contradiction that the claim is false, and let v be a vertex that lies in both $\Lambda(F)$ and $\Lambda(F')$. Then from Observation 1.5.15, there is a J-normal curve γ_1 in the drawing ρ' of J, connecting v to a point in the interior of F, such that γ_1 contains images of at most $z \leq c^*/(16\Delta)$ vertices of J. Similarly, there is a J-normal curve γ_2, connecting v to a point in the interior of F', such that γ_2 contains images of at most $c^*/(16\Delta)$ vertices of J. Let U be the set of all vertices of J whose images lie on either γ_1 or γ_2. By concatenating the two curves and slightly adjusting the resulting curve, we can obtain a curve that originates in the interior of F, terminates in the interior of F', does not intersect any vertices of J, and only intersects those edges of J that are incident to the vertices of U. But then $\mathsf{dist}(F, F') \leq |U| \cdot \Delta < c^* \leq c(F)$, a contradiction to the definition of $c(F)$. $\square$

Step 2: Computing the Witness Graphs. In this step we compute, for every face $F \in \mathcal{F}^*$, its witness graph $W(F)$. The witness graph $W(F)$ consists of two parts. The first part is a collection $\mathcal{Y}(F) = \{Y_1(F), \ldots, Y_{z-3}(F)\}$ of $z-3$ vertex-disjoint cycles. We will ensure that $Y_i(F) \subseteq L_i(F)$ for all $1 \leq i \leq z-3$. The second part is a collection

$\mathcal{Q}(F) = \{Q_1(F), Q_2(F), Q_3(F)\}$ of three vertex-disjoint paths in graph $\tilde{B}$, each of which has a non-empty intersection with each cycle in $\mathcal{Y}(F)$. Graph $W(F)$ is defined to be the union of the cycles in $\mathcal{Y}(F)$ and the paths in $\mathcal{Q}(F)$. The main challenge in this part is to define the path sets $\{\mathcal{Q}(F)\}_{F\in\mathcal{F}^*}$ so that the resulting witness graphs are disjoint.

We now fix a face $F \in \mathcal{F}^*$, and we start by defining the collection $\mathcal{Y}(F) = \{Y_1(F), \ldots, Y_{z-3}(F)\}$ of cycles for it. Let $R \in \mathcal{R}$ be the bad bridge that is responsible for F being a bad face. Then there must be a bad face $\overline{F} \in \mathcal{F}'$, such that $F \neq \overline{F}$, and at least one vertex of $L(R)$ lies on the (inner) boundary of $\overline{F}$. Let $P^*(F)$ be any path that is contained in R and connects a vertex of $L(R)$ on the boundary of F to a vertex of $L(R)$ on the boundary of $\overline{F}$. Note that path $P^*(F)$ is internally disjoint from $\tilde{B}'$.

From Corollary 1.5.16, no vertex of $\overline{F}$ may lie in $\Lambda(F)$. Therefore, for all $1 \leq i \leq z$, there is a simple cycle $Y_i(F) \subseteq L_i(F)$ that separates $\overline{F}$ from F. In other words, if we denote by $D(Y_i(F))$ the unique disc in the drawing ρ' with F as the outer face of the drawing, whose boundary is $Y_i(F)$, then $\overline{F} \subseteq D(Y_i(F))$, and F is disjoint from the interior of $D(Y_i(F))$. Clearly, the boundary of $\overline{F}$ is contained in $D(Y_z(F))$, and $D(Y_z(F)) \subsetneq D(Y_{z-1}(F)) \subsetneq \cdots \subsetneq D(Y_1(F))$, while the boundary of F is disjoint from the interior of $D(Y_1(F))$. We let $\mathcal{Y}(F) = \{Y_1(F), \ldots, Y_{z-3}(F)\}$ be the collection of disjoint cycles associated with F (we note that we exclude that last three cycles $Y_{z-2}(F), \ldots, Y_z(F)$ on purpose).

It remains to define a collection $\mathcal{Q}(F)$ of three disjoint paths in graph $\tilde{B}$, each of which connects a vertex of $Y_1(F)$ to a vertex of $Y_{z-3}(F)$. We emphasize that, while the cycles in $\mathcal{Y}(F)$ are all contained in the current graph $J \subseteq \tilde{B}'$ that only contains real edges of $\tilde{B}$ that have not been deleted yet, the paths in $\mathcal{Q}(F)$ are defined in graph $\tilde{B}$ and are allowed to contain fake edges. Since graph $\tilde{B}$ is 3-connected, it is not hard to see that such a collection $\mathcal{Q}(F)$ of paths must exist. However, we would like to ensure that all paths in the set $\bigcup_{F'\in\mathcal{F}^*} \mathcal{Q}(F')$ are mutually vertex-disjoint. In order to achieve this, we show in the next claim that there exist a desired set $\mathcal{Q}(F)$ of paths that only uses vertices in $\Lambda(F)$. Since all graphs in $\{\Lambda(F')\}_{F'\in\mathcal{F}^*}$ are mutually vertex-disjoint, the path sets $\{\mathcal{Q}(F')\}_{F'\in\mathcal{F}^*}$ are also mutually vertex-disjoint. The proof uses the fact that we have left the "padding" of three layers $L_{z-2}(F), L_{z-1}(F), L_z(F)$ between the cycles in $\mathcal{Y}(F)$ and $J \setminus \Lambda$.

Claim 1.5.18. *For every face $F \in \mathcal{F}^*$, there is a collection $\mathcal{Q}(F)$ of three vertex-disjoint paths in graph $\tilde{B}$, where each path connects a vertex of $Y_1(F)$ to a vertex of $Y_{z-3}(F)$, and only contains vertices of $\Lambda(F)$.*

Proof. Fix some face $F \in \mathcal{F}^*$. Since $\tilde{B}$ is 3-connected, there must be a collection $\mathcal{Q}$ of three vertex-disjoint paths in graph $\tilde{B}$, each of which connects a vertex of $Y_1(F)$ to a vertex of $Y_{z-3}(F)$. Among all such sets $\mathcal{Q}$ of paths we select the one that minimizes the number of vertices of $V(\tilde{B}) \setminus V(\Lambda(F))$ that belong to the paths in $\mathcal{Q}$. We now claim that no vertex of $V(\tilde{B}) \setminus V(\Lambda(F))$ may lie on a path in $\mathcal{Q}$.

Assume that this is false, and let $v \in V(\tilde{B}) \setminus V(\Lambda(F))$ be any vertex that lies on some path in $\mathcal{Q}$. Let graph K be the union of graph $\Lambda(F)$ and the paths in $\mathcal{Q}$. From the definition of the paths in $\mathcal{Q}$, graph $K \setminus \{v\}$ does not contain three vertex-disjoint paths that connect vertices of $Y_1(F)$ to vertices of $Y_{z-3}(F)$. In particular, there are two vertices $x, y \in V(K) \setminus \{v\}$, such that in graph $K \setminus \{v, x, y\}$, there is no path connecting a vertex of $Y_1(F)$ to a vertex of $Y_{z-3}(F)$. We will prove that this is false, reaching a contradiction. Notice that each of the vertices v, x, y must lie on a distinct path in $\mathcal{Q}$. We let $Q \in \mathcal{Q}$ be the path that contains v, so Q does not contain x or y. Notice that, from the definition of shells, for each $z-3 < j \leq z$, graph $L_j(F)$ must contain a simple cycle X_j that separates v from every cycle in $\mathcal{Y}(F)$ in the drawing ρ' of J. At least one of these three cycles $X \in \{X_{z-2}, X_{z-1}, X_z\}$ is disjoint from x and y. Notice that path Q must intersect the cycle X (this is since the drawing ρ' of J is the drawing induced by the unique planar drawing $\rho_{\tilde{B}}$ of $\tilde{B}$, and so X separates the cycles of $\mathcal{Y}(F)$ from v in $\rho_{\tilde{B}}$ as well). We view the path Q as originating at some vertex $a \in V(Y_1(F))$ and terminating at some vertex $b \in V(Y_{z-3}(F))$. Let v_1 be the first vertex of Q that lies on X, and let v_2 be the last vertex of Q that lies on X. Then we can use the segment of Q from a to v_1, the cycle X, and the segment of Q from v_2 to b to construct a path connecting a to b in graph K. Moreover, neither of these three graphs may contain a vertex of $\{x, y, v\}$, and so $K \setminus \{x, y, v\}$ contains a path connecting a vertex of $Y_1(F)$ to a vertex of $Y_{z-3}(F)$, a contradiction. $\square$

The witness graph $W(F)$ is defined to be the union of all cycles in $\mathcal{Y}(F)$ and the three paths in $\mathcal{Q}(F)$. Note that, from Corollary 1.5.17, all witness graphs in $\{W(F) \mid F \in \mathcal{F}^*\}$ are mutually vertex-disjoint. We emphasize that the cycles of $\mathcal{Y}(F)$ only contain real edges of $\tilde{B}$ (that belong to J), while the paths in $\mathcal{Q}(F)$ may contain fake edges of $\tilde{B}$.

Step 3: Defining the Sets of Crossings. The goal of this step is to define, for each face $F \in \mathcal{F}^*$, a set $\chi^F \subseteq \chi(B)$ of $\Omega(c^*/\Delta)$ crossings, such that each crossing in $\chi(B)$ appears in at most two sets of $\{\chi_F\}_{F \in \mathcal{F}^*}$. This will imply that $|\mathcal{F}^*| \leq O(|\chi(B)| \cdot \Delta/c^*)$, thus concluding the proof of Claim 1.5.14.

We now fix a face $F \in \mathcal{F}^*$ and define the set χ^F of crossings. We will first partition the graph $W(F)$ into $z' = \lfloor (z-3)/3 \rfloor$ disjoint subgraphs $W_1(F), \ldots, W_{z'}(F)$, each of which consists of three consecutive cycles in $\mathcal{Y}(F)$, and a set of three paths connecting them. Each such new graph will contribute exactly one crossing to χ^F. Recall that $z = \Theta(c^*/\Delta)$, so $z' = \Theta(c^*/\Delta)$.

We now fix an index $1 \leq i \leq z'$, and define the corresponding graph $W_i(F)$. We start with the set $\mathcal{Y}_i(F) = \{Y_{3i-2}(F), Y_{3i-1}(F), Y_{3i}(F)\}$ of three cycles. Additionally, we define a collection $\mathcal{Q}_i(F)$ of three disjoint paths, connecting vertices of $Y_{3i-2}(F)$ to vertices of $Y_{3i}(F)$, as follows. Consider any of the three paths $Q \in \mathcal{Q}(F)$. We view Q as originating at a vertex $a \in Y_1(F)$ and terminating at a vertex $b \in Y_{z-3}(F)$. From the definition of the cycles, path Q must intersect every cycle in $\mathcal{Y}(F)$. We let v be the last vertex of Q that lies on $Y_{3i-2}(F)$, and we let v' be the first vertex that appears on Q after v and lies on $Y_{3i}(F)$. We let Q^i be the segment of Q between v and v'. Notice that Q^i originates at a vertex of $Y_{3i-2}(F)$, terminates at a vertex of $Y_{3i}(F)$, and the inner vertices of Q_i are disjoint from all cycles in $\mathcal{Y}(F)$ except for Y_{3i-1} (that Q^i must intersect). Moreover, in the drawing ρ' of J where F is viewed as the outer face of the drawing, the interior of the image of Q^i is contained in $D(Y_{3i-2}(F)) \setminus D(Y_{3i}(F))$. We let $\mathcal{Q}_i(F) = \left\{Q^i \mid Q \in \mathcal{Q}(F)\right\}$ be the resulting set of three paths, containing one segment from each path in $\mathcal{Q}(F)$. Initially, we let the graph $W_i(F)$ be the union of the cycles in $\mathcal{Y}_i(F)$ and the paths in $\mathcal{Q}_i(F)$. Notice that for all $1 \leq i < i' \leq z'$, $W_i(F) \cap W_{i'}(F) = \emptyset$. For convenience, we rename the three cycles $Y_{3i-2}(F), Y_{3i-1}(F), Y_{3i}(F)$ in $\mathcal{Y}_i$ by $Y_i^1(F), Y_i^2(F)$ and $Y_i^3(F)$, respectively. Next, we slightly modify the graph $W_i(F)$, as follows. We let $\mathcal{Q}'_i(F)$ be a set of 3 vertex-disjoint paths in $W_i(F)$ that connect vertices of $Y_i^1(F)$ to vertices of $Y_i^3(F)$ and are internally disjoint from $V(Y_i^1(F)) \cup V(Y_i^3(F))$, and among all such paths, we choose those that contain fewest vertices of $V(W_i(F)) \setminus \left(\bigcup_{j=1}^3 V(Y_i^j(F))\right)$ in total, and, subject to this, contain fewest edges of $E(W_i(F)) \setminus \left(\bigcup_{j=1}^3 E(Y_i(F))\right)$. Clearly, set $\mathcal{Q}'_i(F)$ of paths is well defined, since we can use the set $\mathcal{Q}_i(F)$ of paths. We discard from $W_i(F)$ all vertices and edges except for those lying on the cycles in $\mathcal{Y}_i(F)$ and on the paths in $\mathcal{Q}_i(F)$. This finishes the definition of the graph $W_i(F)$.

To recap, graph $W_i(F)$ is the union of (i) three cycles $Y_i^1(F), Y_i^2(F)$ and $Y_i^3(F)$; each of the three cycles is contained in graph J and only contains real edges of graph $\tilde{B}$, and (ii) a set $\mathcal{Q}'_i(F)$ of three disjoint paths, each of which connects a distinct vertex of $Y_i^1(F)$ to a distinct vertex of $Y_i^3(F)$, and is internally disjoint from $V(Y_i^1(F)) \cup V(Y_i^3(F))$. Set $\mathcal{Q}'_i(F)$ of paths is chosen to minimize the number of vertices of $V(W_i(F)) \setminus \left(\bigcup_{j=1}^3 V(Y_i^j(F))\right)$ that lie on the paths. The paths in $\mathcal{Q}'_i(F)$ are contained in graph $\tilde{B}$ and may contain fake edges. All resulting graphs $W_i(F)$ for all $F \in \mathcal{F}^*$ and $1 \leq i \leq z'$ are disjoint from each other. Note that each such graph $W_i(F) \subseteq \tilde{B}$ is a planar graph. We need the following claim.

Claim 1.5.19. *For each $F \in \mathcal{F}^*$ and $1 \leq i \leq z'$, if ψ is any planar drawing of $W_i(F)$ on the sphere, and D, D' are the two discs whose boundary is the image of $Y_i^2(F)$, then the images of $Y_i^1(F), Y_i^3(F)$ cannot lie in the same disc in $\left\{D, D'\right\}$ (in other words, the image* *of $Y_i^2(F)$ separates the images of $Y_i^1(F)$ and $Y_i^3(F)$).*

Proof. Let W be the graph obtained from $W_i(F)$ after all degree-2 vertices are suppressed. We denote by Y^1, Y^2 and Y^3 the cycles corresponding to $Y_i^1(F), Y_i^2(F)$ and $Y_i^3(F)$ in W respectively, and we denote by Q_1, Q_3 and Q_3 the paths corresponding to the paths in $\mathcal{Q}'_i(F)$ in W. Notice that every vertex of W must lie on one of the cycles Y^1, Y^2, Y^3, and on one of the paths Q_1, Q_2, Q_3. Moreover, graph W may not have parallel edges (due to the minimality of the set $\mathcal{Q}'_i(F)$ of paths).

Observe that, from the definition of the cycles in $\mathcal{Y}(F)$, in the unique planar drawing $\rho_{\tilde{B}}$ of $\tilde{B}$ on the sphere, the image of Y^2 separates the images of Y^1 and Y^3. Therefore, there exists a planar drawing of W on the sphere, such that, if D, D' are the two discs whose boundary is the image of Y^2, then the images of Y^1 and Y^3 do not lie in the same disc in $\left\{D, D'\right\}$. Therefore, it suffices to show that W is a 3-connected graph.

Assume for contradiction that this is not the case, and let $\{x, y\}$ be a pair of vertices of W, such that there is a partition (X, X') of $V(W) \setminus \{x, y\}$, with $X, X' \neq \emptyset$, and no edge of W connects a vertex of X to a vertex of X'. For brevity, we denote $\mathcal{Y} = \left\{Y^1, Y^2, Y^3\right\}$, $\mathcal{Q} = \{Q_1, Q_2, Q_3\}$, and we will sometimes say that a cycle $Y \in \mathcal{Y}$ is contained in X (or in X') if $V(Y) \subseteq X$ (or $V(Y) \subseteq X'$, respectively). We will use a similar convention for paths in $\mathcal{Q}$.

We first claim that both x, y must belong to the same cycle of $\mathcal{Y}$. Indeed, assume for contradiction that they belong to different cycles. Then there must be a path $Q \in \mathcal{Q}$ that is disjoint from x, y, with all vertices of Q lying in one of the two sets X, X' (say X). But, since Q intersects every cycle in $\mathcal{Y}$, for each cycle $Y \in \mathcal{Y}$, all vertices of $V(Y) \setminus \{x, y\}$ lie in X (as $Y \setminus \{x, y\}$ is either a cycle or a connected path). Therefore, $X' = \emptyset$, an contradiction.

We denote by Y the cycle in $\mathcal{Y}$ that contain vertices x, y. Note that each of the remaining cycles must be contained in X or contained in X'. We now consider two cases.

The first case is when there is some path $Q \in \mathcal{Q}$ that contains both vertices x and y; assume w.l.o.g. that it is Q_1. Since path Q_2 is disjoint from x and from y, it must be contained in one of the two sets X, X'; assume w.l.o.g. that it is X. Since Q_2 intersects every cycle in $\left\{Y^1, Y^2, Y^3\right\}$, and two of these cycles are disjoint from x, y, we get that both cycles in $\left\{Y^1, Y^2, Y^3\right\} \setminus \{Y\}$ lie in X. Since path Q_3 is disjoint from x, y but intersects each cycle in $\left\{Y^1, Y^2, Y^3\right\}$, it must be contained in X as well. Therefore, every vertex of $X' \cup \{x, y\}$ must belong to $Y \cap Q_1$. But then all vertices of X' must have degree 2, a contradiction.

It remains to consider the case where x and y lie on two different paths of $\{Q_1, Q_2, Q_3\}$, say Q_1 and Q_2. Then path Q_3 is disjoint from x, y, and is contained in one of the sets X, X'; assume without loss of generality that it is X. Notice that path Q_3 contains vertices from all three cycles in $\mathcal{Y}$, therefore, each of the two cycles in $\mathcal{Y} \setminus \{Y\}$ is also contained in X. Set X' then contains vertices of a single cycle in $\mathcal{Y}$ – the cycle Y (that contains the vertices x and y). Since the paths Q_1 and Q_2 connect vertices of Y^1 to vertices of Y^3, and each of them has one endpoint in X and another in X', $Y \neq Y^2$ must hold. We assume without loss of generality that $Y = Y^1$ (the case where $Y = Y^3$ is symmetric). Therefore, every vertex of $X' \cup \{x, y\}$ lies on cycle Y^1, and on either path Q_1 or path Q_2. However, from our construction of set $\mathcal{Q}'_i(F)$, each path of $\mathcal{Q}$ contains exactly one vertex of Y_1 (which serves as its endpoint). Since each of the paths Q_1, Q_2 contains one vertex of $\{x, y\}$ that lies on Y_1, it follows that $X' = \emptyset$, a contradiction.

We conclude that graph W is 3-connected and therefore has a unique planar drawing – the drawing induced by the drawing $\rho_{\tilde{B}}$ of $\tilde{B}$. In that drawing (on the sphere), the image of cycle Y^2 separates the images of cycles Y^1 and Y^3. Therefore, in every planar drawing of W on the sphere, the image of Y^2 separates the images of Y^1 and Y^3. Since graph $W_i(F)$ is obtained from W by subdividing some of its edges, in every planar drawing of $W_i(F)$ on the sphere, the image of $Y_i^2(F)$ separates the images of $Y_i^1(F)$ and $Y_i^3(F)$. $\square$

Lastly, we use the following claim to associate a crossing of $\chi(B)$ with the graph $W_i(F)$.

Claim 1.5.20. *Consider some face $F \in \mathcal{F}^*$ and index $1 \leq i \leq z'$. Let φ^* be the fixed optimal drawing of the graph G. Then there is a crossing (e, e') in this drawing, such that:*

- *either at least one of the edges e, e' is a real edge of $\tilde{B}$ that lies in $W_i(F)$; or*
- *there are two distinct fake edges $e_1, e_2 \in E(\tilde{B})$, that belong to $W_i(F)$, such that $e \in P(e_1)$ and $e' \in P(e_2)$, where $P(e_1), P(e_2) \in \mathcal{P}_{\tilde{B}}$ are the embeddings of the fake edges e_1, e_2, respectively.*

Assuming that the claim is correct, the crossing (e, e') must lie in $\chi(B)$. We denote by χ_i^F the crossing (e, e') obtained by applying Claim 1.5.20 to graph $W_i(F)$. We then set $\chi^F = \left\{\chi_i^F \mid 1 \leq i \leq z'\right\}$. It is easy to verify that $\chi^F \subseteq \chi(B)$, and that it contains $z' = \Omega(c^*/\Delta)$ distinct crossings. Moreover, since all witness graphs in $\{W(F) \mid F \in \mathcal{F}^*\}$ are disjoint from each other, each crossing in $\chi(B)$ appears in at most two sets of $\{\chi_F\}_{F \in \mathcal{F}^*}$. It remains to prove Claim 1.5.20.

Proof of Claim 1.5.20. We fix a face $F \in \mathcal{F}^*$ and an index $1 \leq i \leq z'$. Consider the fixed optimal drawing φ^* of graph G on the sphere. If this drawing contains a crossing (e, e'), where at least one of the edges e, e' is a real edge of $\tilde{B}$ that lies in $W_i(F)$, then we are done. Therefore, we assume from now on that this is not the case. In particular, we can assume that the edges of the cycles $\mathcal{Y}_i(F)$ do not participate in any crossings in φ^* (recall that all edges of these cycles are real edges of $\tilde{B}$.)

Therefore, the image of cycle $Y_i^2(F)$ in φ^* is a simple closed curve. Let D, D' be the two discs whose boundaries are $Y_i^2(F)$. We claim that the images of both remaining cycles, $Y_i^1(F), Y_i^3(F)$ must lie inside a single disc in $\left\{D, D'\right\}$.

Recall that we have defined a path $P^*(F)$, that is contained in some bridge $R \in \mathcal{R}$, and connects some vertex v on the boundary of F to some vertex v' on the boundary of $\overline{F}$. Since both v, v' lie in $\tilde{B}$, and since graph $\tilde{B}$ is connected, there is a path P in $\tilde{B}$ that connects v to v'; we view path P as originating from v and terminating at v'. Since each cycle $Y \in \mathcal{Y}_i(F)$

separates the boundary of F from the boundary of $\overline{F}$ in the drawing ρ' of J, and since $J \subseteq \tilde{B}$ and ρ' is the drawing of J induced by the planar drawing $\rho_{\tilde{B}}$ of $\tilde{B}$, path P must intersect every cycle in $\mathcal{Y}_i(F)$. We let x be the first vertex of P that lies on cycle $Y_i^1(F)$, and y the last vertex on P that lies on cycle $Y_i^3(F)$. Let P' be the sub-path of P connecting v to x, and let P'' be the sub-path of P connecting y to v'. Notice that both paths are internally disjoint from the cycles in $\mathcal{Y}_i(F)$. Next, we denote by $\hat{P}$ the path that is obtained by concatenating the paths P', P^* and P''. Therefore, path $\hat{P}$ connects a vertex $x \in Y_i^1(F)$ to a vertex $y \in Y_i^3(F)$, and it is internally disjoint from the cycles in $\mathcal{Y}_i(F)$. Note however that path $\hat{P}$ may contain fake edges of $\tilde{B}$. For every fake edge $e \in A_{\tilde{B}}$ that lies on $\hat{P}$, we replace e with its embedding $P(e) \in \mathcal{P}_{\tilde{B}}$ given by using Lemma 1.4.4. Recall that the lemma guarantees that the path $P(e)$ is internally disjoint from the vertices of $\tilde{B}$, and that all paths in $\mathcal{P}_{\tilde{B}} = \left\{P_{\tilde{B}}(e') \mid e' \in A_{\tilde{B}}\right\}$ are mutually internally disjoint. Let $\hat{P}'$ be the path obtained from $\hat{P}$ after we replace every fake edge on $\hat{P}$ with its corresponding embedding path. Notice that $\hat{P}'$ still connects x to y and it is still internally disjoint from all cycles in $\mathcal{Y}_i(F)$ (that are also present in G, as they only contain real edges). If the images of the cycles $Y_i^1(F), Y_i^3(F)$ are contained in distinct discs in $\{D, D'\}$, then the endpoints of the path $\hat{P}'$ lie on opposite sides of the image of $Y_i^2(F)$. Since path $\hat{P}'$ is disjoint from cycle $Y_i^2(F)$, at least one edge of $Y_i^2(F)$ must participate in a crossing in φ^*, a contradiction. Therefore, we can assume from now on that the images of both cycles $Y_i^1(F), Y_i^3(F)$ lie inside a single disc in $\{D, D'\}$ (say D).

Next, we use the drawing φ^* of G on the sphere in order to define a corresponding drawing φ of graph $W_i(F)$, as follows. Recall that every vertex and every real edge of $W_i(F)$ belong to G, so their images remain unchanged. Consider now some fake edge $e \in E(W_i(F))$. Let $P(e)$ be the path in G into which this edge was embedded, and let $\gamma(e)$ be the image of this path in φ^* (obtained by concatenating the images of its edges). If curve $\gamma(e)$ crosses itself then we delete loops from it, until it becomes a simple open curve, and we draw the edge e along the resulting curve. Recall that all paths that are used to embed the fake edges of $\tilde{B}$

are internally disjoint from $V(\tilde{B})$ and internally disjoint from each other.

Consider now the resulting drawing φ of $W_i(F)$. As before, the edges of the cycles in $\mathcal{Y}_i(F)$ do not participate in crossings in φ, and, if we define the discs D, D' as before (the discs whose boundary is $Y_i^2(F)$), then the images of $Y_i^1(F), Y_i^2(F)$ lie in the same disc D. From Claim 1.5.19, the drawing φ of $W_i(F)$ is not planar. Let (e_1, e_2) be any crossing in this drawing. It is impossible that e_1 or e_2 are real edges of $W_i(F)$, since we have assumed that no real edges of $W_i(F)$ participate in crossings in φ^*. Therefore, e_1, e_2 must be two distinct fake edges of $W_i(F)$, such that there are edges $e \in P(e_1), e' \in P(e_2)$ whose images in φ^* cross. $\square$

$\square$

1.5.4 Stage 2: Obtaining a Decomposition into Acceptable Clusters

In this subsection we complete the proof of Theorem 1.3.1. We start with a 3-connected graph G with maximum vertex degree Δ, and a planarizing set E' of edges of G. We then use Theorem 1.5.1 to compute a subset E_1 of edges of G, with $E' \subseteq E_1$, such that $|E_1| \leq O\left((|E'| + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \operatorname{poly}(\Delta \log n)\right)$, and a set $\mathcal{C}_1'$ of connected components of $G \setminus E_1$, each of which contains at most μ vertices that are incident to edges of E_1.

The remainder of the algorithm is iterative. We use a parameter $\alpha' = 8\Delta\alpha = \frac{1}{16\alpha_{\mathrm{ARV}}(n)\log_{3/2} n}$. Recall that $\alpha = \frac{1}{128\Delta\alpha_{\mathrm{ARV}}(n)\log_{3/2} n}$ is the well-linkedness parameter from the definition of type-2 acceptable clusters. Throughout the algorithm, we maintain a set $\hat{E}$ of edges of G, starting with $\hat{E} = E_1$, and then gradually adding edges to $\hat{E}$. We also maintain a set A of fake edges, initializing $A = \emptyset$. We denote $\hat{H} = G \setminus \hat{E}$ with respect to the current set $\hat{E}$, and we let $\hat{\mathcal{C}}$ be the set of connected components of $\hat{H} \cup A$, that we refer to as *clusters*. We call the endpoints of the edges of $\hat{E}$ *terminals*, and denote by $\hat{\Gamma}$ the set of terminals. We will ensure that edges of A only connect pairs of terminals. We will also maintain an embedding $\mathcal{P} = \{P(e) \mid e \in A\}$ of the fake edges, where for each edge $e = (u, v) \in A$, path $P(e)$ is contained in graph G, and it connects u to v. We will ensure that all paths in $\mathcal{P}$ are mutually internally disjoint. We also maintain a partition of $\hat{\mathcal{C}}$ into two subsets: set $\mathcal{C}^A$ of *active* clusters, and set $\mathcal{C}^I$ of *inactive* clusters. Set $\mathcal{C}^I$ of inactive clusters is in turn partitioned into two subsets, $\mathcal{C}^I_1$ and $\mathcal{C}^I_2$. We will ensure that every cluster $C \in \mathcal{C}^I_1$ is a type-1 acceptable cluster. In particular, no edges of A are incident to vertices of clusters in $\mathcal{C}^I_1$. For every cluster $C \in \mathcal{C}^I_2$, we denote by $A_C \subseteq A \cap E(C)$ the set of fake edges contained in C. We will maintain, together with cluster C, a planar drawing ψ_C of C on the sphere, such that C is a type-2 acceptable cluster with respect to ψ_C. Additionally, for every fake edge $e = (x, y) \in A_C$, its embedding $P(e)$ is internally disjoint from C. Moreover, we will ensure that there is some cluster $C(e) \in \mathcal{C}^I_1$ containing $P(e) \setminus \{x, y\}$, and for every pair $e, e' \in A$ of distinct edges, $C(e) \neq C(e')$. Lastly, we ensure that no fake edges are contained in an active cluster of $\mathcal{C}^A$.

Vertex Budgets. For the sake of accounting, we assign a budget $b(v)$ to every vertex v in G. The budgets are defined as follows. If $v \notin \hat{G}$, then $b(v) = 0$. Assume now that $v \in \hat{G}$, and let $C \in \hat{\mathcal{C}}$ be the unique cluster containing v. If $C \in \mathcal{C}^I$, then $b(v) = 1$. Otherwise, $b(v) = 8\Delta \cdot \log_{3/2}(|\hat{\Gamma} \cap V(C)|)$. At the beginning of the algorithm, the total budget of all vertices is $\sum_{t \in \hat{\Gamma}} b(t) \leq O(|\hat{\Gamma}| \cdot \Delta \log n) \leq O(|E_1| \cdot \Delta \log n)$. Note that, as the algorithm progresses, the sets $\mathcal{C}^I$ and $\mathcal{C}^A$ evolve, and the budgets may change. We will ensure that, over the course of the algorithm, the total budget of all vertices does not increase. Since the budget of every terminal in $\hat{G}$ is always at least 1, the total budget of all vertices is at least $|\hat{\Gamma}|$ throughout the algorithm, and this will ensure that the total number of terminals at the end of the algorithm is bounded by $O(|E_1| \cdot \Delta \log n) \leq O\left((|E'| + \mathsf{OPT}_{\mathsf{cr}}(G)) \cdot \mathrm{poly}(\Delta \log n)\right)$, and therefore $|\hat{E}|$ is also bounded by the same amount.

Initialization. At the beginning, we let $\hat{E} = E_1$, and we let $\hat{\mathcal{C}}$ be the set of all connected components of the graph $\hat{H} \cup A$, where $\hat{H} = G \setminus \hat{E}$, and $A = \emptyset$. The set $\hat{G}$ of terminals contains all endpoints of edges in $\hat{E}$. Recall that we have identified a subset $\mathcal{C}'_1$ of clusters of $\hat{H}$, each of which contains at most μ terminals. We set $\mathcal{C}^I = \mathcal{C}^I_1 = \mathcal{C}'_1$, $\mathcal{C}^I_2 = \emptyset$, and $\mathcal{C}^A = \hat{\mathcal{C}} \setminus \mathcal{C}^I$. The algorithm proceeds in iterations, as long as $\mathcal{C}^A \neq \emptyset$.

Description of an Iteration

We now describe a single iteration. Let $C \in \mathcal{C}^A$ be any active cluster. If $|\hat{\Gamma} \cap V(C)| \leq \mu$, then we move C from $\mathcal{C}^A$ to $\mathcal{C}^I_1$ (and to $\mathcal{C}^I$), and continue to the next iteration. Clearly, in this case C is a type-1 acceptable cluster, and the budgets of vertices may only decrease.

We assume from now on that $|\hat{\Gamma} \cap V(C)| > \mu$ and denote $\tilde{\Gamma} = \hat{\Gamma} \cap V(C)$. We then apply the algorithm $\mathcal{A}_{\text{ARV}}$ for computing the (approximate) sparsest cut in the graph C, with respect to the set $\tilde{\Gamma}$ of terminals. Let (X, Y) denote the cut that the algorithm returns, and assume w.l.o.g. that $|X \cap \tilde{\Gamma}| \leq |Y \cap \tilde{\Gamma}|$. Denote $E^* = E_C(X, Y)$. Assume first that $|E^*| < \alpha' \cdot \alpha_{\text{ARV}}(n)|X \cap \tilde{\Gamma}|$. Then we delete the edges of E^* from $\hat{H}$ and add them to the set $\hat{E}$. We then replace the cluster C in $\mathcal{C}^A$ by all connected components of $C \setminus E^*$ and continue to the next iteration. Note that we may have added new terminals to $\hat{\Gamma}$ in this iteration: the endpoints of the edges in E^*. Denote by Γ^* the set of endpoints of these edges. The changes in the budgets of vertices are as follows. On the one hand, for every new terminal $t \in \Gamma^*$, the budget $b(t)$ may have grown from 0 to at most $8\Delta \cdot \log_{3/2} n$. Since $|\Gamma^*| \leq 2|E^*| \leq 2\alpha' \cdot \alpha_{\text{ARV}}(n)|X \cap \tilde{\Gamma}|$, the total increase in the budget of new terminals is at most $16\alpha' \cdot \alpha_{\text{ARV}}(n)|X \cap \tilde{\Gamma}| \cdot \Delta \log_{3/2} n$. Note that for terminals in $\tilde{\Gamma} \cap Y$, their budgets can only decrease. On the other hand, since $|\tilde{\Gamma} \cap X| \leq |\tilde{\Gamma} \cap Y|$ and the cut (X, Y) is sufficiently sparse, the total number of terminals that lie in X at the end of the current iteration is at most $2|\tilde{\Gamma}|/3$. Therefore, the budget of every terminal in $\tilde{\Gamma} \cap X$ decreases by at least 8Δ, and the total decrease in the budget is therefore at least $8\Delta \cdot |X \cap \tilde{\Gamma}|$. Since $\alpha' = \frac{1}{16\alpha_{\text{ARV}}(n) \log_{3/2} n}$, the total budget of all terminals does not increase.

We assume from now on that algorithm $\mathcal{A}_{\text{ARV}}$ returned a cut of sparsity at least $\alpha' \cdot \alpha_{\text{ARV}}(n)$. Then we are guaranteed that the set $\tilde{\Gamma}$ of terminals is α'-well-linked in C. We use the following standard definition of vertex cuts:

Definition 11. *Given a graph $\hat{G}$, a* vertex cut *in $\hat{G}$ is a partition (W, X, Y) of $V(\hat{G})$ into three disjoint subsets, with $W, Y \neq \emptyset$, such that no edge of $\hat{G}$ connects a vertex of W to a vertex of Y. We say that the cut is a* 1-vertex cut *if $|X| = 1$, and we say that it is a 2-vertex cut if $|X| = 2$.*

In the remainder of the proof we consider three cases. The first case is when graph C has a 1-vertex cut (W, X, Y), with W and Y containing at least two terminals of $\tilde{\Gamma}$ each. In this case, we delete some edges from C, decomposing it into smaller clusters, and continue to the next iteration. The second case is when C has a 2-vertex cut (W, X, Y), where both W and Y contain at least three terminals of $\tilde{\Gamma}$. In this case, we also delete some edges from C, decomposing it into smaller connected components, and continue to the next iteration. The third case is when neither of the first two cases happens. In this case, we decompose C into a single type-2 acceptable cluster, and a collection of type-1 acceptable clusters. We now proceed to describe each of the cases in turn.

Case 1. We say that Case 1 happens if there is a 1-vertex cut (W, X, Y) of C, with $|W \cap \tilde{\Gamma}|, |Y \cap \tilde{\Gamma}| \geq 2$. Set X contains a single vertex, that we denote by v. Assume w.l.o.g. that $|W \cap \tilde{\Gamma}| \leq |Y \cap \tilde{\Gamma}|$. We start with the following simple claim.

Claim 1.5.21. *$|W \cap \tilde{\Gamma}| < \mu/4$.*

Proof. Assume for contradiction that the claim is false. Consider a bi-partition (W', Y') of $V(C)$, where $W' = W \cup \{v\}$ and $Y' = Y$. We denote by E^* the set of all edges of C incident to the separator vertex v, so $|E^*| \leq \Delta$. Since $E_C(W', Y') \subseteq E^*$, $|E_C(W', Y')| \leq \Delta$. However, since vertices of $\tilde{\Gamma}$ are α'-well-linked in C, we have $|E_C(W',Y')| \geq \alpha' \cdot |W' \cap \tilde{\Gamma}| \geq \alpha' \cdot \mu/4 > \Delta$, (as $\mu = 512\Delta\alpha_{\mathrm{ARV}}(n)\log_{3/2} n$, while $\alpha' = \frac{1}{16\alpha_{\mathrm{ARV}}(n)\log_{3/2} n}$), a contradiction. □

Let E_0 be the set of all edges that connect the separator vertex v to the vertices of W. We add to $\hat{E}$ the edges of E_0. Consider now the graph $C \setminus E_0$. Note that for every connected component of $C \setminus E_0$, either $V(C') \subseteq W$, or $V(C') = Y \cup \{v\}$. We let $\mathcal{S}$ contain all components C' with $V(C') \subseteq W$. Note that every component $C' \in \mathcal{S}$ is a type-1 acceptable cluster. This is because $|\tilde{\Gamma} \cap V(C')| \leq |\tilde{\Gamma} \cap W| \leq \mu/4$, and we have created at most Δ new terminals in C': the endpoints of the edges in E_0. As $\mu > 4\Delta$, every component $C' \in \mathcal{S}$ now contains at most μ terminals. We add all components in $\mathcal{S}$ to the set $\mathcal{C}_1^I$ of inactive components (and also to the set $\mathcal{C}_1^I$). Additionally, we replace the cluster C in $\mathcal{C}^A$ by the subgraph of C induced by vertices of $Y \cup \{v\}$.

It remains to prove that the total budget of all vertices does not grow. Recall that $|\tilde{\Gamma} \cap W| \geq 2$, and the budget of each terminal $t \in \tilde{\Gamma} \cap W$ has decreased from $8\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|)$ to 1. Therefore, the total budget decrease of these terminals is at least $16\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|) - 2$. We have created at most Δ new terminals in set W. For each new terminal, its new budget is 1 since at the end of this iteration it belongs to an inactive cluster. We have created at most one new terminal in set $X \cup Y$ – the vertex v. Since $|\tilde{\Gamma} \cap W| \geq 2$, the total number of terminals in $X \cup Y$ at the end of the iteration is at most $|\tilde{\Gamma}| - 1$. Therefore, the budgets of terminals in $\tilde{\Gamma} \cap Y$ do not increase, and the budget of v is at most $8\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|)$. Altogether, the total budget increase is at most $8\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|) + \Delta$, which is less than $16\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|) - 2$, the total budget decrease of vertices of $\tilde{\Gamma} \cap W$. We conclude that the total budget of all vertices does not increase.

From now on we assume that Case 1 does not happen. We need the following simple observation.

Observation 1.5.22. *Assume that Case 1 does not happen. Let (W, X, Y) be any 1-vertex cut in C, then either $|W| = 1$, or $|Y| = 1$.*

Proof. Assume for contradiction that $|W|, |Y| > 1$. Since Case 1 does not happen, either $|W \cap \tilde{\Gamma}| \leq 1$, or $|Y \cap \tilde{\Gamma}| \leq 1$. Assume w.l.o.g. that $|W \cap \tilde{\Gamma}| \leq 1$. Let $t \in W \cap \tilde{\Gamma}$ be the unique terminal that lies in W (if it exists; otherwise t is undefined), and let u be any vertex that lies in $W \setminus \tilde{\Gamma}$ (since $|W| > 1 \geq |W \setminus \tilde{\Gamma}|$, such a vertex always exists). Note that the removal of v and t (if it exists) from G separates u from vertices of Y in G, causing a contradiction to the fact that G is 3-connected. □

Let U be the set of separator vertices of C. From the above observation, for every separator vertex $u \in U$, there is a unique vertex $u' \in V(C)$ that is a neighbor of u, such that u' has degree 1 in C. Since graph G is 3-connected, u' must belong to $\tilde{\Gamma}$. Let U' be the set of all such vertices. It is easy to verify that graph $C \setminus U'$ is 2-connected.

Case 2. We say that Case 2 happens if Case 1 does not happen, and there is a 2-vertex cut (W, X, Y), with $|W \cap \tilde{\Gamma}|, |Y \cap \tilde{\Gamma}| \geq 3$. Set X contains exactly two vertices, that we denote by x, y. Assume w.l.o.g. that $|W \cap \tilde{\Gamma}| \leq |Y \cap \tilde{\Gamma}|$. The algorithm for Case 2 is very similar to the algorithm for Case 1. We start with the following simple claim, that is similar to Claim 1.5.21, and its proof is almost identical.

Claim 1.5.23. $|W \cap \tilde{\Gamma}| < \mu/4$.

Proof. Assume for contradiction that the claim is false. Consider a bi-partition (W', Y') of $V(C)$, where $W' = W \cup \{x, y\}$ and $Y' = Y$. Let E^* be the set of all edges of C incident to vertices x or y, so $|E^*| \leq 2\Delta$. Since $E_C(W', Y') \subseteq E^*$, $|E_C(W', Y')| \leq 2\Delta$. However, since vertices of $\tilde{\Gamma}$ are α'-well-linked in C, we have $|E_C(W', Y')| \geq \alpha' \cdot |W' \cap \tilde{\Gamma}| \geq \alpha' \cdot \mu/4 > 2\Delta$, (as $\mu = 512\Delta\alpha_{\text{ARV}}(n)\log_{3/2} n$, while $\alpha' = \frac{1}{16\alpha_{\text{ARV}}(n)\log_{3/2} n}$), a contradiction. □

Let E_0 be the set of all edges that connect the vertices x, y to the vertices of W. We add to $\hat{E}$ the edges of E_0. Consider now the graph $C \setminus E_0$ and let $\mathcal{S}$ be the set of its connected components. Note that, for every component of $C' \in \mathcal{S}$, either $V(C') \subseteq W$ or

$V(C') \subseteq Y \cup \{x, y\}$. We let $\mathcal{S}_1 \subseteq \mathcal{S}$ contain all components C' with $V(C') \subseteq W$, and we let $\mathcal{S}_2$ contain all remaining connected components. Note that every component $C' \in \mathcal{S}_1$ is a type-1 acceptable cluster. This is because $|\tilde{\Gamma} \cap V(C')| \leq |\tilde{\Gamma} \cap W| \leq \mu/4$, and we have created at most 2Δ new terminals in C': the endpoints of edges in E_0. As $\mu > 8\Delta$, every component in $\mathcal{S}_1$ now contains at most μ terminals. We add all components in $\mathcal{S}_1$ to the set $\mathcal{C}_1^I$ (and also to $\mathcal{C}_1^I$). Additionally, we replace the cluster C in $\mathcal{C}^A$ by all components in $\mathcal{S}_2$.

It remains to prove that the total budget of all vertices does not grow. The proof again is very similar to the proof in Case 1. Recall that $|\tilde{\Gamma} \cap W| \geq 3$, and the budget of each terminal in $\tilde{\Gamma} \cap W$ has decreased from $8\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|)$ to 1. Therefore, the decrease of their total budgets is at least $24\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|) - 3$. We have created at most 2Δ new terminals in set W – the neighbors of x and y, and each such new terminal has new budget 1 since it belongs to an inactive cluster at the end of this iteration. We have created at most two new terminals in set $X \cup Y$ – terminals x and y. Since $|\tilde{\Gamma} \cap W| \geq 3$, the total number of terminals that belong to set $X \cup Y$ at the end of the current iteration is at most $|\tilde{\Gamma}| - 1$. Therefore, the budgets of terminals in $\tilde{\Gamma} \cap Y$ do not increase, and the budgets of the vertices x and y are at most $8\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|)$. Altogether, the total budget increase is at most $16\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|) + 2\Delta$, which is less than $24\Delta \cdot \log_{3/2}(|\tilde{\Gamma}|) - 3$, the total budget decrease of vertices of $\tilde{\Gamma} \cap W$. We conclude that the total budget of all vertices does not increase. From now on, we can assume that Case 1 and Case 2 did not happen.

Case 3. The third case happens if neither Case 1 nor Case 2 happen. Recall that we have denoted by U the set of all separator vertices of the cluster C. We have defined U' to be the set of all vertices u', such that u' has degree 1 in C, and it has a neighbor in U. We have also shown that $U' \subseteq \tilde{\Gamma}$. Our first step is to add to the edge set $\hat{E}$ every edge of $E(C)$ that connects a vertex of U to a vertex of U', and let $C' = C \setminus U'$. Every vertex $u' \in U'$ is now an isolated vertex in $G \setminus \hat{E}$. For each such vertex $u' \in U'$, we add the cluster $\left\{u'\right\}$ to the set $\mathcal{C}_1^I$ of inactivate type-1 acceptable clusters (and also to $\mathcal{C}^I$). Notice that every vertex

$u \in U$ now becomes a terminal. We denote by $\tilde{\Gamma}' = (\tilde{\Gamma} \cup U) \setminus U'$ the set of terminals in the current cluster C'. From the above discussion, $|\tilde{\Gamma}'| \leq |\tilde{\Gamma}|$, and cluster C' is 2-connected. If $|\tilde{\Gamma}'| \leq \mu$, then C' is a type-1 acceptable cluster. We then add it to the sets $\mathcal{C}_1^I$ (and also to $\mathcal{C}^I$), and terminate the current iteration. Note that the total budget of all vertices does not increase. This is because, before the current iteration, $|\tilde{\Gamma}| \geq \mu$ held, and every vertex in $\tilde{\Gamma}$ had budget at least $8\Delta \log \mu$; while at the end of the current iteration, every terminal in $\tilde{\Gamma}' \cup U'$ has budget 1, and $|\tilde{\Gamma}' \cup U'| \leq 2\mu$. Therefore, we assume from now on that $|\tilde{\Gamma}'| > \mu$, and we refer to the vertices of $\tilde{\Gamma}'$ as terminals. In the remainder of this iteration, we will split the cluster C' into a single type-2 acceptable cluster, and a collection of type-1 acceptable clusters, and we will prove that the total budget of all vertices does not increase. Before we proceed further, we first prove the the following observations about the graph C' that we will use later.

Observation 1.5.24. *The set $\tilde{\Gamma}'$ of terminals is α'-well-linked in C'.*

Proof. Consider any partition (X, Y) of $V(C')$. Then we can augment (X, Y) to a partition (X', Y') of $V(C)$ as follows. Start with $X' = X$ and $Y' = Y$. For every vertex $u \in U$, if $u \in X$, then we add its unique neighbor in U' to X', otherwise we add it to Y'. Note that $|\tilde{\Gamma} \cap X'| \geq |\tilde{\Gamma}' \cap X|$. This is because $\tilde{\Gamma}' = (\tilde{\Gamma} \cup U) \setminus U'$, and for every vertex $u \in U \cap X$, while u may or may not belong to $\tilde{\Gamma}$, it always has a neighbor in $U' \cap \tilde{\Gamma}$. Similarly, $|\tilde{\Gamma} \cap Y'| \geq |\tilde{\Gamma}' \cap Y|$. Since the set $\tilde{\Gamma}$ of terminals is α'-well-linked in C, we get that $|E_{C'}(X, Y)| = |E_C(X', Y')| \geq \alpha' \cdot \min \left\{ |\tilde{\Gamma} \cap X'|, |\tilde{\Gamma} \cap Y'| \right\} \geq \alpha' \cdot \min \left\{ |\tilde{\Gamma}' \cap X|, |\tilde{\Gamma}' \cap Y| \right\}$. Therefore, the set $\tilde{\Gamma}'$ of terminals is α'-well-linked in C'. □

Observation 1.5.25. *For every 2-vertex cut (W, X, Y) of C' with $|W \cap \tilde{\Gamma}'| \leq |Y \cap \tilde{\Gamma}'|$, $|W \cap \tilde{\Gamma}'| \leq 2$.*

Proof. Let (W, X, Y) be a 2-vertex cut of C' with $|W \cap \tilde{\Gamma}'| \leq |Y \cap \tilde{\Gamma}'|$. We augment it to a 2-vertex cut (W', X, Y') of C as follows. Start with $W' = W$ and $Y' = Y$. For every vertex $u \in U$, if $u \in W$, then we add its unique neighbor in U' to W, otherwise we add it to Y'. It is immediate to verify that (W', X', Y') is indeed a 2-vertex cut in C, and that $|W' \cap \tilde{\Gamma}| \geq |W \cap \tilde{\Gamma}'|$ and $|Y' \cap \tilde{\Gamma}'| \geq |Y \cap \tilde{\Gamma}'|$. Since we assumed that Case 2 does not happen, $|W' \cap \tilde{\Gamma}| \leq 2$ must hold, and so $|W \cap \tilde{\Gamma}'| \leq 2$. □

The next observation gives us a planar drawing of C'.

Observation 1.5.26. *There is a pseudo-block B_0 in the block decomposition of $G \setminus E_1$ that is not contained in a component of $\mathcal{C}_1'$, such that $C' \subseteq B_0$. In particular, the associated drawing $\hat{\psi}_{B_0}$ of B_0 naturally induces a planar drawing $\psi_{C'}$ of the cluster C'.*

Proof. Let $H_1 = G \setminus E_1$ and $H_2 = G \setminus \hat{E}$. Clearly $E_1 \subseteq \hat{E}$. Denote by $\mathcal{B}(H_1)$ the block decomposition of the graph H_1. Since C' is a connected graph, there is a pseudo-block in $\mathcal{B}(H_1)$ that contains C' a subgraph. We denote this block by B_0. Note that it is impossible that $V(B_0) \subseteq V(\hat{C})$ for a component $\hat{C} \in \mathcal{C}_1'$, since initially all clusters in $\mathcal{C}_1'$ belong to $\mathcal{C}_1^I$ and will not be processed. □

We now proceed to split the cluster C' into one type-2 acceptable cluster and a collection of type-1 clusters. Recall that C' is 2-connected. We use the algorithm from Theorem 1.4.1, to obtain a the block decomposition $\mathcal{L}$ of C', and let $\tau = \tau(\mathcal{L})$ be the decomposition tree associated with the decomposition $\mathcal{L}$. We let $B \in \mathcal{L}$ be a pseudo-block that contains at least $\mu/4$ terminals of $\tilde{\Gamma}'$, and among all pseudo-blocks with this property, maximizes the distance in τ between its corresponding vertex $v(B)$ and the root of τ, breaking ties arbitrarily. Notice that such a pseudo-block always exists, since C' belongs to $\mathcal{L}$ as a pseudo-block and contains at least μ terminals of $\tilde{\Gamma}'$. Let $v(B_1), \ldots, v(B_q)$ denote the child vertices of $v(B)$ in τ, and let B^c denote the complement block of B. We denote the endpoints of B by (x, y), and, for all $1 \leq i \leq q$, the endpoints of B_i by (x_i, y_i). Recall that $\mathcal{N}(B) = \left\{B^c, B_1, \ldots, B_q\right\}$. We use the following simple observations and their corollaries.

Observation 1.5.27. *Let $\hat{B}$ be a pseudo-block in $\mathcal{L}$ such that $v(\hat{B})$ is a leaf of the tree τ. Then $\hat{B}$ contains terminal of $\tilde{\Gamma}'$ that is not an endpoint of $\hat{B}$. Moreover, if B^c is defined, then it contains a terminal of $\tilde{\Gamma}'$, that is not one of its endpoints.*

Proof. Let x, y be the endpoints of $\hat{B}$. Assume that $\hat{B}$ does not contain a terminal that does not belong to $\{x, y\}$, then the removal of $\{x, y\}$ separates $V(\hat{B}) \setminus \{x, y\}$ from $V(G) \setminus V(\hat{B})$, contradicting the fact that G is 3-connected. The proof for B^c is similar. □

We obtain the following immediate corollary.

Corollary 1.5.28. $|\mathcal{N}(B)| \leq |\tilde{\Gamma}'|$.

Observation 1.5.29. *For all $1 \leq i \leq q$, at most two vertices of $V(B_i) \setminus \{x_i, y_i\}$ belong to $\tilde{\Gamma}'$. Similarly, at most two vertices of $V(B^c) \setminus \{x, y\}$ belong to $\tilde{\Gamma}'$.*

Proof. Fix some $1 \leq i \leq q$, and consider a 2-vertex cut (W, X, Y) of C', where $W = V(B_i) \setminus \{x_i, y_i\}$, $X = \{x_i, y_i\}$, and $Y = V(C') \setminus (X \cup W)$. From the definition of the block B, $|W \cap \tilde{\Gamma}'| < \mu/4$. Since $|\tilde{\Gamma}'| > \mu$, $|Y \cap \tilde{\Gamma}'| \geq 3\mu/4 - 2 > \mu/4 > |W \cap \tilde{\Gamma}'|$. From Observation 1.5.25, $|W \cap \tilde{\Gamma}'| \leq 2$. Therefore, $V(B_i) \setminus \{x_i, y_i\}$ contains at most two terminals of $\tilde{\Gamma}'$. The proof that $V(B^c) \setminus \{x, y\}$ contains at most two terminals of $\tilde{\Gamma}'$ is similar. □

We obtain the following immediate corollary of Observations 1.5.27 and 1.5.29.

Corollary 1.5.30. *There are at most two leaves in tree τ that are not descendants of $v(B)$. Moreover, for all $1 \leq i \leq q$, there are at most two leaves in the subtree of τ rooted at $v(B_i)$.*

We now describe the next steps for processing cluster C', starting with a high-level intuitive overview. From Observation 1.5.26, the associated drawing $\hat{\psi}_{B_0}$ of B_0 naturally induces a planar drawing $\psi_{C'}$ of C'. One can show that C' is a type-2 acceptable cluster with respect to the drawing $\psi_{C'}$, except that we cannot ensure that the size of set $S_2(C')$ – the set of all vertices that participate in 2-separators in C', is sufficiently small (recall that the requirement in the definition of type-2 acceptable clusters is that $|S_2(C')| \leq O(\Delta|\tilde{\Gamma}'|))$. To see this, consider the situation where the sub-tree of τ rooted at the child vertex $v(B_i)$ of $v(B)$, that we denote by τ_i, is a long path. The cardinality of set $S_2(C')$ may be as large as the length of the path, while there are only at most two terminals in B_i.

In order to overcome this difficulty, we need to "prune" the sub-tree τ_i. From Observation 1.5.29, tree τ_i may contain at most two leaves. Assume first for simplicity that τ_i contains exactly one leaf, so τ_i is a path. If we denote by $v(B_i^1), v(B_i^2), \ldots, v(B_i^r)$ the vertices that appear on path τ_i in this order, with $B_i^1 = B_i$. We simply add to $\hat{E}$ all edges of B_i^2 incident to its endpoints, and replace B_i^2 with a fake edge connecting its endpoints in C'. The block B_i^2 then decomposes into a number of a type-1 acceptable clusters. Notice that now the total number of vertices of $S_2(C')$ that block B_i contributes is $O(1)$. If tree τ_i has two leaves, then the pruning process for τ_i is slightly more complicated but similar. We treat the complement block B^c similarly as blocks in $\left\{B_1, \ldots, B_q\right\}$.

We now provide a formal proof. We start with the cluster $\hat{C} = C'$ and the set $A_{\hat{C}} = \emptyset$ of fake edges, and then we iterate. In every iteration, we process a distinct block of $\mathcal{N}(B)$, and modify the cluster $\hat{C}$ by deleting some edges and vertices and adding some fake edges to $A_{\hat{C}}$ and to $\hat{C}$. Throughout the algorithm, we will maintain the following invariants:

I1. $\hat{C}$ is a simple graph;

I2. each 2-separator of $\hat{C}$ is also a 2-separator of C'; and

I3. graph $\hat{C} \setminus A_{\hat{C}}$ is 2-connected.

We now describe an iteration. Consider a child block B_i of block B (the block B^c will be processed similarly), and let τ_i be the sub-tree of τ rooted at $v(B_i)$. As our first step, we construct a set V_i of nodes in τ_i that we will use in order to "prune" block B_i, using the following observation.

Observation 1.5.31. *There is an efficient algorithm, that constructs a subset $V_i \subseteq V(\tau_i)$ containing at most two vertices (where possibly $V_i = \emptyset$), that satisfies the following properties:*

- *for each vertex $\hat{v} \in V_i$, $\hat{v}$ has degree exactly 2 in τ_i, it is not $v(B_i)$ or one of its children, and the parent of $\hat{v}$ in τ_i has degree exactly 2;*
- *if $|V_i| = 2$ then neither vertex is a descendant of the other in τ_i;*
- *if $v(B^*) \in V_i$, then graph $\tilde{B}^*$ is not isomorphic to K_3;*
- *for each vertex $\hat{v} \in V_i$, there is some ancestor vertex $v(B^p)$ of $\hat{v}$ in τ_i, that is not $v(B_i)$, such that $\tilde{B}^p$ is not isomorphic to K_3, and every vertex on the unique path in tree τ connecting $\hat{v}$ to $v(B^p)$ has degree exactly 2; and*
- *if we let τ_i' be the sub-tree obtained from τ_i after we delete, for every vertex $\hat{v} \in V_i$, the sub-tree rooted at the child vertex of $\hat{v}$, then $|V(\tau_i')| \leq 500$.*

Proof. We fix a child block B_i of B and show how to construct vertex set V_i. We say that a vertex $v(B^*) \in \tau_i$ is bad iff graph $\tilde{B}^*$ is isomorphic to K_3. We say that a path $P \subseteq \tau_i$ is bad iff every vertex of P has degree exactly 2 in τ_i, and it is a bad vertex.

Observation 1.5.32. *If P is a bad path in τ_i, then P contains at most 20 vertices.*

Proof. Let P be a bad path in τ_i, and assume for contradiction that it contains 21 vertices. Then every vertex $v(B^*) \in V(P)$ has exactly one child in τ_i, and graph $\tilde{B}^*$ is isomorphic to K_3. It is then easy to verify that graph C' must contain a path P' containing at least five vertices, each of which has degree exactly 2 in C' (every vertex on P' is an endpoint of some block in $\{B^* \mid v(B^*) \in P\}$). Since graph G is 3-connected, every vertex of P' must lie in $\tilde{\Gamma}'$. But then there are at least three vertices of $V(B_i) \setminus \{x_i, y_i\}$ that lie in $\tilde{\Gamma}'$, contradicting Observation 1.5.29. □

From Corollary 1.5.30, τ_i contains at most two leaves. Therefore, there is at most one node in τ_i that has degree 3, and all other nodes has degree 1 or 2. We consider the following cases:

1. if τ_i contains no degree-3 node, and $|V(\tau_i)| \leq 50$, then we set $V_i = \emptyset$;

2. if τ_i contains no degree-3 node, and $|V(\tau_i)| > 50$, then we let V_i consist of a single vertex v, that is at distance at least 25 and at most 50 from the root, and is not a bad vertex; from Observation 1.5.32, such a vertex must exist;

3. if τ_i contains a degree-3 node v', and the distance between v' and $v(B_i)$ is at least 51 in τ_i, then V_i consists of a single vertex v, that is defined exactly like in the previous case;

4. if τ_i contains a degree-3 node v', and the distance between v' and $v(B_i)$ is at most 51 in τ_i, then the subtree of τ_i rooted at v' can be viewed as the union of two paths that share a common endpoint v'; we denote by τ_i^1 and τ_i^2 the two paths. We select at most one vertex on each of the two paths to add to V_i, exactly like in Cases 1 and 2.

It is easy to see that V_i contains most two vertices, and satisfies the properties in Observation 1.5.31. $\square$

We consider the vertices in V_i one-by-one. Let $v(B^{**}) \in V_i$ be a vertex of V_i. Let $v(B^*)$ be the unique child vertex of $v(B^{**})$ in τ_i, and let (x^*, y^*) be the endpoints of block B^*. Note that block B^* contains a path P connecting x^* to y^*. Let $\tilde{E}$ be the set of all edges of B^* that are incident to x^* or to y^*. We add the edges of $\tilde{E}$ to $\hat{E}$. Consider now the graph $\hat{C} \setminus \tilde{E}$. It is immediate to verify that there is one component $\tilde{C}$ of this graph, containing all vertices of $(\hat{C} \setminus B^*) \cup \{x^*, y^*\}$, and all remaining components are a type-1 acceptable clusters that are contained in B^*. We add the components of the latter type to $\mathcal{C}_1^I$ (and also to $\mathcal{C}^I$). If the edge (x^*, y^*) does not lie in $\tilde{C}$, then we add the edge $e' = (x^*, y^*)$ to the set

$A_{\hat{C}}$ of fake edges, and let $P(e') = P$ be its embedding. It is immediate to verify that there is some type-1 acceptable cluster in $\hat{C} \setminus \tilde{E}$ that contains the path $P(e') \setminus \{x^*, y^*\}$. Finally, we update the cluster $\hat{C}$ by first removing all vertices and edges of $B^* \setminus \{x^*, y^*\}$ form it, and then adding the edge (x^*, y^*) if it does not belong to $\hat{C}$. We say that the block B^* is *eliminated* when processing B_i.

The processing of the block B^c is very similar to the processing of the blocks $B_1, \ldots, B_q$, though the details are somewhat more tedious and are omitted here.

Claim 1.5.33. *After each vertex $v(B^{**}) \in V_i$ is processed, the invariants continue to hold.*

Proof. We denote by C_1 and C_2 the cluster $\hat{C}$ before and after vertex $v(B^{**})$ was processed, respectively. Similarly, we denote by A_1 and A_2 the set $A_{\hat{C}}$ of vertices before and after vertex $v(B^{**})$ was processed, respectively. We assume that all invariants held before vertex $v(B^{**})$ was processed. It is immediate to verify that C_2 is a simple graph, since we add a fake edge to $\hat{C}$ only if $\hat{C}$ did not contain it. Therefore, Invariant I1 continues to hold.

Next, we show that $C_2 \setminus A_2$ is 2-connected. Assume for contradiction that this is not the case, and let (W, X, Y) be a 1-vertex cut of $C_2 \setminus A_2$. Let B^* be the child block of B^{**}, and let x^*, y^* be its endpoints. We claim that either (i) $x^* \in W$, $y^* \in Y$, or (ii) $x^* \in Y$, $y^* \in W$ must hold. Indeed, if neither of these holds, then can assume w.l.o.g. that $x^*, y^* \in W \cup X$. Then, by adding to W all vertices of $B^* \setminus \{x^*, y^*\}$, we obtain a 1-vertex cut in $C_1 \setminus A_1$, contradicting the assumption that Invariant I3 held for C_1. We now assume without loss of generality that $x^* \in W$ and $y^* \in Y$.

Since graph $\tilde{B}^{**}$ is not isomorphic to K_3, it must be 3-connected. Therefore, graph $\tilde{B}^{**}$ contains three internally disjoint paths connecting x^* to y^*. Two of these paths, that we denote by P_1, P_2, do not contain the fake edge (x^*, y^*). If we denote by (x^{**}, y^{**}) the endpoints of the block B^{**}, then at least one of these two paths (say P_1) is disjoint from the fake parent-edge (x^{**}, y^{**}) of block B^{**}. Path P_2 may contain the fake edge (x^{**}, y^{**}), but all other edges of P_2 must be real edges of $\tilde{B}^{**}$, as vertex $v(B^{**})$ only has one child in τ_i.

Note that there must be a path $P' \subseteq C_2$ that connects x^{**} to y^{**}, and is internally disjoint from B^{**} (if we denote by $v(B^p)$ the parent vertex of $v(B^{**})$, then, if $\tilde{B}^p$ is not isomorphic to K_3, such a path exists in $\tilde{B}^p$, as graph $\tilde{B}^p$ is 3-connected, and only has one child block. Otherwise, we let $v(B^p)$ be an ancestor of $v(B^{**})$ that is closest to $v(B^{**})$, such that $\tilde{B}^p$ is not isomorphic to K_3, and $v(B^p)$ has only one child in τ_i. We can define the path P' using the graph $\tilde{B}^p$). By combining path P_2 with P' (namely, replacing the fake edge (x^{**}, y^{**}) of P_2 with the path P', if P_2 contains such an edge), we obtain a path P_3 connecting x^* to y^*, that is disjoint from P_1, and contains no fake edges. Note that both paths P_1, P_3 are contained in C_2. But then both paths P_1, P_3 must contain the separator vertex that lies in X, a contradiction. Therefore, Invariant I3 continues to hold.

In order to show that Invariant I2 continues to hold, it suffices to show that every 2-separator in C_2 is also a 2-separator in C_1. Consider any 2-vertex cut (W, X, Y) in C_2, and assume w.l.o.g. that $x^*, y^* \in W \cup X$ (since the fake edge (x^*, y^*) belongs to C_2, the two vertices cannot lie in sets W and Y respectively). By adding all vertices of $B^* \setminus \{x^*, y^*\}$ to set W, we obtain a 2-vertex cut in C_1. Therefore, X is also a 2-separator in C_1. □

Let $\hat{C}$ be the cluster we obtain after all blocks in $\mathcal{N}(B)$ are processed. Recall that from Observation 1.5.26, there is a pseudo-block B_0 in the block decomposition of graph $G \setminus E_1$, such that B_0 is not contained in a cluster of $\mathcal{C}_1'$, and $C' \subseteq B_0$. The associated drawing $\hat{\psi}_{B_0}$ of B_0 induces a planar drawing $\psi_{\hat{C}}$ of $\hat{C}$, where for each fake edge $e \in A_{\hat{C}}$, the edge e is drawn along the drawing of the path $P(e)$ in $\hat{\psi}_{B_0}$. We prove the following Lemma in Section 1.5.5.

Lemma 1.5.34. *Cluster $\hat{C}$ is a type-2 acceptable cluster with respect to the drawing $\psi_{\hat{C}}$.*

In order to complete the proof of Theorem 1.3.1, it remains to show that if Case 3 happens, then the total budget of all vertices does not increase after the cluster C is processed. Recall that $\tilde{\Gamma}$ denotes the set of terminals in C before C was processed, and $|\tilde{\Gamma}| \geq \mu$. Let $\tilde{\Gamma}^{\mathsf{new}}$ be the set of all new terminals that were added to $\tilde{\Gamma}$ over the course of processing C. Notice that every vertex in $\tilde{\Gamma}$ had a budget of at least 8Δ before C was processed. After C was processed, the budget of every terminal in $\tilde{\Gamma} \cup \tilde{\Gamma}^{\mathsf{new}}$ became 1. Therefore, in order to show that the total budget of all vertices did not grow, it is sufficient to show that $|\tilde{\Gamma}^{\mathsf{new}}| \leq 7\Delta|\tilde{\Gamma}|$. Recall that we have created a set U of at most $|\tilde{\Gamma}|$ terminals, and, whenever a block B^* was eliminated, we created at most 2Δ new terminals (neighbors of endpoints of B^*). Since each such block B^* contained at least one terminal of $\tilde{\Gamma}' = (\tilde{\Gamma} \cup U) \setminus U'$, the total number of new terminals that we have created is bounded by $|\tilde{\Gamma}| \cdot 4\Delta$. Therefore, the total budget of all vertices does not grow.

In order to complete the proof of Theorem 1.3.1, it remains to prove Lemma 1.5.34, which we do next.

1.5.5 Proof of Lemma 1.5.34

Recall that $\tilde{\Gamma}$ denotes the set of all terminals that belonged to cluster C before it was processed. We denote by Γ^* the final set of terminals that lie in $\hat{C}$ after C is processed.

Recall that we have already established that graph $\hat{C}$ is planar, and defined its planar drawing $\psi_{\hat{C}}$. From the invariants, $\hat{C}$ is a simple graph, and $\hat{C} \setminus A_{\hat{C}}$ is 2-connected.

Next, we bound the cardinality of the set $S_2(\hat{C})$. Recall that the invariants ensure that a 2-separator in $\hat{C}$ is also a 2-separator in C', so $S_2(\hat{C}) \subseteq S_2(C')$. Moreover, Theorem 1.4.1 ensures that for every vertex $x \in S_2(C')$, either x is an endpoint of a block of $\mathcal{L}$, or there is a vertex x' that is a neighbor of x in C', and it is an endpoint of a block of $\mathcal{L}$. In the former case, we denote by $B(x)$ the largest (with respect to $|V(B(x))|$) block of $\mathcal{L}$ such that x is an endpoint of $B(x)$, and in the latter case, we denote by $B(x)$ the largest block of $\mathcal{L}$, such that x' is an endpoint of $B(x)$. Notice that, if B^* is a block that was eliminated, and x lies in B^* but is not an endpoint of B^*, then x is not a vertex of $\hat{C}$, and it does not belong to $S_2(\hat{C})$. Our algorithm ensures that, for every child block B_i of B, there are at most $O(1)$ vertices x, such that $x \in V(\hat{C})$, and x serves as an endpoint of a block that is a descendant of B_i in τ_i. Therefore, at most $O(\Delta)$ vertices of $S_2(\hat{C})$ may lie in $B_i \cap \hat{C}$. On the other hand, at least one terminal of Γ^* lies in $B_i \cap \hat{C}$. We *charge* these separator vertices to that terminal. Similarly, if B^c is defined, then at most $O(\Delta)$ vertices of $S_2(\hat{C})$ may lie in $B^c \cap \hat{C}$, and at least one terminal of Γ^* lies in $\hat{C} \cap B^c$. Therefore, altogether, $|S_2(\hat{C})| \leq O(\Delta|\Gamma^*|)$, as required.

Next, we show that cluster $\hat{C}$ has the well-linkedness property.

Claim 1.5.35. *The set Γ^* of terminals is α-well-linked in $\hat{C} \setminus A_{\hat{C}}$.*

Proof. We start by showing that the set Γ^* of terminals is 2α-well-linked in $\hat{C}$. Let (X, Y) be any partition of $V(\hat{C})$. Denote $\Gamma_X = \Gamma^* \cap X$ and $\Gamma_Y = \Gamma^* \cap Y$. It suffices to show that $|E_{\hat{C}}(X, Y)| \geq 2\alpha \cdot \min\{|\Gamma_X|, |\Gamma_Y|\}$.

Let $\mathcal{M}$ be the set of all blocks B^* that our algorithm eliminated, and let V^* be the set of all vertices that serve as endpoints of the blocks in $\mathcal{M}$.

Recall that, in addition to the terminals of $\tilde{\Gamma}' = \tilde{\Gamma} \cup U$, the sets Γ_X, Γ_Y of terminals may also contain vertices of V^*. We call such terminals *new terminals*.

We further partition sets Γ_X and Γ_Y as follows. Let $\Gamma_X^1 \subseteq \Gamma_X$ contain all terminals t, such that there is some block $B^* \in \mathcal{M}$ with t being one of its endpoints, and the other endpoint t' of B^* lies in Y. We let $\Gamma_X^2 = \Gamma_X \setminus \Gamma_X^1$. The partition (Γ_Y^1, Γ_Y^2) of Γ_Y is defined similarly. Since the endpoints of every block in $\mathcal{M}$ are connected by a fake edge in $\hat{C}$, it is immediate to verify that

$$|E_{\hat{C}}(X, Y)| \geq |\Gamma_X^1| \text{ and } |E_{\hat{C}}(X, Y)| \geq |\Gamma_Y^1|. \tag{1.1}$$

Next, we construct a cut (X', Y') in graph C' based on the cut (X, Y) of $\hat{C}$, and then use the well-linkedness of the terminals in $\tilde{\Gamma}'$ in graph C' (from Observation 1.5.24) to bound $|E_{\hat{C}}(X, Y)|$. We start with $X' = X$ and $Y' = Y$, and then consider the blocks $B^* \in \mathcal{M}$ one-by-one. Denote the endpoints of B^* by (x^*, y^*).

Note that x^*, y^* may not belong to the terminal set $\tilde{\Gamma}' = \tilde{\Gamma} \cup U$. However, since we have assumed that graph G is 3-connected, vertex set $V(B^*) \setminus \{x^*, y^*\}$ must contain a terminal of

$\tilde{\Gamma}'$. We denote this terminal by t_{B^*}, and we will view this terminal as "paying" for x^* and y^* (if $x^*, y^* \in \Gamma^2_X \cup \Gamma^2_Y$). If both $x^*, y^* \in X$, then we add all vertices of $V(B^*) \setminus \{x^*, y^*\}$ to X, and otherwise we add them to Y. Notice that, if both x^*, y^* lie in the same set in $\{X, Y\}$, then we do not increase the number of edges in the cut (X, Y). Assume now that $x^* \in X$ and $y^* \in Y$ (the other case is symmetric). In this case, we have increased the number of edges in the cut (X, Y) by at most Δ, by adding all edges that are incident to x^* to this cut. Note however that the edge (x^*, y^*) already belonged to this cut (possibly as a fake edge), so we charge this increase in the cut size to this edge. Once all blocks $B^* \in \mathcal{M}$ are processed in this way, we obtain a cut (X', Y') in graph C'. From the above discussion:

$$|E_{C'}(X', Y')| \leq \Delta \cdot |E_{\hat{C}}(X, Y)|. \tag{1.2}$$

Consider now the terminals of Γ_X. Clearly $\Gamma_X \cap \tilde{\Gamma}' \subseteq X'$. For each terminal in Γ^2_X, we have added a terminal of $\tilde{\Gamma}'$ to X' that pays for it, while each newly added terminal of $X' \cap \tilde{\Gamma}'$ pays for at most two terminals in Γ^2_X. Therefore, $|\Gamma^2_X| \leq 2|\tilde{\Gamma}' \cap X'|$, and similarly $|\Gamma^2_Y| \leq 2|\tilde{\Gamma}' \cap Y'|$. Since the set $\tilde{\Gamma}'$ of terminals was α'-well-linked in C', $|E_{C'}(X', Y')| \geq \alpha' \cdot \min\left\{|\tilde{\Gamma}' \cap X'|, |\tilde{\Gamma}' \cap Y'|\right\} \geq \frac{\alpha'}{2} \cdot \min\left\{|\Gamma^2_X|, |\Gamma^2_Y|\right\}$. Combining this with Equation 1.2, we get that $|E_{\hat{C}}(X, Y)| \geq \frac{\alpha'}{2\Delta} \cdot \min\left\{|\Gamma^2_X|, |\Gamma^2_Y|\right\}$. Combining this latter equation with Equation 1.1, and using $\alpha' = 8\Delta\alpha$, we conclude that $|E_{\hat{C}}(X, Y)| \geq \frac{\alpha'}{4\Delta} \cdot \min\{|\Gamma_X|, |\Gamma_Y|\} \geq 2\alpha \cdot \min\{|\Gamma_X|, |\Gamma_Y|\}$. Therefore, the set Γ^* of terminals is 2α-well-linked in $\hat{C}$.

We are now ready to complete the proof of Claim 1.5.35. Consider any fake edge $e = (x^*, y^*) \in A_{\hat{C}}$. Then x^*, y^* are endpoints of some block B^* that we have eliminated. Recall that, if vertex $v(B^{**})$ is the parent of $v(B^*)$ in the corresponding decomposition tree τ_i (or the tree associated with B^c), then $v(B^{**})$ has degree 2 in the tree, and moreover, $\tilde{B}^{**}$ is not isomorphic to K_3. Therefore, graph $\tilde{B}^{**}$ is 3-connected, and contains a path connecting x^* to y^*, that does not contain any fake edges. We denote this path by $P'(e)$. It is immediate to verify that the paths in $\left\{P'(e) \mid e \in A_{\hat{C}}\right\}$ are mutually disjoint. Consider now some cut

(X,Y) in $\hat{C}\setminus A_{\hat{C}}$. Recall that $|E_{\hat{C}}(X,Y)| \geq 2\alpha \cdot \min\{|\Gamma^* \cap X|, |\Gamma^* \cap Y|\}$. Let $A' \subseteq E_{\hat{C}}(X,Y)$ be the set of all fake edges in $E_{\hat{C}}(X,Y)$. Since, for each fake edge $e \in A'$, path $P'(e)$ must contain a real edge in $E_{\hat{C}}(X,Y)$, we get that $|E_{\hat{C}\setminus A_{\hat{C}}}(X,Y)| = |E_{\hat{C}}(X,Y) \setminus A'| \geq |E_{\hat{C}}(X,Y)|/2 \geq \alpha \cdot \min\{|\Gamma^* \cap X|, |\Gamma^* \cap Y|\}$. We conclude that the set Γ^* of terminals is α-well-linked in $\hat{C} \setminus A_{\hat{C}}$. □

Lastly, the next claim will complete the proof of Lemma 1.5.34.

Claim 1.5.36. *Cluster $\hat{C}$ has the bridge consistency property.*

Proof. We denote by ρ the drawing of $\hat{C} \setminus A_{\hat{C}}$ induced by the drawing $\psi_{\hat{C}}$. We now show that, for each bridge $R \in \mathcal{R}_G(\hat{C} \setminus A_{\hat{C}})$, there is a face in the drawing ρ, whose boundary contains all vertices of $L(R)$. Recall that, from Observation 1.5.26, all vertices of $V(\hat{C})$ belong to some block B_0 in the block decomposition of $G \setminus E_1$ that is not contained in a component of $\mathcal{C}_1'$, and the drawing $\psi_{\hat{C}}$ of $\hat{C}$ is induced by the associated drawing $\hat{\psi}_{B_0}$ of B_0. In particular, block B_0 is a good block.

Assume for contradiction that $\hat{C}$ does not have the bridge property, and let $R \in \mathcal{R}_G(\hat{C} \setminus A_{\hat{C}})$ be a witness bridge for $\hat{C}$ and ρ, so no face of the drawing ρ contains all vertices of $L(R)$ on its boundary. Let $\mathcal{F}$ be the set of all faces of the drawing ρ. For every vertex $v \in V(B_0) \setminus V(\hat{C})$, there is a unique face $F(v) \in \mathcal{F}$, such that the image of vertex v in $\hat{\psi}_{B_0}$ lies in the interior of the face F. Recall that T_R is a tree whose leaves are the vertices of $L(R)$, and $T_R \setminus L(R) \subseteq R$. Assume first that $V(T_R) \cap V(B_0) = L(R)$. In other words, no vertex of $T_R \setminus L(R)$ is a vertex of B_0. In this case, there is some bridge $R' \in \mathcal{R}_G(B_0)$ of B_0 that contains $T_R \setminus L(R)$, and so $L(R) \subseteq L(R')$. Since B_0 is a good pseudo-block, there must be a face in the drawing $\hat{\psi}_{B_0}$, the associated drawing of B_0, whose boundary contains all vertices of $L(R)$. Therefore, for some face $F \in \mathcal{F}$, all vertices of $L(R)$ must lie on the boundary of F. Assume now that there is some vertex $v \in V(T_R) \cap V(B_0)$ that does not lie in $L(R)$. Let $F(v)$ be the face of $\mathcal{F}$ whose interior contain the image of v in $\hat{\psi}_{B_0}$ (recall that $v \notin V(\hat{C})$, so it may not lie on a boundary of a face in $\mathcal{F}$). We will show that all vertices of $L(R)$ must lie on the boundary of $F(v)$, leading to a contradiction. Let u be an arbitrary vertex of $L(R)$. It suffices to show that u must lie on the boundary of the face $F(v)$. Let $P \subseteq T_R$ be the unique path connecting v to u in T_R. Let $v = v_1, v_2, \ldots, v_r = u$ be all vertices of P that belong to $V(B_0)$, and assume that they appear on P in this order. It remains to prove the following observation.

Observation 1.5.37. $F(v_1) = F(v_2) = \cdots = F(v_{r-1})$. *Moreover, v_r lies on the boundary of $F(v_{r-1})$.*

Proof. Fix some $1 \leq i \leq r-1$. Assume that the observation is false for some v_i and v_{i+1}. Then there is some face $F' \in \mathcal{F}$, such that v_i lies in the interior of F', but v_{i+1} does not lie in the interior or on the boundary of F' (the latter case is only relevant for $i = r-1$). Since the boundary of F' separates v_i from v_{i+1}, these two vertices cannot lie on the boundary of the same face in the drawing $\hat{\psi}_{B_0}$ of B_0.

Let σ_i be the subpath of P between v_i and v_{i+1}. If σ_i consists of a single edge connecting v_i and v_{i+1}, then either it is a bridge in $\mathcal{R}_G(B_0)$, or it is an edge of B_0. In either case, the endpoints of σ_i must lie on the boundary of a single face of the drawing $\hat{\psi}_{B_0}$, a contradiction (we have used the fact that B_0 is a good pseudo-block and is therefore a planar graph). Otherwise, let σ'_i be obtained from σ_i by deleting the two endpoints from it. Then there must be a bridge $R' \in \mathcal{R}_G(B_0)$ containing σ'_i, with $v_i, v_{i+1} \in L(R')$. But then, since B_0 is a good pseudo-block, there must be a face in the drawing $\hat{\psi}_{B_0}$ of B_0, whose boundary contains v_i and v_{i+1}, leading to a contradiction. □

□

1.6 Computing Drawings of Type-1 Acceptable Clusters – Proof of Theorem 1.3.3

In this section we prove Theorem 1.3.3. We fix a cluster $C_i \in \mathcal{C}_1$. For convenience of notation, we omit the subscript i in the remainder of this section, so in particular χ_i will be denoted by χ. Recall from Property P2 that we have assumed, every terminal $t \in \Gamma(C)$ has degree 1 in C, and degree 2 in G. In particular, for each terminal $t \in \Gamma(C)$, there is exactly one bridge $R \in \mathcal{R}_G(C)$ with $t \in L(R)$. We start by defining a new graph C^+, that is obtained from graph C, as follows. For every bridge $R \in \mathcal{R}_C(G)$, we consider an arbitrary ordering $\left\{t_1(R), t_2(R), \ldots, t_{|L(R)|}(R)\right\}$ of the vertices of $L(R)$, and we add a set $E'(R)$ of $|L(R)|$ edges, connecting these vertices into a cycle according to this ordering. In other words, $E'(R) = \{(t_i(R), t_{i+1}(R)) \mid 1 \leq i \leq |L(R)|\}$, where the indexing is modulo $|L(R)|$. We denote the cycle defined by the edges of $E'(R)$ and vertices of $L(R)$ by J_R. Let $E' = \bigcup_{R \in \mathcal{R}_G(C)} E'(R)$. Then $C^+ = C \cup E'$. We start with the following two useful observations regarding the new graph C^+.

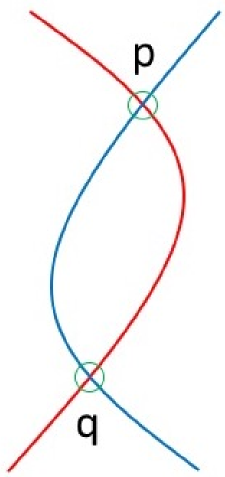
Observation 1.6.1. *Graph C^+ is 3-connected.*

Proof. Assume otherwise, and let $\{x, y\}$ be a 2-separator for graph C^+. Recall that graph C is connected, and, since every vertex $t \in \Gamma(C)$ has degree 1 in C, it cannot be the case that both x and y are terminals in $\Gamma(C)$. Therefore, at least one of the two vertices x, y lies in $V(C) \setminus \Gamma(C)$; we assume w.l.o.g. that it is x. Note that, since $\{x, y\}$ is a 2-separator for C^+, there must be at least two connected components in $C^+ \setminus \{x, y\}$. We let X be the set of vertices of one such connected component, and we let $Y = V(C^+) \setminus (X \cup \{x, y\})$. Consider now any bridge $R \in \mathcal{R}_G(C)$. If $y \notin L(R)$, then, due to the edges of $E'(R)$ that connect all vertices of $L(R)$ to each other via a cycle, either all vertices of R lie in X, or all vertices of R lie in Y. In the former case, we say that bridge R belongs to X, and in the latter we say that it belongs to Y. If $y \in L(R)$, then, since $J_R \setminus \{y\}$ remains a connected graph, either all vertices of $L(R) \setminus \{y\}$ lie in X (in which case we say that bridge R belongs to X), or they all lie in Y (in which case we say that bridge R belongs to Y). Consider now a partition (X', Y') of $V(G) \setminus \{x, y\}$, that is defined as follows. For every vertex $v \in V(C)$, if $v \in X$, then we add v to X', and otherwise we add it to Y'. For every bridge $R \in \mathcal{R}_G(C)$, if R belongs to X, then we add all vertices of $V(R)$ to X', and otherwise we add them to Y'. It is easy to verify that (X', Y') is indeed a partition of $V(G) \setminus \{x, y\}$, and moreover, no edge connects a vertex of X' to a vertex of Y' in G. Therefore, $\{x, y\}$ is a 2-separator in G, a contradiction to the fact that G is 3-connected. $\square$

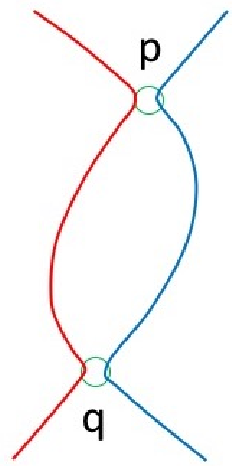
Observation 1.6.2. $\mathsf{OPT}_{\mathsf{cr}}(C^+) \leq O((|\chi| + 1) \cdot \mathrm{poly}(\Delta \log n))$.

Proof. Since every terminal $t \in \Gamma(C)$ belongs to exactly one set in $\{L(R) \mid R \in \mathcal{R}_G(C)\}$, we get that $|\mathcal{R}_G(R)| \leq |\Gamma(C)| \leq \Delta\mu \leq O(\mathrm{poly}(\Delta \log n))$. Recall that we have defined the extension $X_G(C)$ of cluster C, that is a collection of trees, that contains, for every bridge $R \in \mathcal{R}_G(C)$, a tree T_R, whose leaves are the vertices of $L(R)$, and whose inner vertices lie in R. For each such tree T_R, let T'_R be the tree obtained from T_R by suppressing all degree-2 vertices. Since $|L(R)| \leq |\Gamma(C)| \leq O(\mathrm{poly}(\Delta \log n))$, $|E(T'_R)| \leq O(\mathrm{poly}(\Delta \log n))$. Let C^1 be the graph obtained from the union of the cluster C, and all trees in $\left\{T'_R \mid R \in \mathcal{R}_G(C)\right\}$. Let $\tilde{E} = \bigcup_{T \in T'_R} E(T'_R)$. From the above discussion, $|\tilde{E}| \leq O(\mathrm{poly}(\Delta \log n))$.

Consider now the optimal drawing φ of G. We delete from this drawing the images of all edges and vertices, except those lying in C and in $\bigcup_{R \in \mathcal{R}_G(C)} T_R$. By suppressing all degree-2 vertices lying in trees in $\{T_R \mid R \in \mathcal{R}_G(C)\}$ (or, equivalently, by concatenating the images of the pair of edges incident to each such vertex), we obtain a drawing φ^1 of graph C^1. In this drawing, the total number of crossings in which edges of C participate is bounded by $|\chi|$, as in the drawing φ. However, images of edges in $\tilde{E}$ may cross each other many times. For each edge $e \in \tilde{E}$, we first modify its image so it does not cross itself, by removing self-loops as necessary. Additionally, as long as there is a pair $e, e' \in \tilde{E}$ of edges whose images cross more than once, we can modify the images of e and e' to reduce the number of crossings between them, by using a standard uncrossing operation (see Figure 1.1); this operation does not create any new additional crossings, and does not increase the number of crossings in which the edges of $E(C)$ participate. Let φ^1 be the final drawing of graph C^1 that we obtain at the end of this procedure. Then, since $|\tilde{E}| \leq O(\text{poly}(\Delta \log n))$, the number of crossings in φ^1 is bounded by $O(|\chi| + \text{poly}(\Delta \log n))$.



(a) Before: Image of edge e (red) and image of edge e' (blue) cross twice at p and q.



(b) After: New image of edge e (red) and new image of edge e' (blue) no longer cross at p or q.

Figure 1.1: An illustration of uncrossing of the images of a pair of edges that cross more than once.

We perform one additional modification to graph C^1, to obtain a multigraph C^2 as follows. Consider any bridge $R \in \mathcal{R}_G(C)$, and the corresponding tree T'_R. For every edge $e \in E(T'_R)$, we then create $2\Delta\mu$ parallel copies of the edge e. Let T''_R denote the resulting multi-graph that we obtain from tree T'_R. Once every bridge $R \in \mathcal{R}_G(C)$ is processed in this manner, we obtain the final graph C^2. We can obtain a drawing φ^2 of the graph C^2 from drawing φ^1 of C^1 in a natural way: for every tree T'_R, for every edge $e \in E(T'_R)$, we draw the $2\Delta\mu$ copies of the edge e in parallel to the drawing of e, very close to it. Since $\mu \leq \text{poly}(\Delta \log n)$, every crossing in drawing φ^1 may give rise to at most $\text{poly}(\Delta \log n)$ crossings in φ^2, so the total number of crossings in φ^2 is bounded by $O((|\chi| + 1)\,\text{poly}(\Delta \log n))$.

We are now ready to define the drawing of the graph C^+. We start with the drawing φ^2 of graph C^2, which already contains the images of the edges and the vertices of C. It now remains to add the images of the edges of $\tilde{E}$ to this drawing. In order to do so, we consider each bridge $R \in \mathcal{R}_G(C)$ one by one. Fix any such bridge $R \in \mathcal{R}_G(C)$, and consider the corresponding vertex set $L(R) = \left\{t_1(R), t_2(R), \ldots, t_{|L(R)|}(R)\right\}$. For all $1 \le i \le |L(R)|$, we let $Q_i(R)$ be a simple path in graph T''_R connecting $t_i(R)$ to $t_{i+1}(R)$, that corresponds to the unique path connecting $t_i(R)$ to $t_{i+1}(R)$ in the tree T'_R. Since graph C^2 contains $2\Delta\mu$ copies of every edge of T'_R, while $|L(R)| \le \Delta\mu$, we can ensure that the resulting paths $Q_1(R), \ldots, Q_{|L(R)|}(R)$ are mutually edge-disjoint. For all $1 \le i \le |L(R)|$, we let $\gamma_i(R)$ denote the curve obtained by concatenating the images of all edges lying on path $Q_i(R)$ in drawing φ^2. Intuitively, we would like to map, for all $1 \le i \le |L(R)|$, the edge $(t_i(R), t_{i+1}(R))$ to the curve $\gamma_i(R)$. One difficulty with this approach is that several curves in the set $\{\gamma_i(R)\}_{i=1}^{|L(R)|}$ may cross in a single point. This is because several paths in $\{Q_i(R)\}_{i=1}^{|L(R)|}$ may pass through a single vertex v. Obseve however that $|V(T'(R))| \le |L(R)| \le \mathrm{poly}(\Delta\mu)$. For each non-leaf vertex $v \in V(T'(R))$, consider a small disc $\eta(v)$ containing v in its interior, in the current drawing. We slightly modify all curves in $\{\gamma_i(R)\}_{i=1}^{|L(R)|}$ that contain the image of v inside the disc $\eta(v)$ to ensure that every pair of such curves cross at most once inside $\eta(v)$, and no point of $\eta(v)$ is contained in more than two curves. Since the total number of vertices in all graphs in $\left\{T'(R)\right\}_{R \in \mathcal{R}_G(C)}$ is bounded by $O(\mathrm{poly}(\Delta \log n))$, and since $|\tilde{E}| \le O(\mathrm{poly}(\Delta \log n))$, this modification introduces at most $O(\mathrm{poly}(\Delta \log n))$ additional crossings. We then obtain a valid drawing of graph C^+ with at most $O\left((|\chi| + 1)\,\mathrm{poly}(\Delta \log n)\right)$ crossings. $\square$

We now use the following theorem from [12]:

Theorem 1.6.3 (Theorem 8 in full version of [12])**.** *There is an efficient algorithm, that, given a 3-connected graph G with maximum vertex degree Δ and a planarizing set E' of its edges, computes a drawing ψ of G in the plane with $O\big((|E'|^2 + |E'| \cdot \mathsf{OPT}_{\mathsf{cr}}(G))\,\mathrm{poly}(\Delta)\big)$ crossings. Moreover, the drawing of $G \setminus E'$ induced by ψ is planar.*

We note that the statement of Theorem 8 in [12] does not include the claim that the drawing of $G \setminus E'$ induced by ψ is planar. However, it is easy to verify that their algorithm ensures this property, as the algorithm first selects a planar drawing of $G \setminus E'$ and then adds edges of E' to this drawing.

Observe that the set E' of edges is a planarizing set for C^+, as $C^+ \setminus E' = C$ is a planar graph. Therefore, we can use the algorithm from Theorem 1.6.3 to compute a drawing ψ of graph C^+, such that the drawing of C induced by ψ is planar. The number of crossings in ψ is bounded by:

$$O\left((|E'|^2 + |E'| \cdot \mathsf{OPT}_{\mathsf{cr}}(G)) \operatorname{poly}(\Delta)\right) \leq O\left((|\chi| + 1) \operatorname{poly}(\Delta \log n)\right).$$

Note that the drawing ψ of graph C^+ naturally induces a planar drawing ψ' of graph C. Since every terminal $t \in \Gamma(C)$ has degree 1 in C, for every terminal $t \in \Gamma(C)$, there is a unique face $F(t)$ in the drawing ψ', such that t lies on the (inner) boundary of $F(t)$. Unfortunately, it is possible that two terminals t, t' lie in the set $L(R)$ of legs for some bridge $R \in \mathcal{R}_G(C)$, but $F(t) \neq F(t')$. This situation is undesirable as it precludes us from defining the discs $D(R)$ for bridges $R \in \mathcal{R}_G(C)$ with the required properties. In order to overcome this difficulty, we start with $E^* = \emptyset$, and then gradually add edges of $E(C)$ to E^*. We will ensure that, throughout, graph $C \setminus E^*$ remains connected. Eventually, our goal is to ensure that, in the drawing of $C \setminus E^*$ induced by ψ', for every bridge $R \in \mathcal{R}_G(C)$, all vertices of $L(R)$ lie on the boundary of the single face. Once we achieve this, we will slightly modify the images of the edges incident to the terminals of C in a way that will allow us to define the desired set of discs $\{D(R)\}_{R \in \mathcal{R}_G(C)}$.

Before we proceed, we define the notion of distances between faces of the current drawing ψ'. Intuitively, the distance between a pair F, F' of faces in drawing ψ' is the smallest number of edges that need to be deleted from the current drawing, so that faces F and F' merge into a single face. Equivalently, it is the distance between F and F' in the dual graph. A third equivalent definition is the following: let $\gamma(F,F')$ be a curve originating at a point of F and terminating at a point of F' that intersects the current drawing ψ' at images of edges only (and avoids images of vertices). Among all such curves, choose the one minimizing the number of edges whose images it intersects. Denote by $E(F,F')$ the set of all edges that $\gamma(F,F')$ intersects. Then the distance between F and F' is $|E(F,F')|$. We need the following simple claim.

Claim 1.6.4. *For every bridge $R \in \mathcal{R}_G(C)$, and every index $1 \le i \le |L(R)|$, the distance between faces $F(t_i(R))$ and $F(t_{i+1}(R))$ is at most $O\left((|\chi|+1)\operatorname{poly}(\Delta \log n)\right)$.*

Proof. Recall that edge $(t_i(R), t_{i+1}(R))$ lies in graph C^+, and its image in ψ is a curve connecting a point in $F(t_i(R))$ to a point in $F(t_{i+1}(R))$, that intersects the image of C^+ at edges only. Since the total number of crossings in drawing ψ is at most $O\left((|\chi|+1)\operatorname{poly}(\Delta \log n)\right)$, we get that the distance between the two faces is also bounded by $O\left((|\chi|+1)\operatorname{poly}(\Delta \log n)\right)$. □

We now gradually modify the set E^* of edges, by processing the bridges $R \in \mathcal{R}_G(C)$ one by one. When a bridge $R \in \mathcal{R}_G(C)$ is processed, we consider each index $1 \le i \le |L(R)|$ in turn. We now describe an iteration where index i is processed. Let F and F' be the faces in the drawing of graph $C \setminus E^*$ induced by ψ', containing the images of terminals $t_i(R)$ and $t_{i+1}(R)$, respectively. If $F = F'$, then we do nothing. Otherwise, we are guaranteed that the distance between the two faces is at most $O\left((|\chi|+1)\operatorname{poly}(\Delta \log n)\right)$. Then there is a set $E^*_i(R)$ of at most $O\left((|\chi|+1)\operatorname{poly}(\Delta \log n)\right)$ edges of $C \setminus E^*$, such that, in the drawing of the graph $C \setminus (E^* \cup E^*_i(R))$ induced by ψ, the two faces are merged, and so both terminals $t_i(R)$ and $t_{i+1}(R)$ lie on the boundary of a single face. Moreover, it is easy to verify that, if graph $C \setminus E^*$ is connected, and $E^*_i(R)$ is a minimum-cardinality set of edges with the above properties, then graph $C \setminus (E^* \cup E^*_i(R))$ is also connected. We add the edges of $E^*_i(R)$ to E^*, and continue to the next iteration. Once every bridge R is processed, we are guaranteed that, in the drawing of $C \setminus E^*$ induced by ψ, for every bridge $R \in \mathcal{R}$, there is a single face

$F(R)$, whose boundary contains images of all vertices in $L(R)$. We are also guaranteed that graph $C \setminus E^*$ is connected. Since we add at most $O\left((|\chi|+1)\operatorname{poly}(\Delta \log n)\right)$ edges to E^* in each iteration, and the number of iterations is bounded by $|\Gamma(C)| \leq \operatorname{poly}(\Delta \log n)$, we get that at the end of the algorithm $|E^*| \leq O\left((|\chi|+1)\operatorname{poly}(\Delta \log n)\right)$.

Consider now the drawing of graph $C \setminus E^*$ induced by ψ. Recall that for every bridge $R \in \mathcal{R}_G(C)$, we have denoted by $F(R)$ the face in this drawing whose boundary contains all vertices of $L(R)$. We denote by $\mathcal{F} = \{F(R) \mid R \in \mathcal{R}_G(C)\}$. Next, we consider each face $F \in \mathcal{F}$ one by one. Let $\mathcal{R}(F) \subseteq \mathcal{R}_G(C)$ be the set of all bridges R with $F(R) = F$. We select one arbitrary disc D in the interior of the face F, and set, for every bridge $R \in \mathcal{R}(F)$, $D(F) = D$. For every terminal $t \in \bigcup_{R \in \mathcal{R}(F)} L(R)$, we consider the unique edge e_t that is incident to t in C. We extend the image of this edge, so that its endpoint t lies on disc D, but the interior of the edge remains disjoint from D. This is done by appending, to the current drawing of the edge e_t, a curve γ_t, connecting the image of t to a point on the disc D. The set $\Gamma(F) = \left\{\gamma_t \mid t \in \bigcup_{R \in \mathcal{R}(F)} L(R)\right\}$ of curves is defined as follows. Consider a terminal $t \in \bigcup_{R \in \mathcal{R}(F)} L(R)$. We start by letting γ_t be a curve connecting the image of t to a point on the boundary of the disc D, so that γ_t is disjoint from the images of the edges in $E(C) \setminus E^*$, and crosses the image of each edge in E^* at most once. It is easy to verify that such a curve can be constructed, for example, by following the curves $\gamma(F, F')$ that we used in order to merge pairs of faces by adding edges to set E^*. Next, we use standard uncrossing procedure to ensure that the resulting curves in $\Gamma(F) = \left\{\gamma_t \mid t \in \bigcup_{R \in \mathcal{R}(F)} L(R)\right\}$ do not cross each other. This step only modifies the curves in $\Gamma(F)$ and does not introduce any new crossings.

Once we process every face $F \in \mathcal{F}$, we obtain the final drawing ψ^* of the cluster C. Notice that the only difference between ψ^* and the planar drawing of C induced by ψ is that we have modified the images of the edges in set $\{e_t \mid t \in \Gamma(C)\}$, by appending a curve γ_t to the image of each such edge e_t. The total number of crossings in ψ^* is bounded by $|E^*| \cdot |\Gamma(C)| \leq O\left((|\chi|+1)\operatorname{poly}(\Delta \log n)\right)$. From our construction, we also guarantee that graph $C \setminus E^*$ is connected, and that the drawing of $C \setminus E^*$ induced by ψ^* is planar. Additionally, for every bridge $R \in \mathcal{R}_G(C)$, we have defined a disc $D(R)$, such that the images of all vertices in $L(R)$ are drawn on the boundary of $D(R)$, the interior of $D(R)$ is disjoint from the drawing ψ^* of C, and the image of every edge of C is disjoint from $D(R)$, except possibly for its endpoint that lies on $D(R)$. For every pair $R \neq R'$ of bridges, either $D(R) = D(R')$, or $D(R) \cap D(R') = \emptyset$ must hold. This completes the proof of Theorem 1.3.3.

1.7 Obtaining a Canonical Drawing: Proof of Theorem 1.3.4

In this section we provide the proof of Theorem 1.3.4. For brevity, we will refer to type-1 and type-2 acceptable clusters of $\mathcal{C}_1 \cup \mathcal{C}_2$ as type-1 and type-2 clusters, respectively. Throughout this section, we assume that we are given a n-vertex graph G with maximum vertex degree at most Δ, and a decomposition $\mathcal{D} = \left(E'', A, \mathcal{C}_1, \mathcal{C}_2, \{\psi_C\}_{C \in \mathcal{C}_2}, \mathcal{P}(A)\right)$ of G into acceptable clusters. Recall that E'' is a planarizing set of edges for G; the endpoints of the edges in E'' are called terminals, and we denote the set of all terminals by Γ. Set A contains fake edges, whose endpoints must be in Γ. The set of all connected components of $(G \setminus E'') \cup A$ is $\mathcal{C}_1 \cup \mathcal{C}_2$, and we refer to the elements of $\mathcal{C}_1 \cup \mathcal{C}_2$ as clusters. Additionally, we are given, for each cluster $C \in \mathcal{C}_1 \cup \mathcal{C}_2$, a drawing ψ_C of C on the sphere, and, for every bridge $R \in \mathcal{R}_G(C \setminus A_C)$, a disc $D(R)$ on the sphere. We are guaranteed that every cluster in $\mathcal{C}_1$ is a type-1 cluster, and every cluster in $\mathcal{C}_2$ is a type-2 cluster with respect to the drawing ψ_C. Lastly, the set $\mathcal{P}(A)$ of paths defines a legal embedding of the fake edges. We also assume that we are given some drawing φ of G, and our goal is to transform this drawing, so that it becomes canonical with respect to all clusters in $\mathcal{C}_1 \cup \mathcal{C}_2$, such that the total number of crossings only increases slightly. At a high level, the algorithm processes all clusters in $\mathcal{C}_1 \cup \mathcal{C}_2$ one-by-one. In each iteration, we will modify the current drawing of G so that it will become canonical with respect to the cluster C that is processed in that iteration. The main tool that we use in order to iteratively modify the graph is procedure ProcDraw that is described in Section 1.7.3. Before we describe the procedure, we need two additional tools: the notion of non-interfering paths, and the notion of irregular vertices and edges that was introduced in [12]. We describe these two tools in Sections 1.7.1 and 1.7.2 respectively.

1.7.1 Non-Interfering Paths

In this subsection we define the notion of non-interfering paths and prove a lemma that allows us to find such paths. The notion of the non-interfering paths is defined with respect to any given graph H', and will eventually be applied to various subgraphs of $G \cup A$.

We assume that we are given any graph H' and a drawing ψ of H' on the sphere. For every vertex $v \in V(H')$, we let $\eta(v)$ be a small closed disc that contains v in its interior. In particular, no image of the vertices of $V(H') \setminus \{v\}$ appears in the disc; if $\psi(e) \cap \eta(v) \neq \emptyset$ for any edge e, then e must be incident to the vertex v, and $\psi(e) \cap \eta(v)$ must be a simple curve (a curve that does not cross itself) that intersects the boundary of $\eta(v)$ at exactly one point. The discs $\eta(v)$ that correspond to distinct vertices must be disjoint.

We now fix some vertex $v \in V(H')$. Let $\delta(v) = \{e_1, \ldots, e_r\}$ be the set of all edges that are incident to vertex v in H'. For each such edge e_i, let p_i be the unique point on the image of edge e_i that lies on the boundary of the disc $\eta(v)$. Notice that the circular ordering of the points $p_1, \ldots, p_r$ on the boundary of $\eta(v)$ defines a circular ordering $\tilde{\mathcal{O}}(v)$ of the edges in $\delta(v)$. We call this ordering *the ordering of the edges of $\delta(v)$ in ψ, as they enter vertex v*. For each edge $e_i \in \delta(v)$, we let $\sigma_{e_i}(v)$ be a small closed segment of the boundary of the disc $\eta(v)$, that contains the point p_i in its interior, such that all segments in $\{\sigma_{e_i}(v)\}_{i=1}^r$ are mutually disjoint (see Figure 1.2). Next, we define the notion of a thin strip around a path in H'.

Thin strip around a path. Let P be any path in H'. We denote the endpoints of P by u and v, we denote by e_v the unique edge on path P that is incident to vertex v, and we

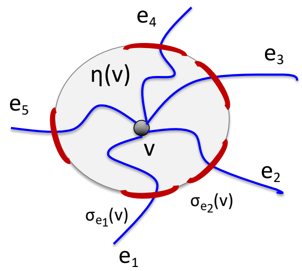


Figure 1.2: Disc $\eta(v)$ and segments $\sigma_{e_i}(v)$.

denote by e_u the unique edge of P that is incident to vertex u. Recall that the image $\psi(P)$ of path P in ψ is a curve obtained by concatenating the images of its edges in ψ. We define a *thin strip* S_P around the image of P in ψ, by adding two curves, γ_L and γ_R, immediately to the left and to the right of the image of P respectively, that follow the image of P. The two curves originate at the two endpoints of the segment $\sigma_{e_v}(v)$, and terminate at the two endpoints of the segment $\sigma_{e_u}(u)$. They do not cross each other except when the image of P crosses itself, and do not intersect the interiors of the discs $\eta(v)$ and $\eta(u)$. The two curves are extremely close to the image of P in ψ. The region of the sphere, whose boundary is the concatenation of $\sigma_{e_v}(v), \gamma_L, \sigma_{e_u}(u)$ and γ_R, and that contains the image of P (except for $\psi(P) \cap \eta(u)$ and $\psi(P) \cap \eta(v)$), defines the thin strip S_P around P. We draw the curves γ_L and γ_R so that the resulting strip S_P contains images of vertices of $P \setminus \{u, v\}$, and no other vertices of H'. Additionally, the only edges of G whose images have a non-empty intersection with S_P are (i) edges that are incident to vertices of $P \setminus \{u, v\}$; and (ii) edges whose images cross the edges of P; see Figure 1.3 for an illustration. For each such edge e, $\psi(e) \cap S_P$ is a collection of disjoint open curves, where each curve contains a point that belongs to the image of P (the corresponding point is either an image of a vertex of P, or a crossing point of e with an edge of P). We can similarly define a thin strip $S_{e'}$ around the image of an edge e' in ψ, by considering a path that only consists of the edge e'.

Definition 12 (Non-interfering Paths)**.** *Let H' be a planar graph and let ψ be a planar drawing of H'. Let $\mathcal{P}$ be a set of paths in H', where for each path $P \in \mathcal{P}$, we denote the endpoints of P by u_P and v_P. We say the paths of $\mathcal{P}$ are* non-interfering with respect to ψ

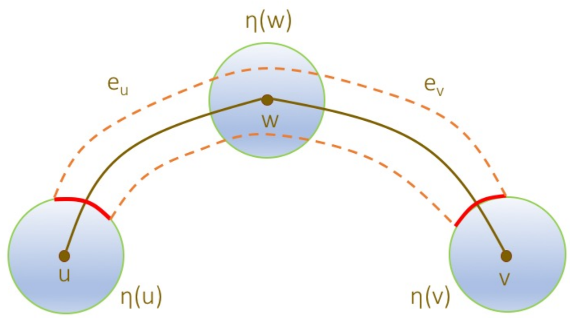


Figure 1.3: A thin strip S_P around path $P = (u, w, v)$. The segments $\sigma_{e_v}(v)$ and $\sigma_{e_u}(u)$ are shown in red and the curves γ_L and γ_R are dashed orange lines.

(see Figure 1.4), iff there exists a collection $\{\gamma_P\}_{P\in\mathcal{P}}$ of curves, such that:

1. *for each path $P \in \mathcal{P}$, the curve γ_P connects $\psi(u_P)$ to $\psi(v_P)$, and is contained in $\eta(u_P) \cup \eta(v_p) \cup S_P$, where S_P is the thin strip around P in ψ; and*

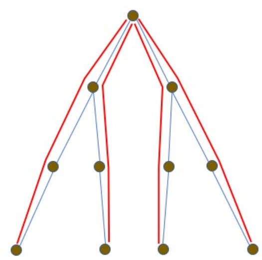
2. *for every pair $P, P' \in \mathcal{P}$ of distinct paths, the curves γ_P and $\gamma_{P'}$ are disjoint.*

The set $\{\gamma_P\}_{P\in\mathcal{P}}$ of curves with the above properties is called a non-interfering representation *of $\mathcal{P}$ with respect to ψ.*

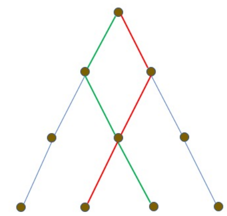
Note that the curve γ_P in the above definition may cross $\psi(P)$ (the image of the path P in drawing ψ) arbitrarily many times. Also note that, as shown in Figure 1.4, non-interfering paths may share vertices and edges. We also use the following two definitions.

Definition 13. *Given a graph H', a set Γ of its vertices, together with another vertex u (that may belong to Γ), a* routing of the vertices of Γ to u *is a set $\mathcal{Q} = \{Q_v \mid v \in \Gamma\}$ of paths, where for each vertex $v \in \Gamma$, path Q_v connects v to u. We sometimes say that set $\mathcal{Q}$ of paths* routes vertices of Γ to u.

Definition 14. *Given a set $\mathcal{Q}$ of paths in a graph H', for each edge $e \in E(H')$, we denote by $\mathrm{cong}_{\mathcal{Q}}(e)$ the* congestion *of the paths in $\mathcal{Q}$ on edge e – the number of paths in $\mathcal{Q}$ that contain e. We denote by $\mathrm{cong}_{H'}(\mathcal{Q}) = \max_{e\in E(H')}\left\{\mathrm{cong}_{\mathcal{Q}}(e)\right\}$ the total congestion caused by the set $\mathcal{Q}$ of paths in H'.*



(a) In this figure we consider a collection of paths connecting every leaf of the tree to its root. These paths are non-interfering, and the curves that are shown in red are their non-interfering representation with respect to this drawing of the tree.



(b) The red path and the green path in this figure are not non-interfering with respect to this drawing.

Figure 1.4: Non-interfering paths and non-interfering representations.

Assume now that we are given a set Γ of vertices of H', a vertex $u \in V(H')$, and a routing $\mathcal{Q}$ of the vertices of Γ to u in H'. Let $\mathcal{O}$ be any ordering of the vertices of H'. We say that $\mathcal{O}$ is *consistent* with the set $\mathcal{Q}$ of paths if, for every path $Q_v \in \mathcal{Q}$, for every pair x, y of distinct vertices of Q, where y lies closer to u than x on Q, vertex x appears before vertex y in $\mathcal{O}$. The following lemma allows us to transform any routing of a set Γ of vertices to a given vertex u into a collection of non-interfering paths, and to compute an ordering of vertices of H' that is consistent with the resulting set of paths. The sets of paths produced by this lemma will be used as guiding paths by procedure ProcDraw in order to modify the drawing of G. The proof of the lemma is deferred to Section 1.9.5.

Lemma 1.7.1. *There is an efficient algorithm, that, given a planar graph H' with a planar drawing ψ of H', a collection Γ of vertices of H' and another vertex $u \in V(H')$ (where possibly $u \in \Gamma$), together with a set $\mathcal{Q}$ of paths routing Γ to u in H', computes another set $\mathcal{Q}'$ of paths routing Γ to u, such that the set $\mathcal{Q}'$ of paths is non-interfering with respect to ψ, and for every edge $e \in E(H')$, $\mathrm{cong}_{\mathcal{Q}'}(e) \leq \mathrm{cong}_{\mathcal{Q}}(e)$. Additionally, the algorithm computes a non-interfering representation $\{\gamma_Q \mid Q \in \mathcal{Q}'\}$ of $\mathcal{Q}'$ and an ordering $\mathcal{O}$ of the vertices of H' that is consistent with the set $\mathcal{Q}'$ of paths.*

Consider the set $\mathcal{Q}'$ of paths given by Lemma 1.7.1. Even though the paths in $\mathcal{Q}'$ are undirected, it may be convenient to think of them as being directed towards u. Let $e = (x, y) \in E(H')$ be an edge, and assume that x appears before y in the ordering $\mathcal{O}$. Let $\mathcal{P}(e) \subseteq \mathcal{Q}'$ be the subset of all paths Q with $e \in E(Q)$. Notice that all paths in $\mathcal{P}(e)$ must traverse the edge e in the direction from x to y, since the ordering $\mathcal{O}$ of $V(H')$ is consistent with respect to $\mathcal{Q}'$. Moreover, if we consider the intersections of the curves $\{\gamma_Q \mid Q \in \mathcal{P}(e)\}$ with the thin strip S_e around the image of the edge $e = (u, v)$ in ψ, then the order in which these curves traverse S_e (e.g. the order in which they intersect the segment $\sigma_e(u)$) naturally defines an ordering of the paths in $\mathcal{P}(e)$. We denote this ordering by $\tilde{\mathcal{O}}(e)$, and we refer to it as the *ordering of the paths in* $\mathcal{P}(e)$ *in the strip* S_e; see Figure 1.5 for an illustration.

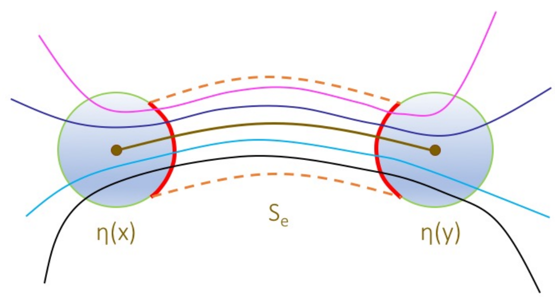


Figure 1.5: The ordering $\tilde{O}(e)$ of the paths in $\mathcal{Q}'_e$ in S_e.

1.7.2 Irregular Vertices and Edges

In this subsection, we provide the definitions of irregular vertices and edges from [12], and then state a lemma from [12] about them. Let $H' = (V, E)$ be a connected graph and let φ and ψ be a pair of drawings of H' in the plane.

As before, we denote by $S_2(H')$ the set of all vertices that participate in 2-separators in H', that is, vertex $v \in S_2(H')$ iff there is another vertex $u \in V$, such that graph $H' \setminus \{u, v\}$ is not connected. We denote by $E_2(H')$ the set of all edges that have both endpoints in set $S_2(H')$.

Definition 15 (Irregular Vertices)**.** *We say that a vertex v of H' is* irregular *(with respect to*

φ and ψ) iff (i) its degree in H' is greater than 2; and (ii) the circular ordering of the edges incident on it, as their images enter v, is different in φ and ψ (ignoring the orientation).

We denote the set of all vertices that are irregular with respect to φ and ψ by $\mathsf{IRG}_V(\varphi, \psi)$, and we call all other vertices *regular*.

Definition 16 (Irregular Paths and Edges)**.** *We say that a path P with endpoints x and y in H' is* irregular *iff x and y both have degree at least 3 in H', all other vertices of P have degree 2 in H', vertices x and y are regular, but their orientations differ in φ and ψ. In other words, the orderings of the edges adjacent to x and to y are identical in both drawings, but the pairwise orientations are different: for one of the two vertices, the orientations are identical in both drawings (say clock-wise), while for the other vertex, the orientations are opposite (one is clock-wise, and the other is counter-clock-wise). An edge e is an* irregular edge *with respect to φ and ψ iff it is the first or the last edge on an irregular path. In particular, if the irregular path only consists of a single edge e, then e is an irregular edge.*

We denote the set of all edges that are irregular with respect to φ and ψ by $\mathsf{IRG}_E(\varphi, \psi)$, and we call all other edges regular. The following lemma is a re-statement of Lemma 2 from Section B from the arxiv version of [12] for the special case where the graph H' is 2-connected.

Lemma 1.7.2. *([12]) Let H' be a 2-connected planar graph, let φ be an arbitrary drawing of H' in the plane, and let ρ be a planar drawing of H'. Then*

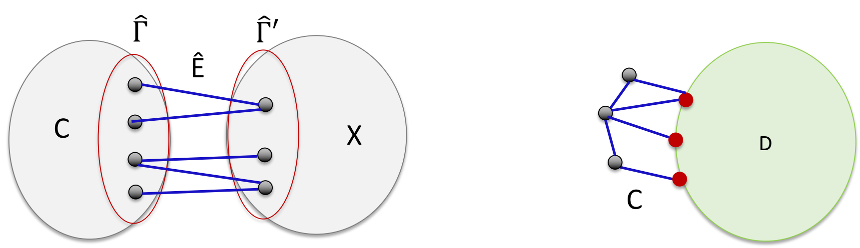
$$|\mathsf{IRG}_V(\varphi, \rho) \setminus S_2(H')| + |\mathsf{IRG}_E(\varphi, \rho) \setminus E_2(H')| \le O(\mathrm{cr}(\varphi)).$$

1.7.3 Main Subroutine: Procedure ProcDraw

In this subsection we describe and analyze procedure ProcDraw, that is central to the proof of Theorem 1.3.4. We note that a similar procedure was introduced in [9] (see Section D of the full version). The procedure will be applied repeatedly to every cluster $C \in \mathcal{C}_1 \cup \mathcal{C}_2$ (and more precisely, to several faces in the drawing ψ_C of the cluster C), with the goal of transforming the current drawing of the graph G into a drawing that is canonical with respect to C.

Intuitively, the input to the procedure consists of two disjoint graphs: graph C (that we can think of as a cluster of $\mathcal{C}_1 \cup \mathcal{C}_2$), and graph X (that we can think of, somewhat imprecisely, as the rest of the graph G, or as some bridge in $\mathcal{R}_G(C)$). Additionally, we are given a set $\hat{E}$ of edges that connect some vertices of X to some vertices of C. We denote by $\hat{\Gamma}$ and by $\hat{\Gamma}'$ the sets of endpoints of the edges in $\hat{E}$ lying in C and X, respectively. Abusing the notation, in this subsection, we denote by G the graph that is the union of X, C, and the set $\hat{E}$ of edges (see Figure 1.6(a)).

We assume that we are given some drawing φ of G on the sphere (which represents, intuitively, the current drawing of the whole graph G), and another drawing ψ of C on the sphere (which will eventually be the canonical drawing ψ_C of C). Furthermore, we are given a closed disc D on the sphere, such that, in the drawing ψ, the images of all edges of C are internally disjoint from D, and the images of all vertices of C are disjoint from D, except that the images of vertices of $\hat{\Gamma}$ lie on the boundary of D (see Figure 1.6(b)). This disc D will correspond to the discs $D(R)$ that we have defined for the various bridges $R \in \mathcal{R}_G(C)$.



(a) Schematic view of graph G.

(b) Drawing ψ of C and the disc D. The vertices of Γ are shown in red.

Figure 1.6: Input to ProcDraw

Lastly, we are given some vertex $u^* \in V(C)$, and a set $\mathcal{Q} = \left\{ Q_v \mid v \in \hat{\Gamma} \right\}$ of paths in C that route $\hat{\Gamma}$ to u^*. Intuitively, we will use the images of the paths in $\mathcal{Q}$ (after slightly modifying them) in the drawing φ as guiding lines in order to modify the drawing φ. We let $J \subseteq G$ be the graph containing all vertices and edges that participate in the paths in $\mathcal{Q}$, and we assume that the drawing of J induced by ψ is planar.

The goal of ProcDraw is to compute a new drawing φ' of G, such that the drawing of C induced by φ' is identical to ψ (but the orientation may be arbitrary), and all vertices and edges of X are drawn in the interior of the disc D, that is defined with respect to ψ. We would also like to ensure that the number of crossings in φ' is not much larger than the number of crossings in φ. We now formally summarize the input and the output of the procedure ProcDraw.

Input. The input to procedure ProcDraw consists of:

- Two disjoint graphs C, X, subsets $\hat{\Gamma} \subseteq V(C)$, $\hat{\Gamma}' \subseteq V(X)$ of vertices, and a set $\hat{E}$ of edges that connect vertices of $\hat{\Gamma}$ to vertices of $\hat{\Gamma}'$, such that every vertex in $\hat{\Gamma} \cup \hat{\Gamma}'$ is an endpoint of at least one edge of $\hat{E}$ (see Figure 1.6(a)). We denote $G = C \cup X \cup \hat{E}$, and we denote the maximum vertex degree in G by Δ;
- A vertex $u^* \in V(C)$, and a set $\mathcal{Q} = \left\{ Q_v \mid v \in \hat{\Gamma} \right\}$ of paths in C that route $\hat{\Gamma}$ to u^*. We refer to the paths in $\mathcal{Q}$ as the *guiding paths* for the procedure, and we let $J \subseteq G$ be the graph containing all vertices and edges that participate in the paths in $\mathcal{Q}$;
- An arbitrary drawing φ of graph G on the sphere; and
- A drawing ψ of graph C on the sphere, such that the drawing of J induced by ψ is planar, and additionally a closed disc D on the sphere, such that, in drawing ψ, the images of vertices of $V(C)$ are disjoint from the disc D, except for the vertices of $\hat{\Gamma}$ whose images lie on the boundary of D, and the images of the edges of $E(C)$ are disjoint from the disc D, except for their endpoints that belong to $\hat{\Gamma}$.

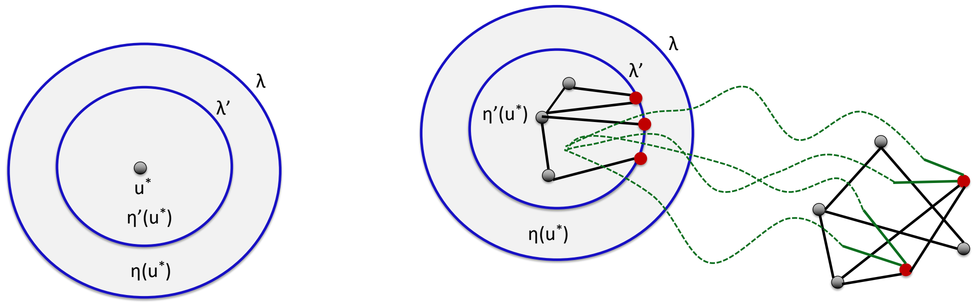
Output. The output of the procedure ProcDraw is a drawing φ' of G on the sphere, that has the following properties:

- The drawing of C induced by φ' is identical to ψ (but the orientation may be different);
- All vertices and edges of X are drawn in the interior of the disc D in φ' (the disc D is defined with respect to ψ); and
- The edges of $\hat{E}$ are drawn inside the disc D, and they only intersect the boundary of D at their endpoints that belong to $\hat{\Gamma}$.

We now describe the execution of the procedure ProcDraw. We start from the drawing φ of the graph G, and then modify it to obtain the desired drawing φ'. For simplicity of exposition, in the remainder of this subsection, we use the following notation. For any drawing $\hat{\varphi}$ of any graph $\hat{G}$, and for any subgraph $\hat{H} \subseteq \hat{G}$, we denote by $\hat{\varphi}_{\hat{H}}$ the unique drawing of $\hat{H}$ induced by the drawing $\hat{\varphi}$ of $\hat{G}$. The procedure consists of two steps.

Step 1. In this step, we consider the drawing φ of G on the sphere, and the disc $\eta(u^*)$ around the vertex u^*. We denote the boundary of this disc by λ, and we let D' be the disc whose boundary is λ, that is disjoint from $\eta(u^*)$ except for sharing the boundary with it. By shrinking the disc $\eta(u^*)$ a little, we obtain another disc $\eta'(u^*) \subseteq \eta(u^*)$, whose boundary is denoted by λ', such that λ' is disjoint from λ (see Figure 1.7(a)).

We then erase, from φ, the images of the vertices and the edges of C (but we keep the images of the edges of $\hat{E}$), and instead place the drawing ψ of C inside the disc $\eta'(u^*)$. Recall that we are given a disc D in the drawing ψ of C, such that the images of the vertices of C are disjoint from the disc D, except that the images of the vertices in $\hat{\Gamma}$, that lie on the boundary of D, and the images of the edges of C are disjoint from the disc D, except for their endpoints that belong to $\hat{\Gamma}$ and lie on the boundary of D. We plant the drawing ψ inside the disc $\eta'(u^*)$ in such a way that the boundary of the disc D coincides with λ' (see



(a) Discs $\eta(u^*)$ and $\eta'(u^*)$, and their boundaries λ and λ', respectively.

(b) After Step 1. Graph C is now drawn inside disc $\eta'(u^*)$, using the drawing ψ, with the vertices of $\hat{\Gamma}$ (shown in red) appearing on λ'. Graph X is drawn outside $\eta(u^*)$, preserving the original drawing φ. The vertices of $\hat{\Gamma}'$ are shown in red, and the images of the edges of $\hat{E}$ in φ are shown in green. We also show, in dashed green curves, the images of the paths in $\mathcal{Q}$ in the original drawing φ.

Figure 1.7: Illustration for Step 1 of ProcDraw.

Figure 1.7(b)). Therefore, all vertices of $\hat{\Gamma}$ are now drawn on the curve λ', and the image of C now appears inside the disc $\eta'(u^*)$. Note that the drawing of $X \cup \hat{E}$ remains unchanged and its image still lies in the interior of the disc D'. We denote the drawing that we obtained after the first step by $\hat{\varphi}$. In order to obtain the final drawing of the graph G, we need to extend the drawings (in $\hat{\varphi}$) of the edges in $\hat{E}$, so that they connect the original images of the vertices in $\hat{\Gamma}'$ to the new images of the vertices in $\hat{\Gamma}$.

Step 2. The goal of this step is to extend the images of the edges $e \in \hat{E}$ in the current drawing, so that they terminate at the new images of the vertices of $\hat{\Gamma}$. We do so by exploiting the images of the paths in $\mathcal{Q}$ in the original drawing φ of G, after slightly modifying them. Specifically, consider the set $\mathcal{Q} = \left\{Q_v \mid v \in \hat{\Gamma}\right\}$ of paths. Recall that $J \subseteq G$ is the graph containing all vertices and edges that participate in the paths in $\mathcal{Q}$. We say that a vertex $v \in V(J)$ is *irregular* if it is irregular with respect to the drawings φ_J and ψ_J of J (that are induced by the drawings φ of G and ψ of C, respectively, where φ is the original drawing of the graph G). We define irregular edges and paths in graph J similarly.

Recall that for every edge $e' \in E(C)$, we have denoted by $\mathrm{cong}_{\mathcal{Q}}(e')$ the congestion of the set

$\mathcal{Q}$ of edges on e' – that is, the total number of paths in $\mathcal{Q}$ that contain the edge e'. Consider now some edge $e = (u, v) \in \hat{E}$, with $u \in \hat{\Gamma}$ and $v \in \hat{\Gamma}'$. We subdivide the edge e with a new vertex t_e, and we denote by $\Gamma^* = \left\{ t_e \mid e \in \hat{E} \right\}$ this new set of vertices. We let set $\hat{E}'$ of edges contain, for each edge $e = (u, v) \in \hat{E}$ with $u \in \hat{\Gamma}$, the edge (u, t_e). Consider the current drawing $\hat{\varphi}$. Once we subdivide each edge $e \in \hat{E}$ with the vertex t_e, this new drawing (that we still denote by $\hat{\varphi}$) now contains the images of the edges in $\hat{E}'$. Similarly, we add the vertices of Γ^* to the original drawing φ of G, and we still denote this new drawing by φ. We denote by $\varphi(C)$ the drawing of C induced by φ, and we similarly denote by $\varphi(C \cup \hat{E}')$ the drawing of $C \cup \hat{E}'$ induced by φ. Lastly, we denote by J' the graph $J \cup \hat{E}'$.

Notice that graph J' is planar, since J is planar. We let $\psi'_{J'}$ be a drawing of J', obtained from the drawing ψ_J of J induced by ψ, by adding the drawings of the edges in $\hat{E}'$ to it, without introducing any new crossings (recall that each such edge connects a vertex of $\hat{\Gamma}$ to a vertex of Γ^*, and that the latter has degree 1 in J'). Consider any edge $(u, t_e) \in \hat{E}'$. If vertex u is a regular vertex (recall that this is defined with respect to the drawings φ_J and ψ_J of J), then we add the drawing of the edge (u, t_e) to ψ so that u remains a regular vertex with respect to the drawings $\varphi_{J'}$ and $\psi'_{J'}$ of J'. In other words, the drawing of the edges in $\hat{E}'$ is added to ψ_J in such a way that every vertex $v \in \hat{\Gamma}$ that was regular with respect to ψ_J and φ_J, remains regular with respect to $\psi'_{J'}$ and $\varphi_{J'}$. Similarly, we can ensure that every edge of $E(J)$ that was regular with respect to ψ_J and φ_J remains regular with respect to $\psi'_{J'}$ and $\varphi_{J'}$.

Consider now some edge $e = (u, v) \in \hat{E}$, with $u \in \hat{\Gamma}$. We denote the corresponding new edge $(u, t_e) \in \hat{E}'$ by $\hat{e}$. By concatenating the path $Q_u \in \mathcal{Q}$ with the edge (u, t_e), we obtain a new path, that we denote by Q'_e, connecting t_e to u^*, such that every vertex of Q'_e except for t_e lies in J. Let $\mathcal{Q}' = \left\{ Q'_e \mid e \in \hat{E} \right\}$ be the resulting set of paths. It is easy to verify that, for every edge $\hat{e} \in \hat{E}'$, $\mathrm{cong}_{\mathcal{Q}'}(\hat{e}) = 1$, and for every edge $e' \in E(J)$, $\mathrm{cong}_{\mathcal{Q}'}(e') \leq \Delta \cdot \mathrm{cong}_{\mathcal{Q}}(e')$.

Next, we apply Lemma 1.7.1 to graph J', its planar drawing $\psi'_{J'}$, and the set $\mathcal{Q}'$ of paths, to obtain a new set $\mathcal{Q}''$ of paths, routing Γ^* to u^* in J', that are non-interfering with respect to $\psi'_{J'}$. The lemma ensures that, for every edge $\hat{e} \in \hat{E}'$, $\mathrm{cong}_{\mathcal{Q}''}(\hat{e}) = 1$, and for every edge $e' \in E(J)$, $\mathrm{cong}_{\mathcal{Q}''}(e') \leq \mathrm{cong}_{\mathcal{Q}'}(e') \leq \Delta \cdot \mathrm{cong}_{\mathcal{Q}}(e')$. We denote, for each edge $e \in \hat{E}$, by $Q''_e \in \mathcal{Q}''$ the unique path originating at the vertex t_e. Additionally, the lemma provides a non-interfering representation $\left\{\gamma_{Q''_e} \mid e \in \hat{E}\right\}$ of the paths in $\mathcal{Q}''$, and an ordering $\mathcal{O}$ of the vertices of $V(J) \cup \Gamma^*$ that is consistent with the paths in $\mathcal{Q}''$. Notice that every vertex $t_e \in \Gamma^*$ has degree 1 in $C \cup \hat{E}'$, so the first edge on path Q''_e must be the edge $\hat{e} \in \hat{E}'$.

Consider some edge $e = (u, u') \in \hat{E}$, with $u \in \hat{\Gamma}$. We will define a curve ζ_e that connects the image of u in the current drawing $\hat{\varphi}$ to the image of u', and is contained in the thin strip $S_{Q''_e}$ around the drawing of path Q''_e in φ, as follows. Denote $Q''_e = (t_e = v_1, v_2, \ldots, v_r = u^*)$. For all $1 \leq i < r$, we denote by $e_i = (v_i, v_{i+1})$ the ith edge on this path. In order to define the curve ζ_e, we will define, for every edge $e_i \in E(Q''_e)$ with $i > 1$, a curve $\zeta_e(e_i)$, that is contained in the thin strip S_{e_i} around the image of e_i in the original drawing φ of G, and connects some point $p'_e(v_i)$ on the boundary of the disc $\eta(v_i)$ to some point $p_e(v_{i+1})$ on the boundary of the disc $\eta(v_{i+1})$. We also define a curve $\zeta_e(e_1)$, connecting the image of vertex $v_1 = t_e$ to some point $p_e(v_2)$ on the boundary of the disc $\eta(v_2)$. Additionally, for all $2 \leq i < r$, we define a curve $\zeta_e(v_i)$, that is contained in $\eta(v_i)$, and connects the point $p_e(v_i)$ to the point $p'_e(v_i)$. Lastly, we define a curve $\zeta_e(v_r)$, that is contained in $\eta(u^*) \setminus \eta'(u^*)$, and connects the point $p_e(v_r)$ that lies on λ to the image of the vertex $u \in \hat{\Gamma}$, that lies on λ'. The final drawing of the edge $e = (u, u')$ is obtained by concatenating the image of the edge (u', t_e) in the current drawing $\hat{\varphi}$, and the curves $\zeta_e(e_1), \zeta_e(v_2), \zeta_e(e_2), \ldots, \zeta_e(e_r), \zeta_e(v_r)$. The resulting curve connects the image of the vertex u' to the image of the vertex u, as required. It now remains to define each of these curves.

Drawing around the vertices. Consider some vertex $v \in V(J)$. Let $\mathcal{P}(v) \subseteq \mathcal{Q}''$ be the set of all paths $Q''_e \in \mathcal{Q}''$ that contain the vertex v. We assume first that $v \neq u^*$. For each path $Q''_e \in \mathcal{P}(v)$, consider the corresponding curve $\gamma_{Q''_e}$ that was defined as part of the non-interfering representation of the paths in $\mathcal{Q}''$ in the drawing $\psi'_{J'}$. We think of this curve as being directed towards the vertex u^*. Note that the curve $\gamma_{Q''_e}$ intersects the boundary of $\eta(v)$ in ψ in exactly two points; we denote the first point by $q_e(v)$ and the second point by $q'_e(v)$. If we denote by e_i, e_{i+1} the edges of Q''_e that appear immediately before and immediately after v on path Q''_e, then point $q_e(v)$ must lie on the segment $\sigma_{e_i}(v)$, and point $q'_e(v)$ must lie on the segment $\sigma_{e_{i+1}}(v)$ of the boundary of $\eta(v)$ in the drawing ψ (see Figure 1.2).

Assume first that vertex v is a regular vertex with respect to the drawings $\psi'_{J'}$ and $\varphi_{J'}$. Then the set $\left\{q_e(v), q'_e(v)\right\}_{Q''_e \in \mathcal{P}(v)}$ of points on the boundary of $\eta(v)$ in the drawing ψ naturally defines the corresponding set $\left\{p_e(v), p'_e(v)\right\}_{Q''_e \in \mathcal{P}(v)}$ of points on the boundary of $\eta(v)$ in the drawing φ (if the orientation of the vertex v is different in the two drawings, then we flip the sets of points accordingly). Moreover, for every path $Q''_e \in \mathcal{P}(v)$, the intersection of the curve $\gamma_{Q''_e}$ with the disc $\eta(v)$ in the drawing $\psi'_{J'}$ naturally defines a curve $\zeta_e(v)$ in the drawing φ, that is contained in the disc $\eta(v)$, and connects point $p_e(v)$ to point $p'_e(v)$. Notice that the resulting curves in $\{\zeta_e(v)\}_{Q''_e \in \mathcal{P}(v)}$ are all mutually disjoint.

Assume now that vertex v is irregular with respect to $\psi'_{J'}$ and $\varphi_{J'}$. Consider any path $Q''_e \in \mathcal{P}(v)$, and let e_i, e_{i+1} be the edges of Q''_e that appear immediately before and immediately after v on path Q''_e. In this case, we let $p_e(v)$ be a point on the segment $\sigma_{e_i}(v)$ of the boundary of the disc $\eta(v)$ in φ, and similarly we let $p'_e(v)$ be a point on the segment $\sigma_{e_{i+1}}(v)$ of the boundary of the disc $\eta(v)$ in φ. We ensure that all points that are added to each segment $\sigma_{e'}(v)$, for all $e' \in \delta(v)$ are distinct, and their ordering within each segment $\sigma_{e'}(v)$ is the same as the ordering of the corresponding points of $\left\{q_e(v), q'_e(v)\right\}_{Q''_e \in \mathcal{P}(v)} \cap \sigma_{e'}(v)$ in $\psi'_{J'}$. For every path $Q''_e \in \mathcal{P}(v)$, we let $\zeta_e(v)$ be an arbitrary curve in φ, that is contained in the disc $\eta(v)$, and connects point $p_e(v)$ to point $p'_e(v)$; we ensure that every pair of curves in $\{\zeta_e(v)\}_{Q''_e \in \mathcal{P}(v)}$ intersect at most once.

Lastly, we consider the case where $v = u^*$. In this case, for each path $Q''_e \in \mathcal{Q}''$, the intersection of the curve $\gamma_{Q''_e}$ with the boundary of the disc $\eta(u^*)$ in ψ is exactly one point, that is denoted by $q_e(u^*)$. We use the set $\{q_e(v)\}_{Q''_e\in\mathcal{Q}''}$ of points on the boundary of $\eta(u^*)$ in the drawing $\psi'_{J'}$ to define the corresponding set $\{p_e(v)\}_{Q''_e\in\mathcal{Q}''}$ of points on the boundary λ of the disc $\eta(u^*)$ exactly as before (where we again consider the cases where u^* is regular or irregular separately). It now remains to define the curves $\zeta_e(u^*)$ for all paths $Q''_e \in \mathcal{Q}''$.

Assume first that vertex u^* is irregular with respect to the drawings $\psi'_{J'}$ and $\varphi_{J'}$. Then for every edge $e = (u, u') \in \hat{E}$ with $u \in \hat{\Gamma}$, we let $\zeta_e(u^*)$ be any curve that is contained in $\eta(u^*) \setminus \eta'(u^*)$, that connects point $p_e(u^*)$ to the image of the vertex u (that lies on the boundary λ' of $\eta(u^*)$), such that each pair of such curves cross at most once.

Lastly, we assume that vertex u^* is regular with respect to $\psi'_{J'}$ and $\varphi_{J'}$. For every vertex $u \in \hat{\Gamma}$, let $\mathcal{S}(u) \subseteq \mathcal{Q}''$ be the set of paths whose second vertex is u. In other words, a path $Q''_e \in \mathcal{S}(u)$ iff u is an endpoint of the edge $e \in \hat{E}$. Since the curves in $\left\{\gamma_{Q''_e}\right\}_{Q''_e\in\mathcal{Q}''}$ are a non-interfering representation of the paths in $\mathcal{Q}''$ in the drawing ψ, for all $u \in \hat{\Gamma}$, there is a contiguous segment $\sigma'(u)$ of the boundary λ of $\eta(u^*)$ in the current drawing $\hat{\varphi}$, such that for every path $Q''_e \in \mathcal{S}(u)$, the point $p_e(u^*)$ lies on segment $\sigma'(u)$. Moreover, we can ensure that the segments $\left\{\sigma'(u) \mid u \in \hat{\Gamma}\right\}$ are disjoint from each other. Since the curves in $\left\{\gamma_{Q''_e}\right\}_{Q''_e\in\mathcal{Q}''}$ are non-interfering, the circular ordering of the segments $\left\{\sigma'(u) \mid u \in \hat{\Gamma}\right\}$ along λ is identical to the circular ordering of the images of the vertices in $\hat{\Gamma}$ on λ'. If the orientations of the two orderings are different, then we flip the current drawing of C, by replacing the current drawing contained in disc $\eta'(u^*)$ with its mirror image. Therefore, we can define, for every vertex $u \in \hat{\Gamma}$, for every path $Q''_e \in \mathcal{S}(u)$, a curve $\zeta_e(u^*)$, that is contained in $\eta(u^*) \setminus \eta'(u^*)$ in the drawing $\hat{\varphi}$, and connects point $p_e(u^*)$ to the image of the vertex u, while ensuring that all resulting curves in $\left\{\zeta_e(u^*) \mid e \in \hat{E}\right\}$ are mutually disjoint from each other.

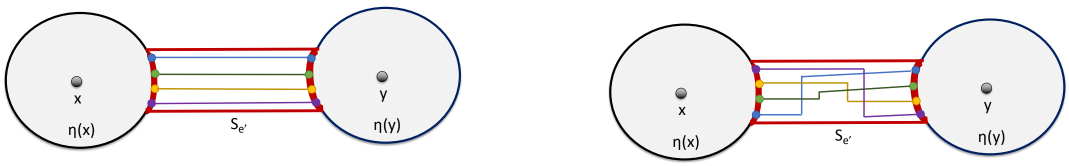
Drawing along the first edge on each path. Consider again an edge $e = (u, u') \in \hat{E}$, with $u \in \hat{\Gamma}$, and denote by $e_1 = (v_1, v_2)$ the first edge on path Q''_e, where $v_1 = t_e$. Recall that the current drawing $\hat{\varphi}$ contains the drawing of the edge $(t_e, v_2) = (v_1, v_2)$. We slightly shorten the corresponding curve, so it still originates at the image of t_e, but now it terminates at the point $p_e(v_2)$ on the boundary of $\eta(v_2)$. This defines the curve $\zeta_e(e_1)$.

Drawing along the edges. Lastly, we define, for every edge $e \in \hat{E}$, the curves $\zeta_e(e_i)$, where e_i is an edge on the path Q''_e, that is not the first edge on the path. In order to do so, we consider any edge $e' \in E(C)$, denoting $e' = (x, y)$. Recall that we are given an ordering $\mathcal{O}$ of the vertices of J that is consistent with the paths in $\mathcal{Q}''$. We assume that x appears before y in this ordering, so every path in $\mathcal{Q}''$ that contains the edge e', traverses it in the direction from x to y. We denote by $\mathcal{P}(e') \subseteq \mathcal{Q}''$ the set of paths that contain the edge e'. Recall that for each such path $Q''_e \in \mathcal{P}(e')$, we have defined a point $p'_e(x)$ on the segment $\sigma_e(x)$ of the boundary of the disc $\eta(x)$ in φ, and a point $p_e(y)$ on the segment $\sigma_e(y)$ on the boundary of the disc $\eta(y)$ in φ. We now consider two cases.

The first case is when either (i) edge e' is a regular edge (with respect to $\varphi_{J'}$ and $\psi'_{J'}$), or (ii) e' is an irregular edge, but it is not the last edge on the corresponding irregular path (since we can view the paths in $\mathcal{Q}''$ as directed towards u^*, and since we are given an ordering $\mathcal{O}$ of the vertices of $V(J)$ that is consistent with the paths in $\mathcal{Q}''$, the notion of the last edge on a path is well defined). In this case, the ordering of the points in $\left\{p'_e(x) \mid Q''_e \in \mathcal{P}(e')\right\}$ on segment $\sigma_e(x)$ is identical to the ordering of the points in $\left\{p_e(y) \mid Q''_e \in \mathcal{P}(e')\right\}$ on segment $\sigma_e(y)$. We can then define, for each path $Q''_e \in \mathcal{P}(e')$, a curve $\zeta_e(e')$ connecting points $p'_e(x)$ and $p_e(y)$ that is contained in the thin strip $S_{e'}$ around e' in φ, such that all resulting curves are mutually disjoint, in a straightforward way (see Figure 1.8(a)).

The second case is when edge e' is an irregular edge with respect to $\varphi_{J'}$ and $\psi'_{J'}$, and it is the last edge on an irregular path. In this case, the ordering of the points in $\left\{p'_e(x) \mid Q''_e \in \mathcal{P}(e')\right\}$ on segment $\sigma_e(x)$ and the ordering of the points in $\left\{p_e(y) \mid Q''_e \in \mathcal{P}(e')\right\}$ on segment $\sigma_e(y)$ are reversed. We can then define, for each path $Q''_e \in \mathcal{P}(e')$, a curve $\zeta_e(e')$ connecting points $p'_e(x)$ and $p_e(y)$ that is contained in the thin strip $S_{e'}$ around e' in φ, such that every pair of the resulting curves intersect exactly once (see Figure 1.8(b)).

This completes the definition of the images ζ_e of the edges $e \in \hat{E}$, and completes the definition of the new drawing φ' of G. It is immediate to verify that drawing φ' has all required properties. It now remains to analyze the number of crossings in this drawing. This analysis will be used later in order to bound the number of crossings in the modified drawing of the input graph G that our algorithm constructs.



(a) Drawing along a regular edge e'.

(b) Drawing along an edge e' that is the last irregular edge of an irregular path.

Figure 1.8: Drawing along edges of $E(C)$.

Analysis of the Number of Crossings. We now analyze the number of crossings in the final drawing φ' of the graph G. First, the images of the edges of $E(C)$ may cross each other in the new drawing φ', and the number of such crossings is bounded by $\mathrm{cr}(\psi)$. The edges of $E(X)$ may also cross each other, and the number of such crossings is bounded by $\mathrm{cr}(\varphi)$. The crossings caused by pairs of edges in $E(C)$ or pairs of edges in $E(X)$ are called *old crossings*. From the above discussion, the total number of old crossings is bounded by $\mathrm{cr}(\varphi)+\mathrm{cr}(\psi)$. We now bound the number of additional crossings, that we call *new crossings*. Note that, the edges of $E(C)$ may only cross the edges of $E(C)$, as the edges of $E(C)$ are drawn inside the disc $\eta'(u^*)$, while all other edges are drawn outside this disc. Therefore, all new crossings are those in which at least one edge of $\hat{E}$ participates, and it remains to bound (i) the number of crossings of the edges of $\hat{E}$ with each other, and (ii) the number of crossings between the edges of $\hat{E}$ and the edges of $E(X)$.

We assume without loss of generality that, in both φ and ψ, and every pair of edges cross at most once. This assumption is only made for the ease of notation; the analysis below works even if the images of pairs of edges are allowed to cross multiple times. We denote by

(e_1, e_2) the crossing caused by the pair e_1, e_2 of edges. Consider the original drawing φ of G. We denote by $\hat{\chi}_1$ the set of all crossings (e_1, e_2) in φ, where $e_1, e_2 \in E(J)$. We denote by $\hat{\chi}_2$ the set of all crossings (e_1, e_2) in φ, where $e_1 \in E(J)$ and $e_2 \notin E(C)$.

Consider some crossing $(e_1, e_2) \in \hat{\chi}_1$. For every pair $e, e' \in \hat{E}$ of edges such that $e_1 \in E(Q''_e)$ and $e_2 \in E(Q''_{e'})$, the new images of the edges e and e' in φ' must cross. Therefore, each crossing $(e_1, e_2) \in \hat{\chi}_1$ contributes $\mathrm{cong}_{\mathcal{Q}''}(e_1) \cdot \mathrm{cong}_{\mathcal{Q}''}(e_2) = O(\mathrm{cong}_{\mathcal{Q}}(e_1) \cdot \mathrm{cong}_{\mathcal{Q}}(e_2) \cdot \Delta^2)$ crossings to $\mathrm{cr}_{\varphi'}(G)$. Consider some crossing $(e_1, e_2) \in \hat{\chi}_2$, where $e_1 \in E(J)$ and $e_2 \notin E(C)$. For every edge $e \in \hat{E}$ such that the path Q''_e contains e_1, the new drawing of e must intersect the new drawing of e_2 in φ'. Therefore, each crossing $(e_1, e_2) \in \hat{\chi}_2$ contributes $\mathrm{cong}_{\mathcal{Q}''}(e_1) = O(\mathrm{cong}_{\mathcal{Q}}(e_1) \cdot \Delta)$ crossings to $\mathrm{cr}_{\varphi'}(G)$.

The only additional crossings in φ' are crossings between the images of the edges of $\hat{E}$ due to the re-ordering of the corresponding curves along irregular vertices and irregular edges. If an edge $e' \in E(J)$ is irregular with respect to $\psi'_{J'}$ and $\varphi_{J'}$, then the paths in set $\mathcal{P}(e') \subseteq \mathcal{Q}''$ may incur up to $(\mathrm{cong}_{\mathcal{Q}''}(e'))^2 \leq (\mathrm{cong}_{\mathcal{Q}}(e') \cdot \Delta)^2$ crossings as they are drawn along the edge e'. Assume now that a vertex $v \in V(J)$ is irregular with respect to $\psi'_{J'}$ and $\varphi_{J'}$, and let $n_v = |\mathcal{P}(v)|$ be the total number of all paths in $\mathcal{Q}''$ that contain the vertex v. Then the number of crossings due to the drawing of these paths in the disc $\eta(v)$ is at most n_v^2. If $e' \in \delta(v)$ is the edge with maximum congestion $\mathrm{cong}_{\mathcal{Q}''}(e')$, then this number of crossings is bounded by $(\Delta \cdot \mathrm{cong}_{\mathcal{Q}''}(e'))^2 \leq \Delta^4 (\mathrm{cong}_{\mathcal{Q}}(e))^2$. Recall that we have ensured that, if a vertex $v \in V(J)$ is irregular with respect to $\psi'_{J'}$ and $\varphi_{J'}$, then it is irregular with respect to ψ_J and φ_J. Similarly, if an edge $e \in E(J)$ is irregular with respect to $\psi'_{J'}$ and $\varphi_{J'}$, then it is irregular with respect to ψ_J and φ_J.

Denote by E^* the set of all edges $e \in E(J)$, such that either e is an irregular edge with respect to ψ_J and φ_J, or at least one endpoint of e is an irregular vertex with respect to ψ_J

and φ_J. Then the total number of new crossings in φ' is bounded by:

$$O\left(\sum_{(e_1,e_2)\in\hat{\chi}_1} \Delta^2 \cdot \mathrm{cong}_{\mathcal{Q}}(e_1)\cdot \mathrm{cong}_{\mathcal{Q}}(e_2) + \sum_{(e_1,e_2)\in\hat{\chi}_2} \Delta\cdot \mathrm{cong}_{\mathcal{Q}}(e_1) + \sum_{e\in E^*} \Delta^4(\mathrm{cong}_{\mathcal{Q}}(e))^2\right).$$

Finally, the total number of crossings in φ' is bounded by:

$$\mathrm{cr}(\varphi)+\mathrm{cr}(\psi)+O\left(\sum_{(e_1,e_2)\in\hat{\chi}_1} \Delta^2 \cdot \mathrm{cong}_{\mathcal{Q}}(e_1)\cdot \mathrm{cong}_{\mathcal{Q}}(e_2) + \sum_{(e_1,e_2)\in\hat{\chi}_2} \Delta\cdot \mathrm{cong}_{\mathcal{Q}}(e_1) + \sum_{e\in E^*} \Delta^4(\mathrm{cong}_{\mathcal{Q}}(e))^2\right)$$

The following observation is immediate from the analysis above and will be useful in bounding the total number of crossings in our algorithm.

Observation 1.7.3. *Let (e_1, e_2) be a new crossing in φ' (that is, edges e_1 and e_2 cross in φ' but do not cross in φ or in ψ). Then either $e_1, e_2 \in \hat{E}$, or one of these edges belongs to $\hat{E}$ and the other to X. Moreover, if $e \in E(X)$, and K_e is the set of all edges of J whose image in φ crosses the image of e, then the total number of new crossings in φ' in which edge e participates is at most $\sum_{e'\in K_e} \Delta \cdot \mathrm{cong}_{\mathcal{Q}}(e')$.*

1.7.4 Completing the Proof of Theorem 1.3.4

We now provide the proof of Theorem 1.3.4, by showing an algorithm that produces the desired drawing φ' of the graph G. The algorithm processes all clusters in $\mathcal{C}_1 \cup \mathcal{C}_2$ one-by-one, with clusters in $\mathcal{C}_1$ processed before clusters in $\mathcal{C}_2$. When a cluster C is processed, we modify the current drawing of the graph G, so that it becomes canonical with respect to C. For convenience, we denote $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$, and we denote the clusters in $\mathcal{C}$ by $C_1, \ldots, C_r$. We assume that the type-1 clusters appear before the type-2 clusters in this ordering.

Our algorithm proceeds by repeatedly invoking procedure ProcDraw. As part of input, the procedure requires a collection $\mathcal{Q}$ of guiding paths. We start by defining, for every cluster

$C_i \in \mathcal{C}$, a collection $\mathcal{Q}_i$ of paths that are contained in C_i, and connect every vertex of $\Gamma(C_i) = \Gamma \cap V(C_i)$ to some fixed vertex u_i^* of C_i.

Defining the Guiding Paths

In this subsection, we define, for every cluster $C_i \in \mathcal{C}$, a collection $\mathcal{Q}_i$ of paths that are contained in C_i, and connect every vertex of $\Gamma(C_i) = \Gamma \cap V(C_i)$ to some fixed vertex u_i^* of C_i. These paths will eventually be used by procedure $\mathsf{ProcDraw}$ as guiding paths. The definition of the set $\mathcal{Q}_i$ of paths is different depending on whether C_i is a type-1 or a type-2 cluster.

Type-1 clusters. Let $C_i \in \mathcal{C}_1$ be a type-1 cluster. Recall that the number of terminals in $\Gamma(C_i)$ is at most $\mu\Delta$. We let u_i^* be an arbitrary vertex of C_i. Recall that we have defined, in Theorem 1.3.3, a set $E^*(C_i)$ of edges, such that graph $C_i \setminus E^*(C_i)$ is connected, and the drawing of $C_i \setminus E^*(C_i)$ induced by ψ_{C_i} is planar. Consider now any spanning tree of $C_i \setminus E^*(C_i)$, rooted at the vertex u_i^*. For every terminal $t \in \Gamma(C_i)$, let Q_t be the unique path connecting t to u_i^* in this tree. We then set $\mathcal{Q}_i = \{Q_t \mid t \in \Gamma(C_i)\}$ be the set of the guiding paths for the cluster C_i. Since $|\mathcal{Q}_i| \leq \mu\Delta$, for every edge $e \in E(C_i)$, $\operatorname{cong}_{\mathcal{Q}_i}(e) \leq \mu\Delta$.

Let $J_i \subseteq C_i$ be the graph obtained from the union of the paths in $\mathcal{Q}_i$. Then J_i is a tree with at most $O(\mu\Delta)$ leaves, and so it has $O(\mu\Delta)$ vertices of degree greater than 2. We denote by ψ_{J_i} the planar drawing of J_i induced by the drawing ψ_{C_i} of C_i, and we denote by φ_{J_i} the drawing of J_i induced by the original drawing φ of G. Clearly, drawing ψ_{J_i} of J_i is planar, and the number of vertices and of edges of J_i that are irregular with respect to ψ_{J_i} and φ_{J_i} is at most $O(\mu\Delta)$. Let $\mathsf{IRG}_i \subseteq E(J_i)$ be the set of all edges $e \in E(J_i)$, such that either e is irregular with respect to the drawings ψ_{J_i} and φ_{J_i} of J_i, or at least one endpoint of e is irregular with respect to these drawings. The following observation is immediate from the above discussion.

Observation 1.7.4. *If $C_i \in \mathcal{C}$ is a type-1 cluster then* $|\mathsf{IRG}_i| \leq O(\mu\Delta)$.

For every edge $e \in E(C_i)$, we define its weight $w(e)$ as follows. We start with $w(e)$ being the number of crossings in the drawing φ of G, in which edge e participates. Additionally, if $e \in \mathsf{IRG}_i$, then we increase $w(e)$ by 1. From the above discussion, we get that the following inequality that will be useful for us later:

$$\sum_{e \in E(C_i)} w(e) \cdot (\mathrm{cong}_{\mathcal{Q}_i}(e))^2 \leq O(\mu^2\Delta^2) \cdot \sum_{e \in E(C_i)} w(e). \tag{1.3}$$

Type-2 clusters. Let $C_i \in \mathcal{C}_2$ be a type-2 cluster. For convenience, we denote $C'_i = C_i \setminus A_{C_i}$. In order to define the set $\mathcal{Q}_i$ of guiding paths for C_i, we use the following lemma, that generalizes Lemma D.10 from [9]. For completeness, we provide the proof of the lemma in Section 1.9.6.

Lemma 1.7.5. *There is an efficient algorithm, that, given an n-vertex planar graph H, non-negative weights $\{w(e)\}_{e \in E(H)}$ on its edges, and a subset $S \subseteq V(H)$ of vertices of H that is α'-well-linked in H, for any parameter $0 < \alpha' < 1$, computes a vertex $u^* \in V(H)$ together with a set $\mathcal{Q}$ of $|S|$ paths in H routing the vertices of S to u^*, such that:*

$$\sum_{e \in E(H)} w(e) \cdot (\mathrm{cong}_{\mathcal{Q}}(e))^2 = O\left(\frac{\log n}{(\alpha')^4} \cdot \sum_{e \in E(H)} w(e)\right).$$

For every edge $e \in E(C'_i)$, we define its weight $w'(e)$ as follows. We start with $w'(e)$ being the number of crossings in the drawing φ of G, in which edge e participates. Additionally, if e is an irregular vertex with respect to the drawing $\psi_{C'_i}$ of C'_i, and the drawing $\varphi_{C'_i}$ of C'_i induced by φ, then we increase $w'(e)$ by 1. Also, for every endpoint of e that is irregular with respect to these two drawings, we increase $w'(e)$ by 1.

We apply Lemma 1.7.5 to graph C'_i with the edge weights $w'(e)$ that we have defined, and the set $S = \Gamma(C_i)$ of its vertices; recall that from the well-linkedness property of type-2 clusters, set $\Gamma(C_i)$ of vertices is α-well-linked in C'_i. The algorithm from the lemma then returns a vertex $u^*_i \in V(C_i)$, and a set $\mathcal{Q}_i = \{Q_t \mid t \in \Gamma(C_i)\}$ of paths in graph C'_i, where each path Q_t connects t to u^*_i, and moreover:

$$\sum_{e \in E(C'_i)} w'(e) \cdot (\mathrm{cong}_{\mathcal{Q}_i}(e))^2 \leq O\left(\frac{\log n}{\alpha^4} \cdot \left(\sum_{e \in E(C'_i)} w'(e) + 1\right)\right). \tag{1.4}$$

As before, we let J_i be the graph obtained from the union of the paths in $\mathcal{Q}_i$. Note that J_i may no longer be a tree. As before, we denote by ψ_{J_i} the drawing of J_i induced by the drawing $\psi_{C'_i}$ of C'_i (which must be planar), and we denote by φ_{J_i} the drawing of J_i induced by the drawing φ of G.

Let $\mathsf{IRG}_i \subseteq E(J_i)$ be the set of all edges $e \in E(J_i)$, such that either e is irregular with respect to the drawings ψ_{J_i} and φ_{J_i} of J_i, or at least one endpoint of e is irregular with respect to these drawings. For every edge $e \in E(C'_i)$, we let its new weight $w(e)$ be defined as follows. Initially, we let $w(e)$ be the number of crossings in the drawing φ of G in which edge e participates. If $e \in \mathsf{IRG}_i$, then we increase $w(e)$ by 1. Clearly, $w(e) \leq w'(e)$, and so:

$$\sum_{e \in E(C'_i)} w(e) \cdot (\mathrm{cong}_{\mathcal{Q}}(e))^2 \leq O\left(\frac{\log n}{\alpha^4} \cdot \left(\sum_{e \in E(C'_i)} w'(e) + 1\right)\right). \tag{1.5}$$

In the remainder of the algorithm, we perform r iterations. The input to the ith iteration is a drawing φ_{i-1} of the graph G, that is canonical with respect to the clusters $C_1, \ldots, C_{i-1}$. The output of the ith iteration is a drawing φ_i of G, that is canonical with respect to clusters $C_1, \ldots, C_i$. Initially, we set $\varphi_0 = \varphi$. We now focus on the description of the ith iteration, when cluster C_i is processed.

Processing a Cluster

We now describe an iteration when cluster $C_i \in \mathcal{C}$ is processed. Recall that, if C_i is a type-2 cluster, then we have denoted $C'_i = C_i \setminus A_{C_i}$. In order to simplify the notation, if C_i is a type-1 cluster, we will denote $C'_i = C_i$. Let $\mathcal{D}_i$ be the set of all discs $D(R)$ for the bridges $R \in \mathcal{R}_G(C'_i)$. For every disc $D \in \mathcal{D}_i$, we let $\mathcal{R}_i^D \subseteq \mathcal{R}_G(C'_i)$ be the set of all the bridges $R \in \mathcal{R}_G(C'_i)$ with $D(R) = D$. We also let X_i^D be the graph obtained by the union of all bridges in $\mathcal{R}_i^D$, and $\hat{E}_i^D$ be the set of all edges of G connecting vertices of C'_i to vertices of X_i^D. Lastly, we let $\hat{\Gamma}_i^D$ be the set of all terminals in $\bigcup_{R\in\mathcal{R}_i^D} L(R)$, and we let $\mathcal{Q}_i^D \subseteq \mathcal{Q}_i$ be the set of all paths originating at the vertices of $\hat{\Gamma}_i^D$. Observe that, in the drawing $\psi_{C'_i}$ of C'_i, all vertices of $\hat{\Gamma}_i^D$ lie on the boundary of the disc D. Let J_i^D be the graph obtained from the union of the paths in $\mathcal{Q}_i^D$. Note that every path $Q_t \in \mathcal{Q}_i$ may participate in at most Δ different path sets in $\left\{\mathcal{Q}_i^D\right\}_{D\in\mathcal{D}_i}$.

Let $G_i^D \subseteq G$ be the sub-graph of G consisting of the union of the graphs C'_i, X_i^D, and the edges of $\hat{E}_i^D$. Let φ^D be the drawing of G_i^D that is induced by the current drawing φ_{i-1} of G. We apply $\mathsf{ProcDraw}$ to graph G_i^D, with the subgraphs $C = C'_i$, $X = X_i^D$, together with the vertex u_i^*, and the set $\mathcal{Q}_i^D$ of paths routing the vertices of $\hat{\Gamma}_i^D$ to u_i^* in C'_i. Recall that the corresponding graph J_i^D (which is obtained from the union of the paths in $\mathcal{Q}_i^D$) is guaranteed to be planar, and its drawing induced by $\psi_{C'_i}$ is also planar. We denote the drawing of the graph G_i^D produced by $\mathsf{ProcDraw}$ by $\hat{\varphi}_i^D$. Recall that the drawing of C'_i induced by $\hat{\varphi}_i^D$ is identical to $\psi_{C'_i}$, and that all vertices of X_i^D are drawn inside the disc D, with the vertices of $\hat{\Gamma}_i^D$ drawn on the boundary of D.

Once every disc $D \in \mathcal{D}_i$ is processed, we combine the resulting drawings $\hat{\varphi}^D$ together, in order to obtain the final drawing φ_i of the graph G. In order to do so, we start by placing the drawing $\psi_{C'_i}$ on the sphere. Next, for every disc $D \in \mathcal{D}_i$, we copy the drawing of graph $X_i^D \cup \hat{E}_i^D$ in $\hat{\varphi}_i^D$ to this new drawing, so that the two copies of the disc D coincide with each other, and the images of the vertices of $\hat{\Gamma}_i^D$ in both drawings coincide. It is immediate to verify that the resulting drawing φ_i of G is canonical with respect to C_i. We next claim that, if drawing φ_{i-1} was canonical with respect to some cluster $C \in \mathcal{C}$, then so is drawing φ_i.

Claim 1.7.6. *Let $C_j \in \mathcal{C}$ be a cluster, such that drawing φ_{i-1} was canonical with respect to C_j. Then drawing φ_i remains canonical with respect to C_j.*

Proof. Observe that there must be some bridge $R \in \mathcal{R}_G(C'_j)$ that contains the graph C'_i. Consider the corresponding disc $D(R)$ in the drawing $\psi_{C'_j}$ of C'_j, and the corresponding disc, that we also denote by $D(R)$, in the drawing φ_{i-1} of G. Recall that in the drawing φ_{i-1}, all vertices and edges of R are drawn in the disc $D(R)$. Let D^* be the disc on the sphere that is the complement of $D(R)$, so the two discs share their boundaries but are otherwise disjoint.

Note that similarly, there must be some bridge $R' \in \mathcal{R}_G(C'_i)$ that contains the graph C'_j. We let $D(R')$ be the corresponding disc in the drawing $\psi_{C'_i}$ of C'_i. Note that the cluster C'_j was unaffected when discs $D \in \mathcal{D}_i \setminus \left\{D(R')\right\}$ were processed, as C'_j is disjoint from the corresponding graphs G_i^D. When disc $D(R')$ was processed, we have deleted the images of vertices and edges of C'_i, and modified the images of the edges of $\hat{E}_i^{D(R')}$ accordingly. However, since graph C'_i is drawn outside disc D^* in φ_{i-1}, this did not change the part of the drawing that lies in D^*. When we computed the final drawing φ_i of G, we have copied the drawing inside the disc $D(R')$ in $\hat{\varphi}_i^D$ to the same disc in φ_i. Since $D^* \subseteq D(R')$, this again did not affect the drawing in D^*. Therefore, the part of the drawing φ_{i-1} of the graph G that appeared in disc D^* remains unchanged in the drawing φ_i. It is then easy to verify that drawing φ_i remains canonical with respect to C_j. □

We let $\varphi' = \varphi_r$ be the drawing of G that we obtain after all clusters of $\mathcal{C}$ are processed. It now remains to analyze the number of crossings in φ'.

Analyzing the Number of Crossings

Consider some cluster $C_i \in \mathcal{C}$. Our goal is to bound the increase in the number of crossings due to iteration i, that is, $\mathrm{cr}(\varphi_i) - \mathrm{cr}(\varphi_{i-1})$. Let χ_i be the set of all crossings (e_1, e_2) in the original drawing φ of G, with $e_1, e_2 \in E(J_i)$. Notice that the drawings of C'_i in φ and φ_{i-1} are identical. Let χ'_i be the set of all crossings (e_1, e_2) in the drawing φ_{i-1} of G with $e_1 \in E(J_i)$ and $e_2 \notin E(C'_i)$. Recall that we have denoted by $\mathsf{IRG}_i \subseteq E(J_i)$ the set of all edges $e \in E(J_i)$, such that either e is irregular with respect to the drawings ψ_{J_i} and φ_{J_i} of J_i, or at least one endpoint of e is irregular with respect to these drawings.

Consider now some disc $D \in \mathcal{D}_i$. Let $\chi_i(D)$ be the set of all crossings (e_1, e_2) in the original drawing φ of G, with $e_1, e_2 \in E(J_i^D)$, and let $\chi'_i(D)$ be the set of all crossings (e_1, e_2) in the drawing φ_{i-1} of G with $e_1 \in E(J_i^D)$ and $e_2 \notin E(C'_i)$. We also denote by $\mathsf{IRG}_i(D) \subseteq E(J_i^D)$ the set of all edges $e \in E(J_i^D)$, such that either e is irregular with respect to the drawings $\psi_{J_i^D}$ and $\varphi_{J_i^D}$ of J_i^D induced by the drawing ψ_{C_i} of C_i and φ of G, respectively, or at least one endpoint of e is irregular with respect to these drawings. It is easy to verify that, if $e \in \mathsf{IRG}_i(D)$, then $e \in \mathsf{IRG}_i$ must hold.

From the analysis of $\mathsf{ProcDraw}$, we get that the number $z_i(D)$ of new crossings in the drawing $\hat{\varphi}_i^D$ is at most:

$$O\Bigg(\sum_{(e_1,e_2)\in\chi_i(D)} \Delta^2 \operatorname{cong}_{\mathcal{Q}_i^D}(e_1) \operatorname{cong}_{\mathcal{Q}_i^D}(e_2) + \sum_{(e_1,e_2)\in\chi'_i(D)} \Delta \operatorname{cong}_{\mathcal{Q}_i^D}(e_1) + \sum_{e\in\mathsf{IRG}_i^D} \Delta^4 (\operatorname{cong}_{\mathcal{Q}_i^D}(e))^2 \Bigg).$$

Consider some crossing $(e_1, e_2) \in \chi_i(D)$. We can view this crossing as contributing $\operatorname{cong}_{\mathcal{Q}_i^D}(e_1) \cdot \operatorname{cong}_{\mathcal{Q}_i^D}(e_2) \cdot \Delta^2$ crossings to the first term of $z_i(D)$. If $\operatorname{cong}_{\mathcal{Q}_i^D}(e_1) \geq \operatorname{cong}_{\mathcal{Q}_i^D}(e_2)$, then we let the edge e_1 "pay" $\left(\operatorname{cong}_{\mathcal{Q}_i^D}(e_1) \cdot \Delta\right)^2 \geq \operatorname{cong}_{\mathcal{Q}_i^D}(e_1) \cdot \operatorname{cong}_{\mathcal{Q}_i^D}(e_2) \cdot \Delta^2$ units for these crossings, and otherwise we let the edge e_2 pay $\left(\operatorname{cong}_{\mathcal{Q}_i^D}(e_2) \cdot \Delta\right)^2 \geq \operatorname{cong}_{\mathcal{Q}_i^D}(e_1) \cdot \operatorname{cong}_{\mathcal{Q}_i^D}(e_2) \cdot \Delta^2$

units for these crossings. Therefore, we obtain the following bound:

$$z_i(D) \le O\Bigg(\sum_{e_1 \in E(J_i^D)} \sum_{e_2 : (e_1,e_2) \in \chi_i(D)} \left(\mathrm{cong}_{\mathcal{Q}_i^D}(e_1) \cdot \Delta\right)^2 + \sum_{e \in \mathsf{IRG}_i^D} \Delta^4 (\mathrm{cong}_{\mathcal{Q}_i^D}(e))^2 + \sum_{(e_1,e_2) \in \chi_i'(D)} \mathrm{cong}_{\mathcal{Q}_i^D}(e_1) \cdot \Delta \Bigg).$$

Summing up over all discs $D \in \mathcal{D}_i$, and noting that, for every edge $e \in E(J_i)$, $\sum_{D \in \mathcal{D}_i} \mathrm{cong}_{\mathcal{Q}_i^D}(e) \le O(\Delta\, \mathrm{cong}_{\mathcal{Q}_i}(e))$, we get that the total increase $\mathrm{cr}(\varphi_i) - \mathrm{cr}(\varphi_{i-1})$ in the number of crossings is bounded by:

$$O\left(\sum_{e_1 \in E(J_i)} \sum_{e_2 : (e_1,e_2) \in \chi_i} \Delta^4 \cdot \left(\mathrm{cong}_{\mathcal{Q}_i}(e_1)\right)^2 + \sum_{e \in \mathsf{IRG}_i} \Delta^6 (\mathrm{cong}_{\mathcal{Q}_i}(e))^2 + \sum_{(e_1,e_2) \in \chi_i'} \Delta^2 \cdot \mathrm{cong}_{\mathcal{Q}_i}(e_1) \right).$$

Recall that for every edge $e \in E(C_i')$, we defined its weight $w(e)$ as follows. First, we let $w(e)$ be the number of crossings in φ in which edge e participates. Then, if $e \in \mathsf{IRG}_i$, we increased $w(e)$ by 1. Therefore, we get that:

$$\mathrm{cr}(\varphi_i) - \mathrm{cr}(\varphi_{i-1}) \le O\left(\sum_{e \in E(J_i)} \Delta^6 \cdot w(e) \cdot \left(\mathrm{cong}_{\mathcal{Q}_i}(e)\right)^2 \right) + O\left(\sum_{(e_1,e_2) \in \chi_i'} \Delta^2 \cdot \mathrm{cong}_{\mathcal{Q}_i}(e_1) \right).$$

We denote $\Upsilon_i = O\left(\sum_{e \in E(J_i)} \Delta^6 \cdot w(e) \cdot \left(\mathrm{cong}_{\mathcal{Q}_i}(e)\right)^2\right)$ and $\Upsilon_i' = O\left(\sum_{(e_1,e_2) \in \chi_i'} \Delta^2 \cdot \mathrm{cong}_{\mathcal{Q}_i}(e_1)\right)$, and we analyze these terms separately.

Bounding $\sum_i \Upsilon_i'$. Consider a cluster $C_i \in \mathcal{C}$, and let $e \in E(G \setminus (C_i' \cup \delta_G(C_i')))$ be an edge of G that does not lie in $C_i' \cup \delta_G(C_i')$. Denote by K_e the set of all edges of J_i whose image in φ_{i-1} crosses the image of e. From Observation 1.7.3, the total number of crossings (e', e)

in φ_i that do not belong to φ_{i-1}, over all edges $e' \in E(G)$ is at most:

$$\sum_{e' \in K_e} \sum_{D \in \mathcal{D}_i} \Delta \cdot \mathrm{cong}_{\mathcal{Q}_i^D}(e') \leq \sum_{e' \in K_e} \Delta^2 \cdot \mathrm{cong}_{\mathcal{Q}_i}(e').$$

Observe that for each such new crossing (e'', e), edge e'' must lie in E''. For every edge $e' \in K_e$, we say that the crossing (e', e) in φ_{i-1} is responsible for $\Delta^2 \mathrm{cong}_{\mathcal{Q}_i}(e')$ new crossings in φ_i. We also say that crossing (e', e) contributes $\Delta^2 \mathrm{cong}_{\mathcal{Q}_i}(e')$ crossings to Υ'_i. If (e', e) is a crossing of φ_{i-1} with $e \in \delta_G(C'_i)$ and $e' \in J_i$, then we say that it contriburtes $\Delta^2 \mathrm{cong}_{\mathcal{Q}_i}(e')$ crossings to Υ'_i, but it is not responsible for any new crossings. Observe that the sum of the contributions of all crossings (e', e) of φ_{i-1} with $e \in E(G \setminus C'_i)$ and $e' \in J_i$ is at least $\Omega(\Upsilon'_i)$.

Consider now some crossing (e_1, e_2) in the original drawing φ of G, and assume that $e_1 \in C_i$ and $e_2 \in C_j$, where $i < j$. The crossing (e_1, e_2) contributes $\mathrm{cong}_{\mathcal{Q}_i}(e_1) \cdot \Delta^2$ crossings to Υ'_i. It is also responsible for $\mathrm{cong}_{\mathcal{Q}_i}(e_1) \cdot \Delta^2$ new crossings of the edge e_2. When cluster C_j is processed, each one of these new crossings contributes $\mathrm{cong}_{\mathcal{Q}_j}(e_2) \cdot \Delta^2$ crossings to Υ'_j. Therefore, altogether, crossing (e_1, e_2) is responsible for $\mathrm{cong}_{\mathcal{Q}_i}(e_1) \cdot \mathrm{cong}_{\mathcal{Q}_j}(e_2) \cdot \Delta^4$ crossings in $\sum_{i'=1}^{r} \Upsilon'_{i'}$. If $\mathrm{cong}_{\mathcal{Q}_i}(e_1) \leq \mathrm{cong}_{\mathcal{Q}_j}(e_2)$, then we make edge e_2 responsible for all these crossings and charge it $(\mathrm{cong}_{\mathcal{Q}_j}(e_2))^2 \cdot \Delta^4 \geq \mathrm{cong}_{\mathcal{Q}_i}(e_1) \cdot \mathrm{cong}_{\mathcal{Q}_j}(e_2) \cdot \Delta^4$ for them, and otherwise, we make edge e_1 responsible for these crossings, and charge it $(\mathrm{cong}_{\mathcal{Q}_j}(e_1))^2 \cdot \Delta^4 \geq \mathrm{cong}_{\mathcal{Q}_i}(e_1) \cdot \mathrm{cong}_{\mathcal{Q}_j}(e_2) \cdot \Delta^4$ for them.

If (e_1, e_2) is a crossing in φ where exactly one of the two edges e_1, e_2 lies in some cluster C_i and the other edge lies in E'', then the analysis is similar except that this crossing only contributes to Υ'_i and is charged to the corresponding edge. If both e_1, e_2 lie in the same cluster C_i, then crossing (e_1, e_2) does not contribute to $\sum_{i'=1}^{r} \Upsilon'_{i'}$. Recall that for every cluster $C_i \in \mathcal{C}$, for every edge $e \in E(C'_i)$, we have defined weight $w(e)$, which is at least the number of crossings in the drawing φ of G in which edge e participates. To summarize, from the above discussion, we get that:

$$\sum_{i=1}^{r} \Upsilon'_i \leq O\left(\sum_{i=1}^{r} \sum_{e\in C'_i} \Delta^4 \cdot w(e) \left(\mathrm{cong}_{\mathcal{Q}_i}(e)\right)^2\right).$$

Altogether, we then get that:

$$\mathrm{cr}(\varphi') - \mathrm{cr}(\varphi) \leq \sum_{i=1}^{r} O\left(\Delta^6 \cdot \sum_{e\in C'_i} w(e) \left(\mathrm{cong}_{\mathcal{Q}_i}(e)\right)^2\right).$$

Final Accounting. Recall that we have denoted, for every cluster $C_i \in \mathcal{C}$, by $\mathsf{IRG}_i \subseteq E(J_i)$ the set of all edges $e \in E(J_i)$, such that either e is irregular with respect to the drawings ψ_{J_i} and φ_{J_i} of J_i, or at least one endpoint of e is irregular with respect to these drawings. We also denote by x_i the total number of crossings in the drawing φ of G in which the edges of C'_i participate. It is then easy to verify that for every cluster $C_i \in \mathcal{C}$, $\sum_{e\in E(C'_i)} w(e) \leq x_i + |\mathsf{IRG}_i|$.

Consider now some cluster $C_i \in \mathcal{C}$, and assume first that $C_i \in \mathcal{C}_1$. From Equation 1.3:

$$\begin{aligned}\sum_{e\in E(C_i)} w(e) \cdot (\mathrm{cong}_{\mathcal{Q}_i}(e))^2 &\leq O(\mu^2\Delta^2) \cdot \sum_{e\in E(C_i)} w(e)\\ &\leq O(\mu^2\Delta^2)(x_i + |\mathsf{IRG}_i|)\\ &\leq O(\mu^3\Delta^3(x_i+1))\\ &\leq O(\mathrm{poly}(\Delta\log n)(x_i+1)).\end{aligned}$$

(We have used the fact that, if C_i is a type-1 cluster then $|\mathsf{IRG}_i| \leq O(\mu\Delta)$, and that $\mu = O(\mathrm{poly}(\Delta\log n)$.)

Assume now that C_i is a type-2 cluster. From Equation 1.5, we get that:

$$\sum_{e\in E(C'_i)} w(e)\cdot(\mathrm{cong}_{\mathcal{Q}}(e))^2 \le O\left(\frac{\log n}{\alpha^4}\cdot\left(\sum_{e\in E(C_i)} w'(e)+1\right)\right).$$

Let $\mathsf{IRG}'_i \subseteq E(C'_i)$ denote the set of all edges $e \in E(C'_i)$, such that either e is an irregular edge with respect to the drawing $\psi_{C'_i}$ of C'_i, and the drawing $\varphi_{C'_i}$ of C'_i induced by φ, or at least one endpoint of e is irregular with respect to these drawings. Recall that $\sum_{e\in E(C'_i)} w'(e) = O(x_i + |\mathsf{IRG}'_i|)$. Therefore, we get that:

$$\sum_{e\in E(C'_i)} w(e)\cdot(\mathrm{cong}_{\mathcal{Q}}(e))^2 \le O\left(\frac{\log n}{\alpha^4}\cdot(x_i+|\mathsf{IRG}'_i|+1)\right) \le O\left(\mathrm{poly}(\Delta\log n)\cdot(x_i+|\mathsf{IRG}'_i|+1)\right),$$

since $\alpha = \Theta(1/\mathrm{poly}(\Delta\log n))$. Altogether, the number of crossings in the new drawing φ' of φ can now be bounded as:

$$\begin{aligned}\mathrm{cr}(\varphi') &\le \mathrm{cr}(\varphi) + O\left(\mathrm{poly}(\Delta\log n)\right)\cdot\left(\sum_{i=1}^{r} x_i + \sum_{C_i\in\mathcal{C}_2}|\mathsf{IRG}'_i| + |\mathcal{C}|\right)\\ &\le O\left(\mathrm{poly}(\Delta\log n)(\mathrm{cr}(\varphi)+|E''|)\right) + O\left(\mathrm{poly}(\Delta\log n)\right)\cdot\sum_{C_i\in\mathcal{C}_2}|\mathsf{IRG}'_i|.\end{aligned}$$

The next claim will then finish the proof of Theorem 1.3.4.

Claim 1.7.7. $\sum_{C_i\in\mathcal{C}_2}|\mathsf{IRG}'_i| \le O\left(\Delta^2(|E''|+\mathrm{cr}(\varphi))\right)$.

Proof. Consider some cluster $C_i \in \mathcal{C}_2$. Recall that set $\mathsf{IRG}'_i \subseteq E(C'_i)$ contains all edges $e \in E(C'_i)$, such that either e is an irregular edge with respect to the drawing $\psi_{C'_i}$ of C'_i, and the drawing $\varphi_{C'_i}$ of C'_i induced by φ, or at least one endpoint of e is irregular with respect to these drawings. In other words, $|\mathsf{IRG}'_i| \le \Delta\cdot|\mathsf{IRG}_V(\varphi_{C'_i},\psi_{C'_i})| + |\mathsf{IRG}_E(\varphi_{C'_i},\psi_{C'_i})|$. Lemma 1.7.2 guarantees that:

$$|\mathsf{IRG}_V(\varphi_{C'_i},\psi_{C'_i})\setminus S_2(C'_i)| + |\mathsf{IRG}_E(\varphi_{C'_i},\psi_{C'_i})\setminus E_2(C'_i)| \le O(\mathrm{cr}(\varphi_{C'_i})) \le O(x_i),$$

where $S_2(C'_i)$ is the set of all vertices that participate in 2-separators in C'_i, and $E_2(C'_i)$ is the set of all edges of C'_i that have both endpoints in $S_2(C'_i)$. Unfortunately, the definition of type-2 acceptable clusters does not provide any bound on the cardinality of the set $S_2(C'_i)$. It does, however, ensure that $|S_2(C_i)| \leq O(\Delta|\Gamma(C_i)|)$, where C_i is the original cluster, that may contain artificial edges. Unfortunately, the original drawing φ of G does not include the drawings of the artificial edges. However, using the embeddings of these edges, we can easily transform drawing φ of G into a drawing $\tilde{\varphi}$ of $\bigcup_{C_i \in \mathcal{C}_2} C_i$, without increasing the number of crossings. Applying Lemma 1.7.2 to the resulting drawings of graphs $C_i \in \mathcal{C}_2$ will then finish the proof. We now turn to provide a more detailed proof.

Consider the original drawing φ of graph G. We transform it into a drawing $\tilde{\varphi}$ of $\bigcup_{C_i \in \mathcal{C}_2} C_i$, as follows. Recall that the decomposition $\mathcal{D}$ of G into acceptable clusters contains an embedding $\mathcal{P} = \{P(e) \mid e \in A\}$ of all artificial edges via paths that are internally disjoint. Moreover, for every edge $e = (x, y) \in A$, there is a type-1 cluster $C(e) \in \mathcal{C}_1$, such that $P(e) \setminus \{x, y\}$ is contained in $C(e)$, and the clusters $C(e)$ are distinct for all edges $e \in A$. We delete from φ all vertices and edges except those participating in graphs C'_i for $C_i \in \mathcal{C}_2$, and in paths in $\mathcal{P}$. By suppressing all inner vertices on the paths in $\mathcal{P}$, we obtain a drawing $\tilde{\varphi}$ of $\bigcup_{C_i \in \mathcal{C}_2} C_i$, that contains at most $\mathrm{cr}(\varphi)$ crossings. Consider now some cluster $C_i \in \mathcal{C}_2$. Let $\tilde{\varphi}_i$ be the drawing of C_i that is induced by $\tilde{\varphi}$. Observe that, if a vertex $v \in V(C_i)$ is irregular with respect to $\varphi_{C'_i}$, $\psi_{C'_i}$, then it must be irregular with respect to $\tilde{\varphi}_i$ and ψ'_{C_i}. Similarly, if an edge $e \in E(C'_i)$ is irregular with respect to $\varphi_{C'_i}$, $\psi_{C'_i}$, then it must be irregular with respect to $\tilde{\varphi}_i$ and ψ'_{C_i}. Therefore, if we denote by IRG''_i the set of all edges $e \in C_i$, such that either e is irregular with respect to $\tilde{\varphi}_i$ and ψ'_{C_i}, or at least one endpoint of e is irregular with respect to these two drawings, then $|\mathsf{IRG}''_i| \geq |\mathsf{IRG}'_i|$. Let x'_i be the total number of crossings in $\tilde{\varphi}_i$. Let $E_2^i \subseteq E(C_i)$ be the set of all edges that are incident to vertices of $S_2(C_i)$ (vertices that participate in 2-separators in C_i). Then, from Lemma 1.7.2:

$$|\mathsf{IRG}''_i \setminus E_2^i| \leq O(\Delta \cdot x'_i).$$

Moreover, from the definition of type-2 acceptable clusters, $|S_2(C_i)| \leq O(\Delta|\Gamma(C_i)|)$, and so $|E_2^i| \leq O(\Delta^2|\Gamma(C_i)|)$. Overall, we conclude that:

$$|\mathsf{IRG}'_i| \leq |\mathsf{IRG}''_i| \leq O(\Delta^2(x'_i + |\Gamma(C_i)|)).$$

Summing up over all clusters $C_i \in \mathcal{C}_2$, we get that:

$$\begin{aligned}\sum_{C_i \in \mathcal{C}_2} |\mathsf{IRG}'_i| &\leq O(\Delta^2) \sum_{C_i \in \mathcal{C}_2} x'_i + O(\Delta^2) \sum_{C_i \in \mathcal{C}_2} |\Gamma(C_i)| \\ &\leq O\left(\Delta^2 \cdot (\mathrm{cr}(\tilde{\varphi}) + |E''|)\right) \\ &\leq O\left(\Delta^2 \cdot (\mathrm{cr}(\varphi) + |E''|)\right).\end{aligned}$$

□

1.8 Handling Non 3-Connected Graphs

So far we have provided the proof of Theorem 1.1.1 for the special case where the input graph G is 3-connected. In this section we extend the proof to arbitrary graphs.

We first compute the block decomposition $\mathcal{B} = \mathcal{B}(G)$ of the input graph G, and denote $\tilde{\mathcal{B}} = \left\{\tilde{B} \mid B \in \mathcal{B}\right\}$. Since each graph $\tilde{B} \in \tilde{\mathcal{B}}$ is 3-connected, we can use the algorithm from Section 1.3.3 to compute, for each graph $\tilde{B} \in \tilde{\mathcal{B}}$, an instance $(G_{\tilde{B}}, \Sigma_{\tilde{B}})$ of the MCNwRS problem, such that the number of edges in $G_{\tilde{B}}$ is at most $O\left(\mathsf{OPT}_{\mathsf{cr}}(\tilde{B}) \cdot \mathrm{poly}(\Delta \log n)\right)$, and $\mathsf{OPT}_{\mathsf{cnwrs}}(G_{\tilde{B}}, \Sigma_{\tilde{B}}) \leq O\left(\mathsf{OPT}_{\mathsf{cr}}(\tilde{B}) \cdot \mathrm{poly}(\Delta \log n)\right)$. We use the following lemma, whose proof is provided at the end of this section.

Lemma 1.8.1. $\sum_{B \in \mathcal{B}} \mathsf{OPT}_{\mathsf{cr}}(\tilde{B}) \leq O(\mathsf{OPT}_{\mathsf{cr}}(G))$.

From Lemma 1.8.1, the total number of edges in all graphs in $\left\{G_{\tilde{B}}\right\}_{\tilde{B} \in \tilde{\mathcal{B}}}$ is at most $O\left(\mathsf{OPT}_{\mathsf{cr}}(G) \cdot \mathrm{poly}(\Delta \log n)\right)$, and $\sum_{\tilde{B} \in \tilde{\mathcal{B}}} \mathsf{OPT}_{\mathsf{cnwrs}}(G_{\tilde{B}}, \Sigma_{\tilde{B}}) \leq O\left(\mathsf{OPT}_{\mathsf{cr}}(G) \cdot \mathrm{poly}(\Delta \log n)\right)$. We obtain a final instance (G', Σ) of the MCNwRS problem by letting G' be the disjoint union of all graphs in $\left\{G_{\tilde{B}}\right\}_{\tilde{B}\in\tilde{\mathcal{B}}}$, and letting $\Sigma = \bigcup_{\tilde{B}\in\tilde{\mathcal{B}}}\Sigma_{\tilde{B}}$. From the above discussion, $|E(G')| \leq O\left(\mathsf{OPT}_{\mathsf{cr}}(G)\cdot \text{poly}(\Delta\log n)\right)$, and, since solutions to all instances $(G_{\tilde{B}},\Sigma_{\tilde{B}})$ can be combined together to obtain a solution to instance (G',Σ), we get that $\mathsf{OPT}_{\mathsf{cnwrs}}(G',\Sigma) \leq O\left(\mathsf{OPT}_{\mathsf{cr}}(G)\cdot \text{poly}(\Delta\log n)\right)$. Assume now that we are given a solution to instance (G',Σ) of MCNwRS of value X. This solution immediately provides solutions $\varphi_{\tilde{B}}$ to all instances in $\left\{(G_{\tilde{B}},\Sigma_{\tilde{B}})\right\}_{\tilde{B}\in\tilde{\mathcal{B}}}$, such that, if we denote by $X_{\tilde{B}}$ the value of the solution $\varphi_{\tilde{B}}$, then $\sum_{\tilde{B}\in\tilde{\mathcal{B}}}X_{\tilde{B}} \leq X$. Using the algorithm from Section 1.3.3, we can compute, for each graph $\tilde{B}\in\tilde{\mathcal{B}}$, a drawing $\varphi_{\tilde{B}}$ of $\tilde{B}$ with at most $O\left(X_{\tilde{B}}+\mathsf{OPT}_{\mathsf{cr}}(\tilde{B})\right)\cdot \text{poly}(\Delta\log n)$ crossings.

We obtain the final drawing of G by "gluing" together the drawings $\left\{\varphi_{\tilde{B}}\right\}_{\tilde{B}\in\mathcal{B}}$ (this part is similar to the algorithm of [12]). In order to describe this step in more detail, we set up some notation. For each pseudo-block $B_0\in\mathcal{B}$, we denote by $\text{Desc}(B_0)$ the collection of all pseudo-blocks $B_1\in\mathcal{B}$, such that vertex $v(B_1)$ is the descendant of vertex $v(B_0)$ in the decomposition tree τ. We note that $B_0\in\text{Desc}(B_0)$. Consider now some pseudo-block $B_0\in\mathcal{B}$ and the drawing $\varphi_{\tilde{B}_0}$ of graph $\tilde{B}_0$ that we defined above. For simplicity of notation, we denote by $N(\tilde{B}_0)$ the total number of crossings in the drawing $\varphi_{\tilde{B}_0}$. We also denote by $N_0(\tilde{B}_0)$ the total number of crossings in the drawing $\varphi_{\tilde{B}_0}$ in which the fake parent-edge $e^*_{\tilde{B}_0}$ participates, and we denote by $N_1(\tilde{B}_0) = N(\tilde{B}_0) - N_0(\tilde{B}_0)$ the total number of all other crossings in $\varphi_{\tilde{B}_0}$. The following lemma allows us to "glue" the drawings $\left\{\varphi_{\tilde{B}}\right\}_{\tilde{B}\in\tilde{\mathcal{B}}}$ together. Its proof is provided at the end of this section.

Lemma 1.8.2. *There is an efficient algorithm, that, given a pseudo-block $B_0\in\mathcal{B}$, and drawings $\left\{\varphi_{\tilde{B}}\right\}_{B\in\text{Desc}(B_0)}$, computes a drawing $\hat{\varphi}_{B_0}$ of graph $B_0\cup\left\{e^*_{\tilde{B}_0}\right\}$ (if $B_0=G$ and edge $e^*_{\tilde{B}_0}$ is undefined, then $\hat{\varphi}_{B_0}$ is a drawing of B_0), such that the following hold:*

- *edges of $E''\cap E(B_0)$ do not participate in any crossings in $\hat{\varphi}_{B_0}$;*
- *the total number of crossings in $\hat{\varphi}_{B_0}$ in which the fake edge $e^*_{\tilde{B}_0}$ participates is at most $4\Delta N_0(\tilde{B}_0)$; and*

- *the total number of all other crossings in $\hat{\varphi}_{B_0}$ is at most* $16\Delta^2\left(\sum_{B\in \mathrm{Desc}(B_0)} N(\tilde{B}) - N_0(\tilde{B}_0)\right)$.

By combining these drawings using Lemma 1.8.2, we obtain a drawing of G, whose number of crossings is bounded by:

$$O(\Delta^2)\cdot \sum_{\tilde{B}\in\tilde{\mathcal{B}}}\left(X_{\tilde{B}}+\mathsf{OPT}_{\mathsf{cr}}(\tilde{B})\right)\cdot \mathrm{poly}(\Delta\log n)\le O\Bigg((X+\mathsf{OPT}_{\mathsf{cr}}(G))\cdot \mathrm{poly}(\Delta\log n)\Bigg).$$

1.8.1 Proof of Lemma 1.8.1

Let φ^* be the optimal drawing of G, so $\mathrm{cr}(\varphi^*)=\mathsf{OPT}_{\mathsf{cr}}(G)$. We assume w.l.o.g. that every pair of edges crosses at most once in φ^*, and for every edge, its image does not cross itself in φ^*. We now define, for each graph $\tilde{B}\in\tilde{\mathcal{B}}$, a drawing $\psi_{\tilde{B}}$ in the plane, using the drawing φ^*. Consider any graph $\tilde{B}\in\tilde{\mathcal{B}}$. Note that, if $\tilde{B}$ is isomorphic to K_3, then it has a planar drawing, so we let $\psi_{\tilde{B}}$ be that planar drawing. Assume now that $\tilde{B}$ is not isomorphic to K_3, so $\tilde{B}\in\tilde{\mathcal{B}}^*$.

In order to obtain the drawing $\psi_{\tilde{B}}$ of $\tilde{B}$, we start from the drawing φ^* of G, and delete from it all edges and vertices, except for the vertices and the real edges of $\tilde{B}$, and all vertices and edges participating in the paths in $\mathcal{P}_{\tilde{B}}$. We partition all crossings of the resulting drawing into five sets. Set $\chi_1(\tilde{B})$ contains all crossings (e_1,e_2), where both e_1 and e_2 are real edges of $\tilde{B}$. Set $\chi_2(\tilde{B})$ contains all crossings (e_1,e_2), where e_1 is a real edge of $\tilde{B}$, and e_2 lies on some path $P_{\tilde{B}}(e)$, where e is a fake edge of $\tilde{B}$. Set $\chi_3(\tilde{B})$ contains all crossings (e_1,e_2), where $e_1\in P_{\tilde{B}}(e)$, $e_2\in P_{\tilde{B}}(e')$; both e and e' are fake edges of $\tilde{B}$ (where possibly $e=e'$); and neither of these edges is the parent-edge $e^*_{\tilde{B}}$. Set $\chi_4(\tilde{B})$ contains all crossings (e_1,e_2), where $e_1\in P_{\tilde{B}}(e)$, $e_2\in P_{\tilde{B}}(e')$, such that e,e' are both fake edges of $\tilde{B}$, and exactly one of these edges is the parent-edge $e^*_{\tilde{B}}$. Lastly, set $\chi_5(\tilde{B})$ contains all crossings (e_1,e_2), where both $e_1,e_2\in P_{\tilde{B}}(e^*_{\tilde{B}})$. We obtain the final drawing $\psi_{\tilde{B}}$ of $\tilde{B}$ from the current drawing, by suppressing all inner vertices on the paths of $\tilde{\mathcal{P}}_{\tilde{B}}$. Additionally, if the image of the edge $e^*_{\tilde{B}}$ crosses itself, then we remove all loops to ensure that this does not happen. Clearly,

$\mathrm{cr}(\psi_{\tilde{B}}) \leq \sum_{i=1}^{4} |\chi_i(\tilde{B})|$.

We denote, for all $1 \leq i \leq 4$, $\chi_i = \bigcup_{\tilde{B} \in \tilde{\mathcal{B}}^*} \chi_i(\tilde{B})$ (we view χ_i as a multiset, so a crossing (e_1, e_2) that belongs to several sets $\chi_i(\tilde{B})$ is added several times to χ_i). It is now enough to show that $\sum_{i=1}^{4} |\chi_i| \leq O(\mathrm{cr}(\varphi^*))$.

Consider some crossing (e_1, e_2) in φ^*. Observe that this crossing may lie in set $\chi_1(\tilde{B})$ for a graph $\tilde{B} \in \tilde{\mathcal{B}}^*$ only if both e_1 and e_2 are real edges of $\tilde{B}$. Since each edge of G may belong to at most one graph in $\tilde{\mathcal{B}}$ as a real edge, crossing (e_1, e_2) appears at most once in χ_1. Therefore, $|\chi_1| \leq \mathrm{cr}(\varphi^*)$. Notice that crossing (e_1, e_2) may lie in set $\chi_2(\tilde{B})$ for a graph $\tilde{B} \in \tilde{\mathcal{B}}^*$ only if either e_1 or e_2 are real edges of $\tilde{B}$. Therefore, using the same reasoning as before, crossing (e_1, e_2) may appear at most twice in χ_2, and so $|\chi_2| \leq 2\,\mathrm{cr}(\varphi^*)$. Assume now that $(e_1, e_2) \in \chi_3(\tilde{B})$, for some graph $\tilde{B} \in \tilde{\mathcal{B}}^*$. Then there are fake edges e, e' in $\tilde{B}$, neither of which is the parent-edge $e^*_{\tilde{B}}$, such that $e_1 \in P_{\tilde{B}}(e)$ and $e_2 \in P_{\tilde{B}}(e')$ (where it is possible that $e = e'$). Since each edge of G may belong to at most 6 paths in $\mathcal{P}$, we get that crossing (e_1, e_2) may appear at most 6 times in χ_3, so $|\chi_3| \leq 6\,\mathrm{cr}(\varphi^*)$. Lastly, assume that $(e_1, e_2) \in \chi_4(\tilde{B})$, for some graph $\tilde{B} \in \tilde{\mathcal{B}}^*$. Then there are fake edges e, e' in $\tilde{B}$, exactly one of which is the parent-edge $e^*_{\tilde{B}}$, such that $e_1 \in P_{\tilde{B}}(e)$ and $e_2 \in P_{\tilde{B}}(e')$. Since each edge of G may belong to at most 6 paths in $\mathcal{P}$, we get that crossing (e_1, e_2) may appear at most 12 times in χ_4, so $|\chi_4| \leq 12\,\mathrm{cr}(\varphi^*)$.

Overall, we get that $\sum_{\tilde{B} \in \tilde{\mathcal{B}}} \mathsf{OPT}_{\mathsf{cr}}(\tilde{B}) \leq \sum_{\tilde{B} \in \tilde{\mathcal{B}}} \mathrm{cr}(\psi_{\tilde{B}}) \leq \sum_{i=1}^{4} |\chi_i| \leq O(\mathsf{OPT}_{\mathsf{cr}}(G))$.

1.8.2 Proof of Lemma 1.8.2

The proof is by induction on the length of the longest path from vertex $v(B_0)$ to a leaf vertex of τ that is a descendant of $v(B_0)$ in τ. The base case is when $v(B_0)$ is a leaf vertex of τ. In this case, graph $\tilde{B}_0$ is exactly $B_0 \cup \left\{ e^*_{\tilde{B}_0} \right\}$, and we set $\hat{\varphi}_{B_0} = \varphi_{\tilde{B}_0}$. From the definition of the set E'' of edges, no edges of $E'' \cap E(B_0)$ participate in crossings in the resulting drawing; the number of crossings in which edge $e^*_{\tilde{B}_0}$ participates is $N_0(\tilde{B}_0)$, and the total number of all other crossings in $\hat{\varphi}_{B_0}$ is $N(\tilde{B}_0) - N_0(\tilde{B}_0)$. Therefore, drawing $\hat{\varphi}_{B_0}$ has all required properties.

Next, we consider an arbitrary pseudo-block $B_0 \in \mathcal{B}$, where $v(B_0)$ is not a leaf of τ. Let $B_1, \ldots, B_r$ be pseudo-blocks whose corresponding vertices $v(B_i)$ are the children of $v(B_0)$ in tree τ. We denote, for all $1 \leq i \leq r$, the endpoints of the block B_i by (x_i, y_i), so that the edge $e^*_{\tilde{B}_i}$ (the fake parent-edge of $\tilde{B}_i$) connects x_i to y_i. Note that graph $\tilde{B}_0$ must also contain an edge $e_i = (x_i, y_i)$. Since parallel edges are not allowed in graph $\tilde{B}_0$, edge e_i may be a real or a fake edge of $\tilde{B}_0$, and moreover, it is possible that for $1 \leq i < j \leq r$, $e_i = e_j$. We use the induction hypothesis in order to compute, for all $1 \leq i \leq r$, a drawing $\hat{\varphi}_{B_i}$ of $B_i \cup \left\{e^*_{\tilde{B}_i}\right\}$ with the required properties.

We denote, for all $1 \leq i \leq r$, by G_i the multi-graph $\tilde{B}_0 \cup \left(\bigcup_{i'=1}^{i-1} B_{i'}\right)$. We start with the drawing $\varphi_0 = \varphi_{\tilde{B}_0}$ of $\tilde{B}_0$, and then perform r iterations. The input to the ith iteration is a drawing φ_{i-1} of graph G_{i-1}, and the output is a drawing φ_i of G_i.

We now describe the execution of the ith iteration. Our starting point is the input drawing φ_{i-1} of graph G_{i-1} on the sphere, and the drawing $\hat{\varphi}_{B_i}$ of $B_i \cup e^*_{\tilde{B}_i}$. Recall that x_i, y_i are the endpoints of the block B_i, and graph G_{i-1} contains the edge $e_i = (x_i, y_i)$. For convenience of notation, we denote the fake parent-edge of $\tilde{B}_i$, whose endpoints are also x_i, y_i by e'_i. Note that there must be some point t'_i on the image of the edge e'_i in $\hat{\varphi}_{B_i}$, such that the segment σ' of the image of e'_i between x_i and t'_i lies on the boundary of a single face in the drawing $\hat{\varphi}_{B_i}$. Let F' denote this face. We view $\hat{\varphi}_{B_i}$ as a drawing on the plane, whose outer face is F'. Similarly, there is a point t_i on the image of the edge e_i in the drawing φ_{i-1} such that the segment σ of the image of e_i between x_i and t_i lies on the boundary of a single face in φ_{i-1}; we denote that face by F. We view φ_{i-1} as a drawing in the plane, where face F is the outer face. Next, we superimpose the drawings φ_{i-1} and $\hat{\varphi}_{B_i}$ in the plane, such that the two resulting drawings are disjoint, except that the image of the vertex x_i is unified in both drawings, and the faces F and F' of the two drawings correspond to the outer face of this new drawing, that we denote by F^*. We add a curve γ to this new drawing, connecting the images of t_i and t'_i, such that γ does not intersects any parts of the current drawing, except for its endpoints. The image of the vertex y_i in the new drawing becomes the image of y_i in φ_{i-1}. Let E_i be the set of all edges of $E(B_i)$ that are incident to y_i. In order to complete the drawing φ_i of G_i, we need to define the drawings of the edges in E_i. Consider any such edge $e = (a, y_i)$. In order to obtain a new drawing of e, we start with the drawing of e in $\hat{\varphi}_{B_i}$, that connects the image of a to the original image of y_i in $\hat{\varphi}_{B_i}$. Next, we follow along the image of the edge e'_i in $\hat{\varphi}_{B_i}$, until the point t'_i. Next, we follow the curve γ, connecting point t'_i to point t_i, and lastly, we follow the image of the edge e_i in the drawing φ_{i-1}, from point t_i to the image of the vertex y_i. We can do so in a way that ensures that the images of the edges in E_i do not cross each other. This defines the final drawing φ_i of the graph G_i. We now analyze its crossings.

Consider any crossing (e, e') in φ_i. We say that it is an *old* crossing iff: (i) crossing (e, e') is present in the drawing φ_{i-1}; or (ii) crossing (e, e') is present in the drawing $\hat{\varphi}_{B_i}$, and neither of the edges e, e' is e'_i. All other crossings in φ_i are called *new crossings*. Note that each such new crossing must involve exactly one edge from E_i. Specifically, for every crossing $(e'_i, \hat{e})$ in $\hat{\varphi}_{B_i}$, in which the edge e'_i participates, we introduce $|E_i|$ new crossings of $\hat{e}$ with edges of E_i. Notice that, if e'_i participates in any crossings in $\hat{\varphi}_{B_i}$, then $E_i \subseteq E''$. The number of such new crossings is then bounded, from the induction hypothesis, by $4\Delta N_0(\tilde{B}_i) \cdot |E_i| \leq 4\Delta^2 N_0(\tilde{B}_i)$. Additionally, for every crossing $(e_i, \hat{e})$ in φ_{i-1}, in which the edge e_i participates, we introduce $|E_i|$ new crossings of $\hat{e}$. We say that crossing $(e_i, \hat{e})$ is *responsible* for these new crossings. Our algorithm ensures that crossing $(e_i, \hat{e})$ may only be present in the drawing φ_{i-1} if edge e_i participated in crossings in $\varphi_{\tilde{B}_0}$. In this case, we are guaranteed that $E_i \subseteq E''$. Therefore, we ensure that all real edges that participate in the new crossings belong to E''. This completes the description of the ith iteration.

Let $\hat{\varphi} = \varphi_r$ be the final drawing of the graph G_r that we obtain. We now bound the number of crossings in $\hat{\varphi}$. We partition the crossings of $\hat{\varphi}$ into three sets: set χ_1 contains all crossings

(e,e'), where, for some $1 \leq i \leq r$, $e,e' \in E(B_i)$. Set χ_2 contains all crossings (e,e'), where $e = e^*_{\tilde{B}_0}$; and set χ_3 contains all other crossings.

For all $1 \leq i \leq r$, let $\chi_1^i \subseteq \chi_1$ be the set of all crossings (e,e'), where $e,e' \in E(B_i)$. From the above discussion, if (e,e') is a crossing in χ_1^i, then either it was present in $\hat{\varphi}_{B_i}$, or it is one of the new crossings. The number of crossings of the former type is bounded, from the induction hypothesis, by $16\Delta^2 \left(\sum_{B \in \mathrm{Desc}(B_i)} N(\tilde{B}) - N_0(\tilde{B}_i)\right)$, while the number of crossings of the latter type is bounded by $4\Delta^2 N_0(\tilde{B}_i)$. Therefore, altogether, $|\chi_1^i| \leq 16\Delta^2 \left(\sum_{B \in \mathrm{Desc}(B_i)} N(\tilde{B})\right)$.

We now proceed to bound the number of crossings in χ_2. Consider any crossing (e,e') in the drawing $\varphi_{\tilde{B}_0}$ of $\tilde{B}_0$, where $e = e^*_{\tilde{B}_0}$. If e' is a real edge of $\tilde{B}_0$, then this crossing is present in χ_2. Otherwise, $e' = e_i$ for some $1 \leq i \leq r$. Then crossing (e,e') is responsible for $|E_i|$ new crossings in χ_2. It is then easy to verify that every crossing (e,e') of $\varphi_{\tilde{B}_0}$ with $e = e^*_{\tilde{B}_0}$ may be responsible for at most Δ crossings in χ_2, and so $|\chi_2| \leq \Delta N_0(\tilde{B}_0)$.

Lastly, we need to bound $|\chi_3|$. We partition set χ_3 of crossings into three subsets: set χ_3' contains all crossings (e,e'), where both e,e' are real edges of $\tilde{B}_0$. It is easy to verify that $|\chi_3'| \leq N_1(\tilde{B}_0)$. Set χ_3'' contains all crossings (e,e'), where e is a real edge of $\tilde{B}_0$, and $e' \in E(B_i)$, for some $1 \leq i \leq r$. In this case, drawing φ_{i-1} of G_{i-1} contained a crossing (e,e_i), that was charged for this new crossing, and the total charge to each such crossing (e,e_i) was at most $|E_i|$. It is then easy to verify that $|\chi_3''| \leq \Delta N_1(\tilde{B}_0)$. Lastly, set χ_3''' contains all remaining crossings (e,e'), where $e \in B_i, e' \in B_j$, for some $1 \leq i \neq j \leq r$. In this case, it is easy to verify that crossing (e_i,e_j) must have been present in drawing $\hat{\varphi}_{\tilde{B}_0}$ of $\tilde{B}_0$. Moreover, each such crossing (e_i,e_j) may be responsible for at most Δ^2 crossings in χ_3''', and so, overall $|\chi_3'''| \leq \Delta^2 N_1(\tilde{B}_0)$. We conclude that $|\chi_3| \leq 2\Delta^2 N_1(\tilde{B}_0)$.

We obtain the final drawing $\hat{\varphi}_{B_0}$ of $B_0 \cup \left\{e^*_{\tilde{B}_0}\right\}$ by deleting, from the drawing $\hat{\varphi}$, the images of all fake edges of $\tilde{B}_0$, except for the edge $e^*_{\tilde{B}_0}$. From the above discussion, the total number of crossings in $\hat{\varphi}_{B_0}$ in which the fake edge $e^*_{\tilde{B}_0}$ participates is at most $4\Delta N_0(\tilde{B}_0)$; and the total number of all other crossings in $\hat{\varphi}_{B_0}$ is at most $16\Delta^2 \left(\sum_{B \in \mathrm{Desc}(B_0)} N(\tilde{B}) - N_0(\tilde{B}_0)\right)$. Moreover, from the discussion above, if edge $e \in E(B_0)$ participates in a crossing in $\hat{\varphi}_{B_0}$, then $e \in E''$ must hold.

1.9 Missing Proofs in the First Part

1.9.1 Proof of Corollary 1.1.3

In this section, we provide the proof of Corollary 1.1.3 from Theorem 1.1.2 and 1.1.1. Suppose we are given a simple n-vertex graph G with maximum vertex degree Δ. We use the algorithm from Theorem 1.1.1 in order to compute an instance $I = (G', \Sigma)$ of MCNwRS, with $m = |E(G')| \leq O\left(\mathsf{OPT}_{\mathsf{cr}}(G) \cdot \mathrm{poly}(\Delta \cdot \log n)\right)$, and $\mathsf{OPT}_{\mathsf{cnwrs}}(I) \leq O\left(\mathsf{OPT}_{\mathsf{cr}}(G) \cdot \mathrm{poly}(\Delta \cdot \log n)\right)$. Notice that, since G is a simple graph, $\mathsf{OPT}_{\mathsf{cr}}(G) \leq |E(G)|^2 \leq n^4$, and $\Delta \leq n$. Therefore, $m = |E(G')| \leq \mathrm{poly}(n)$.

We use the algorithm from Theorem 1.1.2 to compute a solution to instance I of MCNwRS, such that, w.h.p., the number of crossings in the solution is bounded by $2^{O((\log m)^{7/8} \log\log m)} \cdot (\mathsf{OPT}_{\mathsf{cnwrs}}(I) + m)$. Lastly, using the algorithm from Theorem 1.1.1, we efficiently compute a drawing of graph G, with the number of crossings bounded by:

$$\begin{aligned}\left(2^{O((\log m)^{7/8} \log\log m)} \cdot (\mathsf{OPT}_{\mathsf{cnwrs}}(I) + m) + \mathsf{OPT}_{\mathsf{cr}}(G)\right) \cdot \mathrm{poly}(\Delta \log n)\\ \leq O\left(2^{O((\log n)^{7/8} \log\log n)} \cdot \mathrm{poly}(\Delta)\right) \cdot \mathsf{OPT}_{\mathsf{cr}}(G).\end{aligned}$$

1.9.2 Proof of Theorem 1.3.5

In order to simplify the notation, we denote the set $\hat{\mathcal{C}}$ of clusters by $\mathcal{C}$. Consider some cluster $C \in \mathcal{C}$. For every face $F \in \mathcal{F}_C$ of the drawing ψ_C of C, we let $\mathcal{H}_C(F) \subseteq \mathcal{C} \setminus \{C\}$ contain all clusters C' such that $F_C(C') = F$. We denote by $\mathcal{F}'_C \subseteq \mathcal{F}_C$ the set of all faces F with $\mathcal{H}_C(F) \neq \emptyset$.

The proof of the theorem a recursive algorithm. The base case is when, for every cluster $C \in \mathcal{C}$, $|\mathcal{F}'_C| = 1$. For each cluster C, let F_C be the unique face in $\mathcal{F}'_C$. Consider a drawing of the clusters in $\bigcup_{C \in \mathcal{C}} C$ in the plane, such that the drawing of each cluster C is identical to ψ_C, with the face F_C serving as the outer face of the drawing of C, and the images of all clusters are disjoint. Let φ' denote the resulting drawing of $\bigcup_{C\in\mathcal{C}} C$ in the plane, and let φ be the corresponding drawing on the sphere. Then it is easy to see that φ' defines a feasible solution for the input instance of the $\mathsf{Cluster\ Placement}$ problem.

Assume now that there is at least one cluster $C \in \mathcal{C}$, with $|\mathcal{F}'(C)| > 1$. For every face $F \in \mathcal{F}'(C)$, we define a new sub-instance of the current instance of the $\mathsf{Cluster\ Placement}$ problem, that only includes the clusters of $\mathcal{H}_C(F) \cup \{C\}$; the faces $F_{C_1}(C_2)$ for pairs $C_1, C_2 \in \mathcal{H}_C(F)$ of clusters are defined exactly as before. Notice that, if the original instance of the $\mathsf{Cluster\ Placement}$ problem had a feasible solution, then each resulting sub-problem must also have a feasible solution. We solve each such sub-problem recursively, and obtain, for every face $F \in \mathcal{F}'(C)$, a drawing φ_F of $\bigcup_{C'\in\mathcal{H}_C(F)\cup C} C'$ on the sphere. We let D_F denote a disc in this drawing that contains the images of all clusters if $\mathcal{H}_C(F)$ but is disjoint form the image of C. In order to obtain a solution to the original instance of the problem, we start with the drawing ψ_C of the cluster C on the sphere. For every face $F \in \mathcal{F}'(C)$, we copy the contents of the disc D_F in the drawing φ_F to the interior of the face F. Once we process every face $F \in \mathcal{F}'(C)$, we obtain a drawing of $\bigcup_{C'\in\mathcal{C}} C'$. Moreover, it is easy to verify that, if the original instance of the $\mathsf{Cluster\ Placement}$ problem had a feasible solution, then we obtain a feasible solution for this instance.

1.9.3 Proof of Lemma 1.3.8

It is enough to show that, for every face $F \in \mathcal{F}$, there is a solution φ^F to instance (G^F, Σ^F) of MCNwRS, such that, if we denote by χ^F the set of all crossings of edges in φ^F, then $\sum_{F\in\mathcal{F}} |\chi^F| \leq O(\mathsf{OPT}_{\mathsf{cr}}(G) \cdot \mathrm{poly}(\Delta \log n))$.

We now fix a face $F \in \mathcal{F}$ and construct a solution φ^F to instance (G^F, Σ^F) of MCNwRS. Our starting point is the drawing φ of graph G' given by Observation 1.3.6. We delete from φ the images of all vertices and edges, except for the vertices and edges of the clusters in $\mathcal{H}(F)$, and the images of the edges in E^F. Recall that, in the resulting drawing, no edge of

$\bigcup_{C\in\mathcal{H}(F)} E(C)$ may participate in crossings. Since, for every ordered pair $(C,C')\in\mathcal{H}(C)$ of clusters, the image of C' in the resulting drawing must be contained in the face $F_C(C')$, there must be a face F' in the resulting drawing that contains the images of every edge in E^F. Viewing face F' as the outer face of a planar drawing of the resulting graph, we can contract the image of each cluster $C\in\mathcal{H}(F)$ into a single point, that we view as the image of the corresponding vertex $v(C)$, without increasing the number of crossings. Therefore, we obtain a drawing $\tilde{\varphi}^F$ of the graph G^F on the sphere, and moreover:

$$\sum_{F\in\mathcal{F}} \mathrm{cr}(\tilde{\varphi}^F) \le \mathrm{cr}(\varphi) \le O(\mathsf{OPT}_{\mathsf{cr}}(G)\cdot \mathrm{poly}(\Delta\log n)).$$

Consider again some face $F\in\mathcal{F}$. Notice that the drawing $\tilde{\varphi}^F$ of G^F is not necessarily consistent with the rotation system Σ^F. Next, we modify the drawing $\tilde{\varphi}^F$ in order to make it consistent with Σ^F, while only introducing a small number of crossings. Recall that we have defined a set $\Gamma\subseteq V(G')$ of vertices called terminals – the set of all vertices that serve as endpoints of the edges in E''.

We process the vertices $v\in V(G^F)$ one-by-one. For every vertex $v\in V(G^F)$, we let $\eta(v)$ be a small disc around the drawing of v in $\tilde{\varphi}^F$. When vertex v is processed, we slightly modify the images of the edges of $\delta(v)$ in disc $\eta(v)$, so that the circular order in which the edges of $\delta(v)$ enter v becomes identical to $\mathcal{O}_v$.

Consider now some vertex $v=v(C)\in V(G^F)$. Assume first that $C\in\mathcal{C}_1$. In this case, $|\delta(v)|\le \mathrm{poly}(\Delta\log n)$. For every edge $e\in\delta(v)$, we replace the segment of the drawing of e in disc $\eta(v)$, so that all resulting curves enter the image of the vertex v in the order $\mathcal{O}_v$, and every pair of curves intersects at most once. This introduces at most $\mathrm{poly}(\Delta\log n)$ new crossings.

Assume now that $C\in\mathcal{C}_2$. Let $\mathcal{O}'_v$ be the order in which the images of edges of $\delta(v)$ enter v in the current drawing $\tilde{\varphi}^F$. Recall that we have defined a cycle $K^F(C)\subseteq C$, which is the intersection of the cluster C and the boundary of the face F, and $\Gamma^F(C)\subseteq\Gamma(C)$ is the set of terminals that appear on $K^F(C)$. Recall that we have defined an ordering $\tilde{\mathcal{O}}^F(C)$ of the terminals in $\Gamma^F(C)$ to be the circular order of the terminals of $\Gamma^F(C)$ along the cycle $K^F(C)$, and the ordering $\mathcal{O}_v$ of $\delta(v)$ was defined based on $\tilde{\mathcal{O}}^F(C)$. Since the edges of C may not participate in crossings in φ, and since the drawing of C in φ is identical to ψ_C, for every terminal $t \in \Gamma^F(C)$, the edges of $\delta(v)$ that are incident to t appear consecutively in the ordering $\mathcal{O}'_v$ (and from the definition, they also appear consecutively in the ordering $\mathcal{O}_v$). The orderings of edges of $\delta(v)$ that are incident to different terminals in $\mathcal{O}_v$ and $\mathcal{O}'_v$ must both be consistent with $\tilde{\mathcal{O}}_v$. Therefore, the only difference between the orderings $\mathcal{O}_v$ and $\mathcal{O}'_v$ is that for every terminal $t \in \Gamma^F(C)$, the edges of $\delta(v)$ that are incident to t may appear in different orders in $\mathcal{O}_v$ and $\mathcal{O}'_v$. Consider the small disc $\eta(v)$ around the vertex v in the current drawing $\tilde{\varphi}^F$ of G^F. We can assume that this disc is small enough so it does not contain any crossings. For every edge $e \in \delta(v)$, let p_e be the point that is the intersection of the current image of e and the boundary of the disc $\eta(v)$ in $\tilde{\varphi}^F$. Then for every terminal $t \in \Gamma^F(C)$, the points p_e corresponding to the edges of $\delta(v)$ that are incident to t appear consecutively on the boundary of $\eta(v)$. We rearrange the images of all edges of $\delta(v)$ that are incident to t inside $\eta(v)$, so that they enter v in the order consistent with $\mathcal{O}_v$. This introduces, for every terminal $t \in \Gamma^F(C)$, at most $O(\Delta^2)$ new crossings. Once we process all vertices of G^F, we obtain the final drawing φ^F of G^F that is consistent with the rotation system Σ^F. We now bound the total number of new crossings that this procedure has introduced.

Consider a cluster C. Note that for every terminal $t \in \Gamma(C)$, there can be at most Δ faces $F \in \mathcal{F}$, such that t lies on the boundary of F in drawing $\tilde{\varphi}$ of $\bigcup_{C \in \mathcal{C}} C$. Therefore, there are at most $\Delta|\Gamma(C)|$ pairs (t, F), where $t \in \Gamma(C)$ and $F \in \mathcal{F}$, such that t lies on the boundary of F. We denote the set of all such pairs for cluster C by $\Pi(C)$.

If C is a type-1 cluster, then the total increase in the number of crossings due to rearranging edges entering vertex $v(C)$ in all graphs $\left\{G^F\right\}_{F \in \mathcal{F}}$ is at most $O(|\Pi(C)|^2\Delta^2) \leq O(\Delta^4|\Gamma(C)|^2) \leq O(\text{poly}(\Delta \log n))$.

If C is a type-2 cluster, then every pair $(t, F) \in \Pi(C)$ contributes at most Δ^2 crossings (by rearranging the images of edges of $\delta(v)$ in $\tilde{\varphi}^F$ that are incident to t).

Altogether, the number of new crossings is bounded by:

$$\begin{aligned} O(|\Gamma|\Delta^3) + O(|\mathcal{C}_1| \operatorname{poly}(\Delta \log n)) &\leq O(|\Gamma| \operatorname{poly}(\Delta \log n)) \\ &\leq O(|E''| \operatorname{poly}(\Delta \log n)) \\ &\leq O(\mathsf{OPT}_{\mathsf{cr}}(G) \operatorname{poly}(\Delta \log n), \end{aligned}$$

and so the total number of crossings in all drawings in $\left\{\varphi^F\right\}_{F \in \mathcal{F}}$ is $O(\mathsf{OPT}_{\mathsf{cr}}(G) \operatorname{poly}(\Delta \log n))$.

1.9.4 *Proof of Lemma 1.4.4*

We assume w.l.o.g. that graph G is 2-connected, as otherwise we can prove the theorem for each of its super-blocks $Z \in \mathcal{Z}(G)$ separately. We denote by $\mathcal{B} = \mathcal{B}(G)$ the block decomposition of G, $\tilde{\mathcal{B}} = \left\{\tilde{B} \mid B \in \mathcal{B}\right\}$, and we let $\tilde{\mathcal{B}}^* \subseteq \tilde{\mathcal{B}}$ contain all graphs $\tilde{B} \in \tilde{\mathcal{B}}$ that are not isomorphic to K_3. We denote by $\tau = \tau(\mathcal{B})$ the decomposition tree corresponding to the block decomposition $\mathcal{B}$ of G. If vertex $v(B_1)$ is a child of vertex $v(B_2)$ in tree τ, then we say that B_1 is a *child block* of B_2. For a block $B \in \mathcal{B}$, we also denote by $\mathrm{Desc}(B)$ the set of all *descendant blocks* of B. The set $\mathrm{Desc}(B)$ contains all blocks B_1 where vertex $v(B_1)$ is a descendant of vertex $v(B)$ in the tree τ. We note that the set $\mathrm{Desc}(B)$ also contains the block B.

Consider a block $B \in \mathcal{B}$. For convenience, we call the fake edge $e^*_{\tilde{B}}$ connecting the endpoints of B (if it exists) a *bad* fake edge, and all other fake edges of $\tilde{B}$ are called good fake edges. We define a set $A'_{\tilde{B}}$ of fake edges as follows: if B is isomorphic to K_3, then $A'_{\tilde{B}} = \emptyset$, and otherwise, $A'_{\tilde{B}}$ contains all good fake edges of $\tilde{B}$. Consider now some good fake edge $e = (u, v) \in A'_{\tilde{B}}$. Then there is some child block B_1 of B with endpoints u, v. A *valid embedding* of the fake edge $e = (u, v) \in A'_{\tilde{B}}$ is a path $P(e)$ that connects u to v, is internally disjoint from $\tilde{B}$, and is contained in a block B_1 that is a child block of B, whose endpoints are (u, v). A *valid embedding* of the set $A'_{\tilde{B}}$ of fake edges is a collection $\mathcal{Q}(B) = \left\{P(e) \mid e \in A'_{\tilde{B}}\right\}$ of paths, were for each edge $e \in A'_{\tilde{B}}$, path $P(e)$ is a valid embedding of e. Note that from the definition, we are guaranteed that the paths in $\mathcal{Q}(B)$ are internally disjoint. The following lemma is central to the proof of Lemma 1.4.4.

Lemma 1.9.1. *There is an efficient algorithm, that, given a block $B \in \mathcal{B} \setminus \{G\}$, computes, for every descendant-block $B_1 \in \mathrm{Desc}(B)$, a valid embedding $\mathcal{Q}(B_1)$ of the set $A'_{\tilde{B}_1}$ of fake edges, and additionally a collection $\mathcal{P}_1(B)$ of 6 paths in B, connecting the endpoints of B, such that, if we denote by $\mathcal{P}_2(B) = \bigcup_{B_1 \in \mathrm{Desc}(B)} \mathcal{Q}(B_1)$, then the paths in $\mathcal{P}_1(B) \cup \mathcal{P}_2(B)$ cause congestion at most 6 in B.*

We note that, from the definition of valid embeddings, all paths in $\mathcal{P}_2(B)$ must be contained in B.

We prove Lemma 1.9.1 below, after we complete the proof of Lemma 1.4.4 using it. Recall that graph G itself is a block in the decomposition $\mathcal{B}$. Let $B_1, \ldots, B_r$ be the child blocks of G. We apply the algorithm from Lemma 1.9.1 to each such block B_i separately, obtaining the sets $\mathcal{P}_1(B_i), \mathcal{P}_2(B_i)$ of paths. Let $\mathcal{Q}^* = \bigcup_{i=1}^{r} \mathcal{P}_2(B_i)$. Then set $\mathcal{Q}^*$ contains, for every block $B \in \mathcal{B} \setminus \{G\}$, a set $\mathcal{Q}(B)$ of paths that defines a valid embedding of the set $A'_{\tilde{B}}$ of fake edges of $\tilde{B}$. For all $1 \leq i \leq r$, let (x_i, y_i) be the endpoints of the block B_i. We embed the fake edge (x_i, y_i) of $A'_{\tilde{G}}$ into any of the 6 paths $P(x_i, y_i) \in \mathcal{P}_1(B_i)$. We then set $\mathcal{Q}(G) = \{P(x_i, y_i) \mid 1 \leq i \leq r\}$. Note that $\mathcal{Q}(G)$ is a valid embedding of the good fake edges of G. Adding the paths in $\mathcal{Q}(G)$ to the set $\mathcal{Q}^*$, we obtain a set of paths in G, that cause edge-congestion at most 6, and contain, for every pseudo-block $B \in \mathcal{B}$, a valid embedding of the set $A'_{\tilde{B}}$ of fake edges. Lastly, consider any block $B \in \mathcal{B} \setminus \{G\}$. Let (x, y) be the endpoints of B, and let B^c be its complement block. Then B^c contains a path connecting x to y. We let the embedding $P(e^*_{\tilde{B}})$ of the bad fake edge $e^*_{\tilde{B}}$ be any path in B^c that connects x to y. We then set $\mathcal{P}_{\tilde{B}} = \mathcal{Q}(B) \cup \left\{P(e^*_{\tilde{B}})\right\}$. Observe that, from the definition of valid embeddings of bad fake edges, all paths in $\mathcal{P}_{\tilde{B}}$ are mutually internally disjoint, and they are internally disjoint from $\tilde{B}$. As discussed above, the paths in set $\mathcal{P} = \bigcup_{\tilde{B} \in \tilde{\mathcal{B}}^*(G)} \left(\mathcal{P}_{\tilde{B}} \setminus \left\{P_{\tilde{B}}(e^*_{\tilde{B}})\right\}\right)$ cause congestion at most 6 in G. Therefore, in order to complete the proof of Lemma 1.4.4, it is now enough to prove Lemma 1.9.1.

Proof of Lemma 1.9.1. The proof is by induction on the length of the longest path from $v(B)$ to a leaf vertex of τ that is a descendant of $v(B)$ in τ. The base case is when $v(B)$ is a leaf vertex of τ. Let (x, y) denote the endpoints of B. Observe that in this case, $\tilde{B}$ is obtained from block B by adding the bad fake edge $e^*_{\tilde{B}}$ to it, and this is the only fake edge in $\tilde{B}$. In particular, $A'_{\tilde{B}} = \emptyset$, so we can set $\mathcal{Q}(B) = \emptyset$. Block B must contain at least one path P connecting x to y. We let $\mathcal{P}_1(B)$ contain 6 copies of this path. Setting $\mathcal{P}_2(B) = \emptyset$, we get valid sets $\mathcal{P}_1(B), \mathcal{P}_2(B)$ of paths for B, with $\mathcal{P}_1(B) \cup \mathcal{P}_2(B)$ causing edge-congestion at most 6 in B.

Consider now an arbitrary block $B \in \mathcal{B} \setminus \{G\}$, such that $v(B)$ is not a leaf of τ, and let $B_1, \ldots, B_r$ be its child blocks. Using the induction hypothesis, we compute, for all $1 \leq i \leq r$, the sets $\mathcal{P}_1(B_i), \mathcal{P}_2(B_i)$ of paths that are contained in B_i, such that $\mathcal{P}_1(B_i) \cup \mathcal{P}_2(B_i)$ cause edge-congestion at most 6 in B_i. We now consider three cases.

Case 1. $\tilde{B} = K_3$. In this case, $A'_{\tilde{B}} = \emptyset$, and so $\mathcal{Q}(B) = \emptyset$ is a valid embedding of the edges in $A'_{\tilde{B}}$. We set $\mathcal{P}_2(B) = \bigcup_{i=1}^{r} \mathcal{P}_2(B_i)$. Clearly, set $\mathcal{P}_2(B)$ contains, for each block $B^* \in \mathrm{Desc}(B)$, a valid embedding $\mathcal{Q}(B^*)$ of the edges of $A'_{\tilde{B}^*}$. It now remains to define a set $\mathcal{P}_1(B)$ of 6 paths connecting the endpoints of B. Let (x, y) denote the endpoints of B. Then there is a path P in $\tilde{B}$, that is disjoint from the bad fake edge $e^*_{\tilde{B}}$, connecting x to y. This path contains two edges, that we denote by e_1 and e_2. Initially, we let $\mathcal{P}_1(B)$ contain six copies of the path P, that we denote by $P_1, \ldots, P_6$. Assume first that $e_1 = (x_1, y_1)$ is a fake edge, and assume w.l.o.g. that the child block B_1 of B has endpoints x_1 and y_1. Then we replace, for all $1 \leq i \leq 6$, the edge e_1 on path P_i by the ith path in set $\mathcal{P}_1(B_1)$ (recall that this path connects x_1 to y_1 in B_1). If e_2 is a fake edge, then we proceed similarly. As a result, set $\mathcal{P}_1(B)$ now contains 6 paths that are contained in B, each of which connects x to y. Moreover, it is easy to verify that the paths in $\mathcal{P}_1(B) \cup \mathcal{P}_2(B)$ cause edge-congestion at most 6 in B.

Case 2. $\tilde{B} \neq K_3$, and block B has a single child-block. We denote by B_1 the child block of B. Let (x, y) be the endpoints of B, and let (x_1, y_1) be the endpoints of B_1. Let $e_1 = (x_1, y_1)$ be the unique fake edge in $A'_{\tilde{B}}$. We let the embedding of this edge $P(e_1)$ be any of the 6 paths in $\mathcal{P}_1(B_1)$. We then let $\mathcal{Q}(B) = \{P(e_1)\}$ be a valid embedding of the edges of $A'_{\tilde{B}}$, and $\mathcal{P}_2(B) = \mathcal{Q}(B) \cup \mathcal{P}_2(B_1)$. Clearly, set $\mathcal{P}_2(B)$ contains, for each block $B^* \in \mathrm{Desc}(B)$, a valid embedding $\mathcal{Q}(B^*)$ of the edges of $A'_{\tilde{B}^*}$. It now remains to define a set $\mathcal{P}_1(B)$ of 6 paths connecting the endpoints of B. Since graph $\tilde{B}$ is 3-connected, there are at least three internally disjoint paths connecting x to y in $\tilde{B}$. Since graph $\tilde{B}$ contains only two fake edges, at least one of these paths, that we denote by P, does not contain fake edges. We then let $\mathcal{P}_1(B)$ contain 6 copies of the path P. Clearly, the paths in $\mathcal{P}_1(B) \cup \mathcal{P}_2(B)$ cause edge-congestion at most 6 in B.

Case 3. $\tilde{B} \neq K_3$, and block B has at least two child-blocks. Let (x, y) be the endpoints of B, and, for all $1 \leq i \leq r$, let (x_i, y_i) be the endpoints of the child block B_i. We also denote by $e_i = (x_i, y_i)$ the fake edge in $A'_{\tilde{B}}$ connecting the endpoints of e_i (if it exists). For each block B_i, we let $P_i^* \in \mathcal{P}_1(B_i)$ be an arbitrary path in $\mathcal{P}_1(B_i)$, and we let the embedding $P(e_i)$ of fake edge e_i (if it exists) be P_i^*. We then define $\mathcal{Q}(B) = \left\{P(e_i) \mid e_i \in A'_{\tilde{B}}\right\}$. It is immediate to verify that $\mathcal{Q}(B)$ is a valid embedding of the fake edges in $A'_{\tilde{B}}$. We then set $\mathcal{P}_2(B) = \mathcal{Q}(B) \cup (\bigcup_{i=1}^r \mathcal{P}_2(B_i))$, so set $\mathcal{P}_2(B)$ contains, for each block $B^* \in \mathrm{Desc}(B)$, a valid embedding $\mathcal{Q}(B^*)$ of the edges of $A'_{\tilde{B}^*}$. It now remains to define a set $\mathcal{P}_1(B)$ of 6 paths connecting the endpoints of B.

Since graph $\tilde{B}$ is 3-connected, it contains at least three internally disjoint paths connecting x to y. At least two of these paths, that we denote by P and P', are disjoint from the bad fake edge $e^*_{\tilde{B}}$. We let $\mathcal{P}_1(B) = \left\{P_1, P_2, P_3, P'_1, P'_2, P'_3\right\}$, where initially, paths P_1, P_2, P_3 are copies of P, and paths P'_1, P'_2, P'_3 are copies of P'. Next, we consider every fake edge of P and P' one-by-one. Let $e_i \in P$ be a fake edge on path P, so $e_i = (x_i, y_i)$, and the child block B_i of B has endpoints x_i, y_i. Recall that $\mathcal{P}_1(B_i)$ contains 6 paths connecting x_i to y_i; one of these paths has been used as P^*_i, and the remaining five paths have not been used yet. For each $1 \leq j \leq 3$, we replace, on path P_j, the edge e_i, with the jth path in $\mathcal{P}_1(B_i) \setminus \left\{P^*_i\right\}$. Once we process every fake edge on path P and on path P' in this fashion, we obtain a final set $\mathcal{P}_1$ of 6 paths connecting x to y in graph B. Moreover, it is easy to verify that the paths in $\mathcal{P}_1(B) \cup \mathcal{P}_2(B)$ cause edge-congestion at most 6. □

1.9.5 Proof of Lemma 1.7.1

In order to simplify the notation, we denote H' by H.

Let $\hat{H}$ be a graph that is obtained from H, by adding a new vertex s to it, and connecting it to every vertex $v \in \Gamma$. We set edge capacities in graph $\hat{H}$ as follows. Each edge in $\{(s, v) \mid v \in \Gamma\}$ has capacity 1, and each edge $e \in E(H)$ has capacity $\mathrm{cong}_{\mathcal{Q}}(e)$. Next, we compute maximum flow F from s to u in graph $\hat{H}$. Since all edge capacities are integral, we can ensure that the flow F is integral as well. It is immediate to verify that the value of the maximum s-u flow in $\hat{H}$ is $|\Gamma|$. Moreover, we can ensure that for every edge $e = (u', v')$, the flow is only sent in one direction (either from u' to v' or from v' to u', but not both). We can also ensure that the flow is acyclic. Using the standard flow-path decomposition of the flow F, and deleting the first edge from each resulting flow-path, we obtain a set $\tilde{\mathcal{Q}} = \left\{\tilde{Q}_v \mid v \in \Gamma\right\}$ of directed paths in graph $\hat{H}$, where each path $\tilde{Q}_v$ connects v to u. Moreover, for every edge $e \in E(\hat{H})$, $\mathrm{cong}_{\tilde{\mathcal{Q}}}(e) \leq \mathrm{cong}_{\mathcal{Q}}(e)$.

Let H^* be the directed graph that is obtained by taking the union of the paths in $\tilde{\mathcal{Q}}$, whose edges are directed towards the edge u; for every edge $e \in E(H)$, we add $\mathrm{cong}_{\tilde{\mathcal{Q}}}(e)$ copies of

e to this graph. Notice that H^* is a Directed Acyclic Graph, and so there is an ordering $\mathcal{O}$ of its vertices, such that, for every pair $x, y \in V(H^*)$ of vertices, if there is a path from x to y in H^*, then x appears before y in this ordering. In particular, ordering $\mathcal{O}$ must be consistent with the paths in $\tilde{\mathcal{Q}}$. Notice that we can use the drawing ψ of H in order to obtain a drawing of graph H^*, as follows. First, we delete from ψ all edges and vertices that do not participate in the paths in $\tilde{\mathcal{Q}}$. Next, every edge $e = (u, v)$ that remains in the current drawing, we create $\operatorname{cong}_{\tilde{\mathcal{Q}}}(e)$ copies of the edge e, each of which is drawn along the original drawing of e, with the copies being drawn close to each other. For each such edge e, we will view the different copies of e as different edges, and we think of each of these copies as belonging to a distinct path in $\tilde{\mathcal{Q}}$. We denote this new drawing of graph H^* by ψ^*. Note that ψ^* is a planar drawing.

For every vertex $v \in V(H^*)$, we let $\eta(v)$ be a small disc around the image of v in ψ^*. We denote by $\delta^+(v)$ the set of all edges that are leaving v in H^*, by $\delta^-(v)$ the set of all edges entering v, and by $\delta(v) = \delta^+(v) \cup \delta^-(v)$. For every edge $e \in \delta(v)$, we denote by $p_e(v)$ the unique point on the boundary of $\eta(v)$ that the image of e in ψ^* contains. We use the following simple observation.

Observation 1.9.2. *There is an efficient algorithm to compute, for every vertex $v \in V(H^*)$, a perfect matching $M(v) \subseteq \delta^-(v) \times \delta^+(v)$ between the edges of $\delta^-(v)$ and $\delta^+(v)$, and, for every pair $(e, e') \in M(v)$ of edges, a curve $\zeta(e, e')$ that is contained in $\eta(v)$ and connects $p_e(v)$ to $p_{e'}(v)$, such that all curves in $\left\{\zeta(e, e') \mid (e, e') \in M(v)\right\}$ are disjoint from each other.*

Proof. Observe first that the paths in $\tilde{\mathcal{Q}}$ define a perfect matching between edges of $\delta^-(v)$ and edges of $\delta^+(v)$, as each path $Q \in \tilde{\mathcal{Q}}$ that contains v must contain exactly one edge from $\delta^-(v)$ and exactly one edge from $\delta^+(v)$.

We maintain a disc η in the plane; originally, $\eta = \eta(v)$, and we gradually delete some areas of η, making it smaller. We also start with $M(v) = \emptyset$, and we perform $|\delta^-(v)|$ iterations. In every iteration, we select a pair $e \in \delta^-(v)$, $e' \in \delta^+(v)$ of edges that appear consecutively in the current drawing. We add (e, e') to $M(v)$, and we delete both of these edges from the current drawing. Additionally, we select a curve $\zeta(e, e')$, that is contained in the current disc η, connecting $p_e(v)$ to $p_{e'}(v)$, such that curve $\zeta(e, e')$ is disjoint from the images of all edges that remain in the drawing, and is close to the boundary of the disc η. Curve $\zeta(e, e')$ splits the disc η into two discs, η' and η'', where exactly one of the discs (we assume that it is η') contains the image of v, while the other disc is disjoint from the images of all edges that remain in the drawing. We set $\eta = \eta'$ and continue to the next iteration. It is immediate to see that, when the algorithm terminates, we obtain the desired matching $M(v)$, and a set $\left\{\zeta(e, e') \mid (e, e') \in M(v)\right\}$ of curves with the desired properties. $\square$

We now gradually transform the paths in $\tilde{\mathcal{Q}}$ in order to turn them into a set of non-interfering paths, as follows. We process all vertices in $V(H^*)$ one-by-one, according to the ordering $\mathcal{O}$. We now describe an iteration when vertex v is processed. Let $\mathcal{P}(v) \subseteq \tilde{\mathcal{Q}}$ be the set of paths containing the vertex v. For every path $\tilde{Q}_t \in \mathcal{P}(v)$, we delete the unique edge of $\delta^+(v)$ that lies on this path, thereby decomposing $\tilde{Q}_t$ into two sub-paths: path P_t^1 connecting t to v, and path P_t^2 connecting some vertex v' that is incident to an edge of $\delta^+(v)$ to u. Let $\mathcal{P}_1(v) = \left\{P_t^1 \mid \tilde{Q}_t \in \mathcal{P}(v)\right\}$, and let $\mathcal{P}_2(v) = \left\{P_t^2 \mid \tilde{Q}_t \in \mathcal{P}(v)\right\}$. We will now "glue" these paths together using the matching $M(v)$. Specifically, we construct a new set $\tilde{\mathcal{Q}}' = \left\{\tilde{Q}'_t \mid t \in \Gamma\right\}$ of paths as follows. Consider any vertex $t \in \Gamma$. If the original path $\tilde{Q}_t \in \tilde{\mathcal{Q}}$ does not lie in $\mathcal{P}(v)$, then we let $\tilde{Q}'_t = \tilde{Q}_t$. Otherwise, consider the unique path P_t^1 in $\mathcal{P}_1(v)$ that originates at t, and let e be the last edge on this path. Let e' be the unique edge of $\delta^+(v)$ that is matched to the edge e by the matching $M(v)$, and let $\tilde{Q}_{t'} \in \mathcal{P}(v)$ be the unique path that contained e'. We then let the new path $\tilde{Q}'_t$ be the concatenation of the path $P_1(t)$ and the path $P_2(t)$, thereby making e and e' consecutive on this path.

Once we process every vertex of H^*, we obtain the final collection $\mathcal{Q}'$ of paths, routing Γ to u. Clearly, for every edge $e \in E(H)$, $\mathrm{cong}_{\mathcal{Q}'}(e) \leq \mathrm{cong}_{\mathcal{Q}}(e)$. It is also easy to verify that the ordering $\mathcal{O}$ of the vertices of $V(H^*)$ is consistent with the paths in $\mathcal{Q}'$, since every path in $\mathcal{Q}'$ is a directed path in the directed acyclic graph H^*. We extend $\mathcal{O}$ to an ordering that includes all vertices of H arbitrarily. We next show that the paths in $\mathcal{Q}'$ are non-interfering, by providing a non-interfering representation of these paths. This representation exploits the drawing ψ^* of the graph H^*, and the curves $\left\{\zeta_{e,e'}(v)\right\}_{v\in V(H^*),(e,e')\in M(v)}$ that we defined. Consider any path $Q \in \tilde{\mathcal{Q}}'$, and let $Q = (v_1, v_2, \ldots, v_r = u)$. For all $1 \leq i < r$, we denote $e_i = (v_i, v_{i+1})$. The corresponding curve $\gamma(Q)$ is a concatenation of the following curves from the drawing ψ^*:

- The image of edge e_1, from the image of v_1, to the point $p_{e_1}(v_2)$ on the boundary of the disc $\eta(v_2)$;
- For all $1 \leq i < r$, curve $\zeta_{e_i,e_{i+1}}$, connecting the point $p_{e_i}(v_{i+1})$ to the point $p_{e_{i+1}}(v_{i+1})$ in $\eta(v_{i+1})$;
- For all $1 < i < r$, the image of the edge e_i, between points $p_{e_i}(v_i)$ and $p_{e_i}(v_{i+1})$; and
- Image of edge e_{r-1}, from $p_{e_{r-1}}(v_{r-1})$ to the image of v_r.

It is immediate to verify that we can draw all segments of $\gamma(Q)$, such that all resulting curves in $\left\{\gamma(Q')\right\}_{Q'\in\mathcal{Q}'}$ are disjoint from each other, and each of them is drawn in the thin strip S_Q around the image of Q' in ψ^*. From the definition of the drawing ψ^*, the resulting curves are a valid non-interfering representation of the paths in $\mathcal{Q}'$ with respect to the original drawing ψ of H.

1.9.6 Proof of Lemma 1.7.5

Throughout the section, we use R to denote the $(r \times r)$ grid, for some parameter r that is an integral power of 2, and we use I to denote the set of vertices in its last row.

We use the following lemma.

Lemma 1.9.3. *There is an efficient algorithm, that, given an n-vertex planar graph H and a subset S of $|S| = r$ vertices of $V(H)$ that are α'-well-linked in H, for some $0 < \alpha' < 1$, computes a distribution $\mathcal{D}$ over pairs $(u^*, \mathcal{Q})$, where u^* is a vertex of H, and $\mathcal{Q}$ is a collection of paths in H routing vertices of S to u^*, such that the distribution $\mathcal{D}$ has support size at most $O(r^2)$, and for each edge $e \in E(H)$,*

$$\mathbb{E}_{(u^*,\mathcal{Q})\in\mathcal{D}}[(\mathrm{cong}_{\mathcal{Q}}(e))^2] = O\left(\frac{\log r}{(\alpha')^4}\right).$$

We provide the proof of Lemma 1.9.3 later, after we complete the proof of Lemma 1.7.5 using it. Let $\mathcal{D}$ be the distribution we get from the algorithm in Lemma 1.9.3 applied to graph H, set S and parameter α'. From linearity of expectation, $\mathbb{E}_{(u^*,\mathcal{Q})\in\mathcal{D}}[\sum_{e\in E(H)} w(e) \cdot (\mathrm{cong}_{\mathcal{Q}}(e))^2] = O\left(\frac{\log r}{(\alpha')^4} \cdot \sum_{e\in E(H)} w(e)\right)$. Clearly, there exists a pair $(\hat{u}^*, \hat{\mathcal{Q}})$ with non-zero probability in $\mathcal{D}$, such that $\sum_{e\in E(H)} w(e) \cdot (\mathrm{cong}_{\hat{\mathcal{Q}}}(e))^2 = O\left(\frac{\log r}{(\alpha')^4} \cdot \sum_{e\in E(H)} w(e)\right)$. Since the distribution $\mathcal{D}$ has support size $O(r^2)$, such a pair can be found by checking all pairs $(u^*, \mathcal{Q})$ with non-zero probability in $\mathcal{D}$.

The remainder of this section is dedicated to the proof of Lemma 1.9.3.

We use the following claim from [9] and its corollary. We note that the claim appearing in [9] is somewhat weaker, but their proof immediately implies the stronger result that we state below.

Claim 1.9.4 (Claim D.11 from [9]). *There is a distribution $\mathcal{D}$ over pairs $(u^*, \mathcal{Q})$, where u^* is a vertex of R, and $\mathcal{Q}$ is a collection of paths in R connecting every vertex of I to u^*, such that, for each edge $e \in E(R)$, $\mathbb{E}_{(u^*,\mathcal{Q})\in\mathcal{D}}[(\mathrm{cong}_{\mathcal{Q}}(e))^2] = O(\log r)$. Moreover, such a distribution with support size at most $O(r^2)$ can be computed efficiently.*

We say that a graph J is a *minor* of a graph G, iff there is a function h, mapping each vertex $v \in V(J)$ to a connected subgraph $h(v) \subseteq G$, and each edge $e = (u, v) \in E(J)$ to a path $h(e)$ in G connecting a vertex of $h(u)$ to a vertex of $h(v)$, such that: (i) for all

$u, v \in V(J)$, if $u \neq v$, then $h(u) \cap h(v) = \emptyset$; and (ii) the paths in the set $\{h(e) \mid e \in E(J)\}$ are mutually internally disjoint, and they are internally disjoint from $\bigcup_{v \in V(J)} h(v)$. A function h satisfying these conditions is called a *model* of J in G. We use the following corollary of Claim 1.9.5.

Corollary 1.9.5. *There is an efficient deterministic algorithm, that, given a graph G that contains R as a minor, together with the model h of R in G, and for each vertex $x \in V(R)$, a vertex $v_x \in h(x)$, computes a distribution $\tilde{\mathcal{D}}$ on pairs $(\tilde{u}^*, \tilde{\mathcal{Q}})$, where $\tilde{u}^*$ is a vertex in G, and $\tilde{\mathcal{Q}}$ is a collection of paths in G connecting every vertex of $\{v_x \mid x \in I\}$ to $\tilde{u}^*$, such that the distribution has support size at most $O(r^2)$, and for each edge $e \in E(G)$, $\mathbb{E}_{(\tilde{u}^*, \tilde{\mathcal{Q}}) \in \tilde{\mathcal{D}}}[(\mathrm{cong}_{\tilde{\mathcal{Q}}}(e))^2] = O(\log r)$.*

Proof. For each vertex $x \in V(R)$, we let $\delta(x)$ be the set of edges incident to x in R, so $|\delta(x)| \leq 4$. For each edge $e \in \delta(x)$, we denote by $b_x(e)$ the vertex in $h(x)$ that serves as the endpoint of the path $h(e)$. We denote $B(x) = \{b_x(e) \mid e \in \delta(x)\}$. We now select: (i) for each pair $b_x(e), b_x(e')$ of distinct vertices of $B(x)$, a path $P^x_{e,e'}$ in $h(x)$ that connects $b_x(e)$ to $b_x(e')$; and (ii) for each vertex $b_x(e) \in B(x)$, a path W^x_e connecting v_x to $b_x(e)$. We call these paths *auxiliary paths in $h(x)$*.

We now apply Claim 1.9.4 to R. Let $\mathcal{D}$ be the distribution over pairs $(u^*, \mathcal{Q})$ that we get, where u^* is a vertex of R, and $\mathcal{Q}$ is a collection of paths in R connecting every vertex of I to u^*. We now use the model h of R in G, and the auxiliary paths to transform the distribution $\mathcal{D}$ into another distribution $\tilde{\mathcal{D}}$ over pairs $(\tilde{u}^*, \tilde{\mathcal{Q}})$, where $\tilde{u}^*$ is a vertex in $\{h(x) \mid x \in V(R)\}$, and $\tilde{\mathcal{Q}}$ is a collection of paths in G connecting every vertex of $\{v_x \mid x \in I\}$ to $\tilde{u}^*$, as follows. For each pair $(u^*, \mathcal{Q})$ with non-zero probability in $\mathcal{D}$, we define a corresponding pair $(\tilde{u}^*, \tilde{\mathcal{Q}})$ as follows. We set $\tilde{u}^* = v_{u^*}$. Let $Q = (x_1, \ldots, x_{r-1}, x_r = u^*)$ be a path in $\mathcal{Q}$, where we denote $e_i = (x_i, x_{i+1})$ for each $1 \leq i \leq r-1$. We let $\tilde{Q}$ be the path obtained by concatenating the paths $W^{x_1}_{e_1}, h(e_1), P^{x_2}_{e_1,e_2}, h(e_2), P^{x_3}_{e_2,e_3}, \ldots, h(e_{r-1}), W^{x_r}_{e_{r-1}}$. It is easy to verify that the path $\tilde{Q}$ is a path in G that connects v_{x_1} to v_{u^*}. We then let $\tilde{\mathcal{Q}} = \left\{\tilde{Q} \mid Q \in \mathcal{Q}\right\}$. Therefore,

$\tilde{\mathcal{Q}}$ is a collection of paths in G connecting every vertex of $\{v_x \mid x \in I\}$ to $\tilde{u}^*$. To define $\tilde{\mathcal{D}}$, we simply assign, for every pair $(u^*, \mathcal{Q})$ with non-zero probability in $\mathcal{D}$, the same probability to the pair $(\tilde{u}^*, \tilde{Q})$.

It remains to show that, for each edge $\tilde{e} \in E(G)$, $\mathbb{E}_{(\tilde{u}^*, \tilde{\mathcal{Q}}) \in \tilde{\mathcal{D}}}[(\mathrm{cong}_{\tilde{\mathcal{Q}}}(\tilde{e}))^2] = O(\log r)$. We first consider an edge $\tilde{e}$ that does not belong to any subgraph of $\{h(x) \mid x \in V(R)\}$. Clearly either $\tilde{e}$ belongs to a unique path $h(e)$ in $\{h(e) \mid e \in R\}$, or it does not belong to any path in $\{h(e) \mid e \in R\}$. If the latter case happens, then $\mathbb{E}_{(\tilde{u}^*, \tilde{\mathcal{Q}}) \in \tilde{\mathcal{D}}}[(\mathrm{cong}_{\tilde{\mathcal{Q}}}(\tilde{e}))^2] = 0$. If the former case happens, then $\mathbb{E}_{(\tilde{u}^*, \tilde{\mathcal{Q}}) \in \tilde{\mathcal{D}}}[(\mathrm{cong}_{\tilde{\mathcal{Q}}}(\tilde{e}))^2] \leq \mathbb{E}_{(u^*, \mathcal{Q}) \in \mathcal{D}}[(\mathrm{cong}_{\mathcal{Q}}(e))^2] = O(\log r)$. Consider now an edge $\tilde{e}$ in $h(x)$ for some vertex $x \in V(R)$. Note that, from the construction of $\tilde{\mathcal{D}}$, whenever the edge $\tilde{e}$ is contained in some path $\tilde{Q}$, the corresponding path Q in R has to contain at least one edge of $\delta(x)$. Therefore, for each pair $(u^*, \mathcal{Q})$ with non-zero probability in $\mathcal{D}$, $\mathrm{cong}_{\tilde{\mathcal{Q}}}(\tilde{e}) \leq \sum_{e \in \delta(x)} \mathrm{cong}_{\mathcal{Q}}(e)$. As a result, $\mathbb{E}_{(\tilde{u}^*, \tilde{\mathcal{Q}}) \in \tilde{\mathcal{D}}}[(\mathrm{cong}_{\tilde{\mathcal{Q}}}(\tilde{e}))^2] \leq \mathbb{E}_{(u^*, \mathcal{Q}) \in \mathcal{D}}[(4 \cdot \max_{e \in \delta(x)} \{\mathrm{cong}_{\mathcal{Q}}(e)\})^2] = O(\log r)$. $\square$

We use the following lemma, whose proof is deferred to Section 1.9.6.

Lemma 1.9.6. *There exists an efficient algorithm that, given a planar graph G with maximum vertex degree Δ and a set S of r vertices that is α'-well-linked in G for some $0 < \alpha' < 1$, computes (i) an R-minor in G, where R is the $(k \times k)$ grid and $k = \Omega(\alpha' r / \mathrm{poly}(\Delta))$ and k is an integral power of 2, together with a model h of R in G; (ii) for each vertex $x \in V(R)$, a vertex $v_x \in h(x)$ in G; and (iii) k edge-disjoint paths in G, each connecting a distinct vertex of S to a distinct vertex of $\{v_x \mid x \in I\}$.*

We now prove Lemma 1.9.3 using Corollary 1.9.5 and Lemma 1.9.6. Recall that we are given a planar graph H and a set $S \subseteq V(H)$ of vertices that are α'-well-linked for some $0 < \alpha' < 1$. Let φ be a planar drawing of H. Since the maximum vertex-degree in H could be as large as n, the size of the grid minor obtained by directly applying Lemma 1.9.6 to H may be too small for us. We therefore construct a graph H' from H, that has constant maximum vertex-degree. We start from H, and process every vertex of $V(H)$ as follows. Let v be a vertex of $V(H)$, let $d = \deg_H(v)$ and let $e_1, \ldots, e_d$ be the edges incident to v in H, indexed according to the circular ordering in which they enter the image of v in the drawing φ. We let R_v be the $(d \times d)$-grid, and we denote the vertices of its first row by $x_1(v), \ldots, x_d(v)$. We then replace the vertex v by the graph R_v, and let, for each $1 \le i \le d$, the edge e_i be now incident to vertex $x_i(v)$. Let H' be the graph obtained after all vertices in $V(H)$ are processed. It is easy to see that the max vertex-degree of H' is 4, and H can be simply obtained from H' by contracting each cluster in $\{R_v \mid v \in V(H)\}$ back into the vertex v, so each edge of H is also an edge of H'. We use the following simple observations whose proofs are straightforward and are omitted here.

Observation 1.9.7. *Let $\mathcal{Q}'$ be a set of paths in H'. For each path $Q' \in \mathcal{Q}'$, let Q be the path obtained from Q' by contracting, for each vertex $v \in V(H)$, every edge of R_v that lies on path Q'. Define $\mathcal{Q} = \left\{Q \mid Q' \in \mathcal{Q}'\right\}$. Then for each edge $e \in E(H)$, $\mathrm{cong}_{\mathcal{Q}}(e) \le \mathrm{cong}_{\mathcal{Q}'}(e)$.*

Observation 1.9.8. *The set $S' = \{x_1(v) \mid v \in S\}$ of vertices is α'-well-linked in H'.*

For a set $\mathcal{Q}'$ of paths in H, we denote $\mathrm{cong}(\mathcal{Q}') = \max_{e \in E(H)} \left\{\mathrm{cong}_{\mathcal{Q}'}(e)\right\}$.

First, we apply the algorithm in Lemma 1.9.6 to graph H' and the input vertex set S, to compute a model h of an R-minor in H', where R is the $(k \times k)$-grid with $k = \Omega(\alpha' r)$. We also obtain, for each vertex $x \in V(R)$, a vertex $v_x \in h(x)$ in H'; and a set of k edge-disjoint paths in H', each connecting a distinct vertex of S' to a distinct vertex of $\{x_1(z) \mid z \in I\}$. We denote this set of paths by $\mathcal{Q}_1$, and for each vertex $v \in S$, we denote the path in $\mathcal{Q}_1$ that contains v as one of its endpoints by Q^1_v if such a path exists.

Let $S'_1 \subseteq S'$ be the set of endpoints of paths in $\mathcal{Q}_1$ that lie in S, so $|S_1| = k = \Omega(\alpha' r)$. We arbitrarily partition the set $S' \setminus S'_1$ of vertices into groups $S'_2, \ldots, S'_t$, where each group of $S'_2, \ldots, S'_{t-1}$ contains exactly $|S'_1|$ vertices, and the last group S'_t contains at most $|S'_1|$ vertices, so $t = O(1/\alpha')$. Since the set S' of vertices is α'-well-linked in H', for each $2 \le i \le t$, there exists a set $\mathcal{P}_i$ of paths in H', each connecting a distinct vertex of S'_i to a distinct vertex of S'_1, such that $\mathrm{cong}(\mathcal{P}_i) = O(1/\alpha')$. Additionally, let the set $\mathcal{P}_1$ contain, for each vertex

$v \in S'_1$, a path that only contains the single vertex v. Denote $\mathcal{Q}_2 = \bigcup_{1 \le i \le t} \mathcal{P}_i$. Then set $\mathcal{Q}_2$ contains, for each vertex $v \in S'$, a path connecting v to a vertex in S'_1, such that each vertex $u \in S'_1$ serves as the endpoint of at most $O(1/\alpha')$ paths in $\mathcal{Q}_2$, and $\mathrm{cong}(\mathcal{Q}_2) \le O(1/(\alpha')^2)$. For each path $Q \in \mathcal{Q}_2$, let $s(Q) \in S'_1$ be the endpoint of Q in S'_1.

Next, we use the algorithm from Corollary 1.9.5 to compute a distribution $\tilde{\mathcal{D}}$ on pairs $(\tilde{u}^*, \tilde{\mathcal{Q}})$, where $\tilde{u}^*$ is a vertex in H', and $\tilde{\mathcal{Q}}$ is a collection of paths in H' routing vertices of $\{x_1(z) \mid z \in I\}$ to $\tilde{u}^*$, such that distribution with support size at most $O(r^2)$, for each edge $e \in E(H')$, $\mathbb{E}_{(\tilde{u}^*, \tilde{\mathcal{Q}}) \in \tilde{\mathcal{D}}}[(\mathrm{cong}_{\tilde{\mathcal{Q}}}(e))^2] = O(\log k) = O(\log r)$.

We now construct a distribution $\hat{\mathcal{D}}$ on pairs $(\hat{u}^*, \hat{\mathcal{Q}})$, where $\hat{u}^*$ is a vertex in H' and $\hat{\mathcal{Q}}$ is a collection of paths in H' routing S' to $\hat{u}^*$, as follows. Consider a pair $(\tilde{u}^*, \tilde{\mathcal{Q}})$ in distribution $\tilde{\mathcal{D}}$ with non-zero probability. We let the set $\hat{\mathcal{Q}}$ contain, for each path $Q \in \mathcal{Q}_2$, a path formed by the concatenation of (i) the path $Q \in \mathcal{Q}_2$; (ii) the path $Q^1_{s(Q)} \in \mathcal{Q}_1$ (the path in $\mathcal{Q}_1$ whose endpoint in S is $s(Q)$); and (iii) the path in $\tilde{\mathcal{Q}}$ that connects $s(Q)$ to $\tilde{u}^*$. It is clear that the set $\hat{\mathcal{Q}}$ contains, for each $v \in S'$, a path that connects v to $\tilde{u}^*$. We add the pair $(\tilde{u}^*, \hat{\mathcal{Q}})$ to distribution $\hat{\mathcal{D}}$ with the same probability as the pair $(\tilde{u}^*, \tilde{\mathcal{Q}})$ in distribution $\tilde{\mathcal{D}}$.

From the definition of the set $\hat{\mathcal{Q}}$ of paths, and the property that each vertex in S'_1 serves as the endpoint of at most $O(1/\alpha')$ paths in $\mathcal{Q}_2$, we get that each path in $\mathcal{Q}_1$ serves as a subpath of at most $O(1/\alpha')$ paths in $\hat{\mathcal{Q}}$, and the same holds for $\tilde{\mathcal{Q}}$. Therefore, for each edge $e \in H'$,

$$\begin{aligned}\mathrm{cong}_{\hat{\mathcal{Q}}}(e) &= O(1/\alpha') \cdot \mathrm{cong}_{\mathcal{Q}_1}(e) + \mathrm{cong}_{\mathcal{Q}_2}(e) + O(1/\alpha') \cdot \mathrm{cong}_{\tilde{\mathcal{Q}}}(e)\\ &= O(1/(\alpha')^2) + O(1/\alpha') \cdot \mathrm{cong}_{\tilde{\mathcal{Q}}}(e).\end{aligned}$$

Therefore, for each edge $e \in H'$,

$$\begin{aligned}\mathbb{E}_{(\hat{u}^*, \hat{\mathcal{Q}}) \in \hat{\mathcal{D}}}[(\mathrm{cong}_{\hat{\mathcal{Q}}}(e))^2] = O(1/(\alpha')^4) + O(1/(\alpha')^3) \cdot \mathbb{E}_{(\tilde{u}^*, \tilde{\mathcal{Q}}) \in \tilde{\mathcal{D}}}[\mathrm{cong}_{\tilde{\mathcal{Q}}}(e)]\\ + O(1/(\alpha')^2) \cdot \mathbb{E}_{(\tilde{u}^*, \tilde{\mathcal{Q}}) \in \tilde{\mathcal{D}}}[(\mathrm{cong}_{\tilde{\mathcal{Q}}}(e))^2] = O(\log r/(\alpha')^4).\end{aligned}$$

Finally, we define the distribution $\mathcal{D}$ on pairs $(u^*, \mathcal{Q})$ where u^* is a vertex in H and $\mathcal{Q}$ is a collection of paths in H routing S to u^*, as follows. Consider a pair $(\hat{u}^*, \hat{\mathcal{Q}})$ in $\hat{\mathcal{D}}$ with non-zero probability. We let $\mathcal{Q}$ contain, for every path $\hat{Q} \in \hat{\mathcal{Q}}$ connecting a vertex of $x_1(v_1)$ to a vertex of $x_1(v_2)$ for a pair v_1, v_2 of vertices of S, the corresponding path in H connecting vertex v_1 to vertex v_2 (obtained from Q' by contracting each cluster in $\{R_v \mid v \in V(H)\}$ back into the vertex v). From Lemma 1.9.7, for each edge in $E(H)$, $\mathrm{cong}_{\mathcal{Q}}(e) \leq \mathrm{cong}_{\hat{\mathcal{Q}}}(e)$. Therefore, it follows immediately that $\mathbb{E}_{(u^*,\mathcal{Q})\in\mathcal{D}}[(\mathrm{cong}_{\mathcal{Q}}(e))^2] = O(\log r/(\alpha')^4)$. In order to complete the proof of Lemma 1.9.3, it now remains to prove Lemma 1.9.6, which we do next.

Proof of Lemma 1.9.6

We now provide the proof of Lemma 1.9.6. Our proof uses techniques similar to those used in the proof of Theorem 3.1 of [6].

We assume that we are given some fixed drawing of G on the sphere. We fix a point ν on the sphere that does not belong to the image of G. A *contour* λ with respect to this drawing is a simple closed curve that does not contain point ν, and only intersects the drawing at the vertices of G. We denote by V_λ the set of vertices of G whose image lies on λ, and we refer to $|V_\lambda|$ as the *length* of λ. We say that a subset $A \subseteq V_\lambda$ of vertices is *contiguous* iff A contains all vertices of G that lie on a contiguous subcurve of λ. Clearly, a contour λ separates the sphere into two open regions. We define the interior $\mathrm{ins}(\lambda)$ of λ to be the region not containing the point ν, and define the graph G_λ to be the subgraph of G consisting of all edges and vertices whose image lies in $\lambda \cup \mathrm{ins}(\lambda)$.

The proof consists of two steps. Throughout the proof, we set $\beta = \lceil \alpha' r/(100\Delta) \rceil$. In the first step, we will construct a contour λ such that at least half of vertices of S lie in $\mathrm{ins}(\lambda)$, and the following additional properties hold:

P1. $|V_\lambda| = \beta$;

P2. for each pair $A, B \subseteq V_\lambda$ of disjoint equal-cardinality contiguous subsets of vertices, there exists $|A|$ node-disjoint paths in G_λ connecting vertices of A to vertices of B; and

P3. there exist a set of $\lfloor \beta/2 \rfloor$ edge-disjoint paths in G, each connecting a distinct vertex of S that lies inside the interior of λ to a distinct vertex of V_λ.

In the second step, we will use the contour constructed in the first step in order to compute a grid minor and the edge-disjoint paths connecting vertices of S to it. Before we describe each step in details, we state and prove the following observation.

Observation 1.9.9. *If λ is a contour such that $|V_\lambda| \leq \beta$, then the number of vertices of S that lie in the interior of λ is either at most $r/10$ or at least $9r/10$.*

Proof. Let r' be the number of vertices of S that lie in the interior of the contour λ, and let r'' be the number of vertices of S that lie in the exterior of the contour λ. Assume first that $r' \leq r''$. Since the vertices of S are α'-well-linked in G, $|\operatorname{out}_G(V(G_\lambda) \setminus V_\lambda)| \geq \alpha' r'$. Note that every edge in $\operatorname{out}_G(V(G_\lambda) \setminus V_\lambda)$ must be incident to a vertex of V_λ, so $\alpha' r' \leq \Delta |V_\lambda| \leq \alpha' r/25$. Therefore, $r' \leq r/25$. Assume now that $r' > r''$. It is easy to see that we can derive that $r'' \leq r/25$ similarly. Therefore, $r' \geq r - r'' - |V_\lambda| \geq 9r/10$. □

We say that a contour λ is *short* iff $|V_\lambda| \leq \beta$, and we say that a contour λ is *fat* iff the number of vertices of S that lie in $\operatorname{ins}(\lambda)$ is at least $9r/10$. We use the following claim that appears in the (first paragraph of) proof of Theorem 3.1 of [6].

Claim 1.9.10. *There is an efficient algorithm that, given a short and fat contour λ of length less than β, computes another short and fat contour λ' of length exactly β, such that $G_{\lambda'} \subsetneq G_\lambda$.*

Step 1. Computing a Contour

We now describe the algorithm for the first step. The algorithm maintains a contour $\hat{\lambda}$, that is initialized to be the small circle around the point ν that does not intersect any vertices of G. The algorithm will iteratively update $\hat{\lambda}$, and will continue to be executed as long as not all properties P1, P2, P3 are satisfied by $\hat{\lambda}$. Note that each of these properties can be checked efficiently. Clearly, $\hat{\lambda}$ is short and fat initially. We will ensure that this is true for all curves $\hat{\lambda}$ that are considered over the course of the algorithm. Moreover, as we will see in the description, graph G_{λ} will become smaller after each iteration. Therefore, the algorithm will eventually terminate and output a desired contour. We now describe an iteration. We distinguish between the following three cases.

Case 1. Property P1 is not satisfied. In this case we simply apply the algorithm in Claim 1.9.10 to $\hat{\lambda}$, update $\hat{\lambda}$ to be the contour $\hat{\lambda}'$ that we obtain, and then continue to the next iteration. From Claim 1.9.10 and Observation 1.9.9, the new contour is short, fat, and satisfies that $G_{\hat{\lambda}'} \subsetneq G_{\hat{\lambda}}$.

Case 2. Property P2 is not satisfied. In this case we let A, B be a pair of disjoint contiguous subsets of V_{λ} such that $|A| = |B|$, and there does not exist a set of $|A|$ node-disjoint paths connecting vertices of A to vertices of B. From among all such pairs of subsets, we choose one where $|A|$ is minimized. We use the following claim, which is an immediate corollary of Theorem 3.6 in [34].

Claim 1.9.11. *There is a simple non-closed curve J, such that:*

- *J is entirely contained in $\hat{\lambda} \cup ins(\hat{\lambda})$;*
- *J only intersects the drawing of $G_{\hat{\lambda}}$ at its vertices, and only intersects with $\hat{\lambda}$ at the endpoints a, b of J (and a, b are not vertices of G);*

- *if we denote by U_1 and U_2 the two subcurves of $\hat{\lambda}$ connecting a and b, then either all vertices of A lie on U_1 and all vertices of B lie on U_2, or all vertices of A lie on U_2 and all vertices of B lie on U_1; and*

- *the number of vertices lying on J is at most $|A| - 1$.*

Moreover, such a curve J can be found efficiently.

We compute such a curve J, and assume without loss of generality that vertices of A lie on U_1 and vertices of B lie on U_2. We define the new contour λ_1 to be the concatenation of U_1 and J, and the new contour λ_2 to be the concatenation of U_2 and J. Clearly, at least one of λ_1 and λ_2 contains at least $r/3$ vertices of S in its interior. Assume without loss of generality that it is λ_1. Then $|V_{\lambda_1}| \leq |V_{\hat{\lambda}}| - |B| + |J| \leq \beta - 1$. From Observation 1.9.9, the interior of λ_1 contains at least $9r/10$ vertices of S. We update $\hat{\lambda}$ to be λ_1 and continue to the next iteration. From the above discussion, λ_1 is a short and fat contour, and satisfies that $G_{\lambda_1} \subsetneq G_{\hat{\lambda}}$.

Case 3. Property P3 is not satisfied. Let $\tilde{G}_{\hat{\lambda}}$ be a graph that is obtained from $G_{\hat{\lambda}}$ by adding to it (i) two new vertices s, t; (ii) for each vertex v of S that lies in the interior of $\hat{\lambda}$, an edge (s, v); and (iii) for each vertex v' in $V_{\hat{\lambda}}$, an edge (t, v'). We assign capacity 1 to each edge of $\tilde{G}_{\hat{\lambda}}$, and then compute the minimum cut separating s from t in $\tilde{G}_{\hat{\lambda}}$. Since the property P3 is not satisfied, the minimun cut has value at most $\lfloor \beta/2 \rfloor - 1$. Denote by E' the set of edges in the cut. Let $E'_1 \subseteq E'$ contain all edges in E' that are incident to s, so $|E'_1| \leq \lfloor \beta/2 \rfloor$. Let $S' \subseteq S$ be the set of vertices in S that lie in the interior of $\hat{\lambda}$, and are not an endpoint of an edge in E'_1, so $|S'| \geq 4r/5$. Let $E'_2 = E' \setminus E'_1$, so $|E'_2| \leq \lfloor \beta/2 \rfloor$. Note that, in the dual graph of $G_{\hat{\lambda}}$ with respect to its drawing, the edges corresponding to the edges of E'_2 form a set of cycles that separate the faces corresponding to vertices of S' from the faces corresponding to vertices of $V_{\hat{\lambda}}$, and these cycles naturally form a set of disjoint closed curves in the drawing of G, such that each vertex of S' lies in the interior of one of the curves, and all vertices of $V_{\hat{\lambda}}$ lie in the exterior of each of these curves. It is not hard to see that, each of these closed curves can be further transformed into a contour, by shifting every intersection between the curve and an edge of G to an endpoint of the edge, such that (i) the resulting contour contains the same set of vertices and edges in its interior as the closed curve; and (ii) the length of the contour is at most the number of intersections between the curve and the drawing of $G_{\hat{\lambda}}$. Let $\lambda_1, \ldots, \lambda_l$ be the contours that we obtain. From the above discussion, the total length of $\lambda_1, \ldots, \lambda_l$ is at most $\lfloor \beta/2 \rfloor$. Using reasoning similar to that in the proof of Observation 1.9.9, one of these contours contains at least $9r/10$ vertices of S' (otherwise, the removal of all vertices lying on the countours separates the graph into connected components, each of which contains fewer than $9r/10$ vertices of S', contradicting the well-linkedness of S). Assume without loss of generality that it is λ_1. We then update $\hat{\lambda}$ to λ_1 and continue to the next iteration. Clearly, λ_1 is short and fat, and satisfies that $G_{\lambda_1} \subsetneq G_{\hat{\lambda}}$.

Step 2. Constructing the Grid Minor

Let λ be the contour that is obtained from the first step. We denote $V(\lambda) = \{v_1, \ldots, v_\beta\}$, where the vertices in $V(\lambda)$ are indexed in the clockwise order of their appearance on λ. Denote $\gamma = \lfloor \beta/4 \rfloor$. We partition the vertices on λ into 4 consecutive subsets of cardinality γ each: for each $0 \leq i \leq 3$, $B_i = \{v_j \mid i\gamma + 1 \leq h \leq (i+1)\gamma\}$. From property P2, we can find a set $\mathcal{P}_0$ of γ node-disjoint paths connecting vertices of B_0 to vertices of B_2, and another set $\mathcal{P}_1$ of γ node-disjoint paths connecting vertices of B_1 to vertices of B_3. We now compute a grid minor in G_λ from the sets $\mathcal{P}_0$ and $\mathcal{P}_1$ of paths.

Let H be the graph consisting of all paths in $\mathcal{P}_0$ and $\mathcal{P}_1$. We first iteratively modify H as follows. If there is an edge e in H, such that in the graph $H \setminus \{e\}$, there is a set of γ node-disjoint paths connecting vertices of B_0 to vertices of B_2, and another set of γ node-disjoint paths connecting vertices of B_1 to vertices of B_3, then we delete e from H and continue to the next iteration. We call such an edge e an *irrelevant* edge. We iteratively remove irrelevant edges from H in this way until we are not able to do so. Let $\hat{H}$ be the remaining graph, so $\hat{H}$ does not contain any irrelevant edge. Let $\hat{\mathcal{P}}_0$ be a set of γ node-disjoint paths connecting vertices of B_0 to vertices of B_2 in $\hat{H}$, and let $\hat{\mathcal{P}}_1$ be a set of γ node-disjoint paths connecting vertices of B_1 to vertices of B_3 in $\hat{H}$.

We claim that, for each path $P \in \hat{\mathcal{P}}_0$ and each path $P' \in \hat{\mathcal{P}}_1$, $P \cap P'$ is a path. Note that this implies that combining the sets $\hat{\mathcal{P}}_0$ and $\hat{\mathcal{P}}_1$ of paths yields a $(\gamma \times \gamma)$-grid minor in G_λ. We now prove the claim. We call the paths in $\hat{\mathcal{P}}_0$ *vertical paths* and view them as being directed from vertices of B_0 to vertices of B_2. We call the paths in $\hat{\mathcal{P}}_1$ *horizontal paths*, and view them as being directed from vertices of B_1 to vertices of B_3. Denote $\hat{\mathcal{P}}_0 = \left\{P_1, P_2, \ldots, P_\gamma\right\}$, where for each $1 \leq i \leq \gamma$, the endpoint in B_0 of path P_i is v_i. Denote $\hat{\mathcal{P}}_1 = \left\{P'_1, P'_2, \ldots, P'_\gamma\right\}$, where for each $1 \leq j \leq \gamma$, the endpoint in B_1 of path P'_j is $v_{\gamma+j}$. Note that the planar drawing of G induces a planar drawing of $\hat{H}$. Since all vertices of B_0, B_1, B_2, B_3 lie on the contour γ, the image of each path $P_i \in \hat{\mathcal{P}}_0$ separates the interior of λ into two regions, that we call the *left region of* P_i and *right region of* P_i, respectively. In particular, the left region of P_i contains the image of $P_1, \ldots, P_{i-1}$, and right region of P_i contains the image of $P_{i+1}, \ldots, P_\gamma$. We define the left and right regions for each path $P'_j \in \hat{\mathcal{P}}_1$ similarly.

Assume the claim is false, and assume without loss of generality that some vertical path P visits some horizontal path P' more than once. Therefore, either there is a subpath of P whose image lies in the right region of P' and does not contain any vertex of P' as its inner vertex, that we call a *bump*, or there is a subpath of P whose image lies in the left region of P' and does not contain any vertex of P' as its inner vertex, that we call a *pit*. We show that neither bumps nor pits may exist, thus completing the proof of the claim. We now show that bumps do not exist. The arguments for pits are symmetric. Assume for contradiction that there is a bump. Consider a bump that is created by a path $P \in \hat{\mathcal{P}}_0$ and a path $P' \in \hat{\mathcal{P}}_1$, and let u, w be two vertices shared by P and P'. We say that a bump is *aligned* iff u appears before w on both paths P, P', or w appears before u on both paths P, P'. Since we can reverse the direction of the paths in $\hat{\mathcal{P}}_0, \hat{\mathcal{P}}_1$, we can assume without loss of generality that there exists an aligned bump.

We now take the aligned bump that, among all pairs $P_i \in \hat{\mathcal{P}}_0, P'_j \in \hat{\mathcal{P}}_1$ of paths that form an aligned bump that minimizes j, minimizes i, namely

$$j = \min \left\{ j' \mid \exists i', \text{ s.t. } P_{i'}, P'_{j'} \text{ form an aligned bump} \right\}, \text{ and}$$

$$i = \min \left\{ i \mid P_i, P'_j \text{ form an aligned bump} \right\}.$$

Let u, w be the vertices shared by P_i and P'_j, with u appearing before w on both P_i and P'_j. Let Q be the subpath of P_i between u and w, and let Q' be the subpath of P'_j between u and w. We now distinguish between the following two cases, depending on whether or not Q' contains a vertex of some other path $P_{i'} \in \hat{\mathcal{P}}_0$ with $i' \neq i$ as an inner vertex. We first assume that Q' does not contain such a vertex, then we claim that the first edge e of Q is irrelevant. To see this, observe first that no path of $\hat{\mathcal{P}}_1$ may contain e, since otherwise the paths in $\hat{\mathcal{P}}_1$ are not node-disjoint. Therefore, if we modify the path P_i in $\hat{\mathcal{P}}_0$ by replacing the segment Q by Q', then we obtain a new pair $\hat{\mathcal{P}}'_0, \hat{\mathcal{P}}_1$ of sets of node-disjoint paths in $\hat{H} \setminus e$, where $\hat{\mathcal{P}}'_0$ routes B_0 to B_2 and $\hat{\mathcal{P}}_1$ routes B_1 to B_3. This contradicts the fact that $\hat{H}$ contains no irrelevant edges. We now consider the case where Q' does contain a vertex u' from a path $P_{i'} \in \hat{\mathcal{P}}_0$ with $i' \neq i$. From the definition of an aligned bump, u' lies in the left region of P_i, and therefore $i' < i$. Since we view the path $P_{i'}$ as being directed from a vertex of B_0 to a vertex of B_2, it is easy to see that the subpath of $P_{i'}$ between u' and its endpoint in B_2 must contain another vertex of P'_j. Let w' be the first such vertex on the subpath of $P_{i'}$ between u' and its endpoint in B_2, and we denote by Q'' the subpath of $P_{i'}$ between u' and w'. We claim that Q'' does not contain any vertex of another path $P'_{j'}$ with $j' \neq j$. To see this, observe first that the image of Q'' lies in the left region of P'_j, since otherwise the pair $P_{i'}, P'_j$ of paths also creates an aligned bump, contradicting to the choice of i. Observe next that Q'' cannot contain any vertex of another path $P'_{j'}$ with $j' < j$, since otherwise the pair

$P_{i'}, P'_{j'}$ of paths also creates an aligned bump, contradicting to the choice of j. Therefore, Q'' does not contain any vertex of another path $P'_{j'}$ with $j' \neq j$. We now show that the edge of P'_j going out of u' (that we denote by e') is irrelevant. First, since $u' \in P_{i'} \cap P'_j$, no other path of $\hat{\mathcal{P}}_0 \cup \hat{\mathcal{P}}_1$ may contain u', and therefore no other path of $\hat{\mathcal{P}}_0 \cup \hat{\mathcal{P}}_1$ may contain the edge e'. We can then modify the path P'_j in $\hat{\mathcal{P}}_2$ by replacing its segment between u' and w' by Q'. Clearly, we obtain a new pair $\hat{\mathcal{P}}_0, \hat{\mathcal{P}}'_1$ of sets of node-disjoint paths in $\hat{H} \setminus e$, where $\hat{\mathcal{P}}_0$ routes B_0 to B_2 and $\hat{\mathcal{P}}'_1$ routes B_1 to B_3. This contradicts the fact that $\hat{H}$ contains no irrelevant edges. Therefore, no bumps may exists.

Let h' be the model that embeds the $(\gamma \times \gamma)$-grid into G_λ.

Since the property P3 is satisfied, we can efficiently find a set $\hat{\mathcal{P}}$ of at least $\lfloor \beta/2 \rfloor$ edge-disjoint paths, each connecting a distinct vertex of S to a distinct vertex of V_λ. Let $V' \subseteq V_\lambda$ be the set of endpoints of these paths lying in V_λ. Recall that vertex set V_λ is partitioned into four contiguous subsets B_0, B_1, B_2, B_3. Therefore, at least one of these four vertex sets (say B_0) contains at least $\lfloor \beta/2 \rfloor/4$ vertices of V'. We view the paths connecting vertices of B_0 to vertices of B_2 as forming the columns of the grid, and we view the paths connecting vertices of B_1 to vertices of B_3 as forming the rows of the grid. Therefore, each column of R' corresponds to a vertex in B_0. We let $V'_0 = B_0 \cap V'$, and let $\mathcal{P}' \subseteq \hat{\mathcal{P}}$ contain all paths with an endpoint in V'_0. Lastly, we delete from R' all columns that do not correspond to vertices of V'_0 and delete arbitrary $\gamma - |V'_0|$ columns. Let R be the resulting $|V'_0| \times |V'_0|$ grid and let h be the model induced by h'. From the discussion, $|V'_0| = \Theta(\beta) = \Theta(\alpha' r/\Delta)$. For each vertex $x \in V(R)$, we select an arbitrary vertex of $h(x)$ as v_x. Denoting by I the set of vertices in last row of R, it is easy to see that each path of $\mathcal{P}'$ that connects a vertex of S to a vertex of B_0 can be extended to a path that connects a vertex of S to a vertex of $\{v_x \mid x \in I\}$, by concatenating it with a subpath of $\hat{\mathcal{P}}_0$. We denote by $\mathcal{P}$ the resulting paths obtained from $\mathcal{P}'$ and $\hat{\mathcal{P}}_0$. It is clear that the paths in $\mathcal{P}$ are edge-disjoint. This completes the construction of the grid minor R and the set $\mathcal{P}$ of edge-disjoint paths connecting S to the vertices of $\{v_x \mid x \in I\}$, thus completing the proof of Lemma 1.9.6.

CHAPTER 2

PACKING LOW-DIAMETER SPANNING TREES

2.1 Introduction

Edge connectivity of a graph is one of the most basic graph theoretic parameters, with various applications to network reliability and information dissemination. A key tool for leveraging high edge connectivity of a graph is *tree packing*: a large collection of (nearly) edge-disjoint spanning trees. A celebrated result of Tutte [36] and Nash-Williams [30] shows that for every k-edge connected graph, there is a tree packing $\mathcal{T}$ containing $\lfloor k/2 \rfloor$ edge-disjoint trees. This beautiful theorem has numerous algorithmic applications since it was first proved.

As the diameter of a graph is a central graph measure (e.g., the diameter determines the round complexity of distributed algorithms for various central graph problems, including minimum spanning tree, global minimum cut, shortest s-t path, and so on), it is important to obtain a tree packing where each tree has a small diameter. Unfortunately, the Tutte-Nash-Williams Theorem provides no guarantee on the diameter of the individual trees in $\mathcal{T}$. In the worst case, trees in $\mathcal{T}$ may have diameter that is as large as $\Omega(|V(G)|)$, even if the diameter of the original graph G is very small. A recent work of Ghaffari and Kuhn [19] shows that for any large enough n and any $k \geq 1$, there is a k-edge-connected n-vertex graph of diameter $\Theta(\log n)$, such that, in any partitioning of the graph into spanning subgraphs, all but $O(\log n)$ of the subgraphs have diameter $\Omega(n/k)$. In light of this result, it is natural to consider the following key question:

> *Is it possible to compute a tree packing whose diameter is* sublinear *in n, provided that the diameter of the input graph is* sublogarithmic *in n?*

Our second key question aims at crystallizing the main challenge to computing low-diameter tree packing. So far, we have compared the diameter of the tree packing to the diameter of the original graph. However, as observed above, the results of [19] indicate that there may be a large gap between these two measures, even for graphs whose diameter is logarithmic in n. A more natural reference point is the following. We say that a graph G is (k, D)*-connected*, iff for every pair $u, v \in V(G)$ of distinct vertices, there are k edge-disjoint paths connecting u to v in G, such that the length of each path is bounded by D. Clearly, if there is a tree packing of edge-disjoint trees of diameter at most D into G, then G must be (k, D)-connected. The question is whether the reverse is also true, if we allow a small congestion and a small slack in the diameter of the trees. The celebrated result of Tutte and Nash-Williams shows that, if every pair of vertices in G has k edge-disjoint paths connecting them, then there are $\lfloor k/2 \rfloor$ edge-disjoint spanning trees in G. However, this result is not length-preserving, in the sense that the tree paths may be much longer than the original paths connecting pairs of vertices. Our goal is then to provide such a length-preserving transformation from collections of short edge-disjoint paths connecting pairs of nodes in G to a low-diameter tree packing.

> *Given a (k, D)-connected graph G, can one obtain a tree packing of $\widetilde{\Omega}(k)$ trees of diameter $\widetilde{O}(D)$ into G, with small edge-congestion?*

In the second part of this thesis, we answer both questions in the affirmative. For the first question, we show two efficient algorithms, that, given a k-edge connected n-vertex graph G of diameter at most D, construct a low-diameter tree packing. We complement this result by an almost matching lower bound. We address the second question by providing an efficient algorithm, that, given a (k, D)-connected graph G, computes a collection of k spanning trees of diameter at most $O(D \log n)$ each, that cause edge-congestion of $O(\log n)$. We then show several applications of these structural results on distribution computation.

2.1.1 Our Results

Our graph-theoretic results consider two main settings: in the first setting, the input graph is k-edge connected, and has diameter at most D; in the second setting, the input graph is

(k, D)-connected.

Packing Trees into Low-Diameter Graphs. We prove the following two theorems that allow us to pack low-diameter trees into low-diameter graphs.

Theorem 2.1.1. *There is an efficient randomized algorithm, that, given any positive integers D, n, k, and an n-vertex k-edge-connected graph G of diameter at most D, computes a collection $\mathcal{T}' = \{T'_1, \ldots, T'_{\lfloor k/2 \rfloor}\}$ of $\lfloor k/2 \rfloor$ spanning trees of G, such that each edge of G appears in at most two of the trees in $\mathcal{T}'$, and, with high probability, each tree $T'_i \in \mathcal{T}'$ has diameter $O((101k \ln n)^D)$.*

As we show later, the diameter bound of Theorem 2.1.1 is nearly the best possible. Unfortunately, the trees in the packing given by Theorem 2.1.1 may share edges. Next, we generalize the classical result of Karger [22] to obtain a packing of completely edge-disjoint trees of small diameter, in the following theorem.

Theorem 2.1.2. *There is an efficient randomized algorithm that, given an n-vertex k-edge-connected graph G of diameter at most D, such that $k > 1000 \ln n$, computes a collection $\{T_1, \ldots, T_r\}$ of $r = \Omega(k/\ln n)$ edge-disjoint spanning trees of G, such that with probability $1 - 1/\operatorname{poly}(n)$, each resulting tree T_i has diameter $O(k^{D(D+1)/2})$.*

We note that while the diameter bound in Theorem 2.1.2 is slightly weaker than that obtained in Theorem 2.1.1, and the number of the spanning trees is somewhat lower, its advantage is that the resulting trees are guaranteed to be edge-disjoint. Moreover, the algorithm in Theorem 2.1.2 is very simple: we construct r graphs $G_1, \ldots, G_r$ with $V(G_i) = V(G)$ for all i, by sampling every edge of G into one of these graphs independently. We then compute a spanning tree T_i in each such graph G_i, and show that its diameter is suitably bounded. As such, this algorithm is easy to use in the distributed setting.

Lastly, we show that our upper bounds are close to the best possible if $k \gg D$, by proving the following lower bound.

Theorem 2.1.3. *For all positive integers n, k, D, η, α such that $k/(4D\alpha\eta)$ is an integer and $n \geq 3k \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$, there exists a k-edge connected simple graph G on n vertices of diameter at most $2D+2$, such that, for any collection $\mathcal{T} = \{T_1, \ldots, T_{k/\alpha}\}$ of k/α spanning trees of G that causes edge-congestion at most η, some tree $T_i \in \mathcal{T}$ has diameter at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$.*

Note that, in particular, any collection $\mathcal{T}$ of $\Omega(k)$ trees that are either edge-disjoint, or cause a constant edge-congestion, must contain a tree of diameter $\Omega\left(\left(\frac{k}{cD}\right)^D\right)$ for some constant c. Even if we are willing to allow a polylogarithmic edge-congestion, and to settle for $\Theta(k/\operatorname{poly}\log n)$ trees, at least one of the trees must have diameter $\Omega\left(\left(\frac{k}{D \operatorname{poly}\log n}\right)^D\right)$.

Packing Trees into (k, D)-connected Graphs. We next consider (k, D)-connected graphs and show an algorithm that computes a tree packing, that is near-optimal in both the number of trees and in the diameter.

Theorem 2.1.4. *There is an efficient randomized algorithm, that, given any positive integers D, k, n with $k \leq n$, and a (k, D)-connected n-vertex graph G, computes a collection $\mathcal{T} = \{T_1, \ldots, T_k\}$ of k spanning trees of G, such that, for each $1 \leq \ell \leq k$, tree T_ℓ has diameter at most $O(D \log n)$, and with probability at least $1 - 1/\operatorname{poly}(n)$, each edge of G appears in $O(\log n)$ trees of $\mathcal{T}$.*

2.1.2 Organization

We start with some basic definitions and notations in Section 2.2. We provide the proof of Theorem 2.1.1 in Section 2.3, the proof of Theorem 2.1.2 in Section 2.4, the proof of Theorem 2.1.3 in Section 2.5, and the proof of Theorem 2.1.4 in Section 2.6. We then discuss applications of our graph-theoretic results to distributed computation in Section 2.7.

2.2 Preliminaries

We continue to use the notations introduced in Section 1.2, and introduce some new notations and definitions that will be used in the second part of the thesis here.

Let $G = (V, E)$ be a graph. For a pair $u, v \in V(G)$ of vertices of G, we denote by $\mathsf{dist}_G(u, v)$ the length of the shortest path connecting u to v in G, and we denote by $\mathrm{diam}(G)$ the diameter of G, namely $\mathrm{diam}(G) = \max_{u,v \in V} \mathsf{dist}_G(u, v)$. For a path P in G, we denote by $|P|$ its length, that is, the number of edges in P.

We say that two paths P, P' are *edge-disjoint*, iff $E(P) \cap E(P') = \emptyset$. We say that two paths P, P' are *internally disjoint*, iff for every vertex $v \in V(P) \cap V(P')$, v is an endpoint of both paths. Given a set $\mathcal{P} = \{P_1, \ldots, P_r\}$ of paths of G, we say that the paths of $\mathcal{P}$ are *edge-disjoint* iff every edge of G belongs to at most one path of $\mathcal{P}$, and we say that the paths of $\mathcal{P}$ are *internally disjoint* iff every pair of paths in $\mathcal{P}$ are internally disjoint. We say that the set $\mathcal{P}$ of paths causes *congestion* η iff every edge $e \in E(G)$ belongs to at most η paths in $\mathcal{P}$.

Let T be a tree rooted at r. For each integer $i \geq 0$, we say that a node $v \in V(T)$ is at the ith level of T if the length of the unique path connecting v to r in T is i. We let $V_i(T)$ be the set of all nodes that lie on the ith level of the tree T, and we denote $V_{\leq i}(T) = \bigcup_{t=0}^{i} V_t(T)$. Therefore, the root lies at level 0, the children of the root are at level 1 and so on. For a collection $\mathcal{T} = \{T_1, \ldots, T_r\}$ of spanning trees of G, we say that the trees of $\mathcal{T}$ are *edge-disjoint* if every edge of G belongs to at most one tree of $\mathcal{T}$. We say that the set $\mathcal{T}$ of trees causes *congestion* η iff every edge $e \in E(G)$ belongs to at most η trees in $\mathcal{T}$.

Flows and cuts. Let $\mathcal{P}$ be the set of all paths in G. A *flow* f in G is defined to be an assignment of non-negative values $\{f(P)\}_{P \in \mathcal{P}}$ to all paths $P \in \mathcal{P}$. A path $P \in \mathcal{P}$ is called a *flow-path* of F iff $f(P) > 0$. The *value* of the flow f is $\sum_{P \in \mathcal{P}} f(P)$. Let P be a flow-path that originates at $u \in V(G)$ and terminates at $u' \in V(G)$. We say that the node u *sends*

$f(P)$ units of flow to u' along the path P. For each edge $e \in E(G)$, we define the congestion of the flow f on the edge e to be $\sum_{P \in \mathcal{P}: e \in P} f(P)$, namely the total amount of flow of f through e. The total congestion of flow f is the maximum congestion of f on any edge of G. A *cut* in a graph G is a bipartition of its vertex set V into non-empty subsets. The *value* of a cut $(S, V \setminus S)$ is $|E_G(S, V \setminus S)|$.

2.3 Low-Diameter Tree Packing with Small Edge-Congestion: Proof of Theorem 2.1.1

In this section we provide the proof of Theorem 2.1.1.

We start by showing that, if we are given a graph G, and a collection $\{T_1, \ldots, T_k\}$ of edge-disjoint spanning trees of G, such that the diameter of the tree T_k is at most $2D$ (but other trees may have arbitrary diameters), then we can efficiently compute another collection $\left\{T'_1, \ldots, T'_{k-1}\right\}$ of edge-disjoint spanning trees of G, such that the diameter of each resulting tree T'_i is bounded by $O((101k \ln n)^D)$ with high probability.

Theorem 2.3.1. *There is an efficient randomized algorithm, that, given any positive integers D, k, n, an n-vertex graph G, and a collection $\{T_1, \ldots, T_k\}$ of k spanning trees of G, such that the trees $T_1, \ldots, T_{k-1}$ are edge-disjoint, and the diameter of T_k is at most $2D$, computes a collection $\{T'_1, \ldots, T'_{k-1}\}$ of edge-disjoint spanning trees of G, such that, with probability at least $1 - 1/\operatorname{poly}(n)$, for each $1 \leq i \leq k-1$, the diameter of tree T'_i is bounded by $O((101k \ln n)^D)$.*

Theorem 2.1.1 easily follows by combining Theorem 2.3.1 with the results of Kaiser [21], who gave a short elementary proof of the tree-packing theorem of Tutte [36] and Nash-Williams [30]. His proof directly translates into an efficient algorithm, that, given a k-edge connected graph G, computes a collection of $\lfloor k/2 \rfloor$ edge-disjoint spanning trees of G. In order to complete the proof of Theorem 2.1.1, we use the algorithm of Kaiser [21] to compute an arbitrary collection $\mathcal{T} = \left\{T_1, \ldots, T_{\lfloor k/2 \rfloor}\right\}$ of edge-disjoint spanning trees of G, and compute another arbitrary BFS tree T^* of G. Since the diameter of G is at most D, the diameter of T^* is at most $2D$. We then apply Theorem 2.3.1 to the collection $\left\{T_1, \ldots, T_{\lfloor k/2 \rfloor}, T^*\right\}$ of spanning trees, to obtain another collection $\mathcal{T}' = \left\{T'_1, \ldots, T'_{\lfloor k/2 \rfloor}\right\}$ of spanning trees, such that each edge of G belongs to at most 2 trees of $\mathcal{T}'$, and with high probability, the diameter of each tree in $\mathcal{T}'$ is at most $O((101k \ln n)^D)$. We note that, since we allow parallel edges, the trees in the set $\left\{T_1, \ldots, T_{\lfloor k/2 \rfloor}, T^*\right\}$ are edge-disjoint in graph $G \cup E(T^*)$.

The main technical tool that we use in order to prove of Theorem 2.3.1 is the following theorem, that allows one to "fix" a diameter of a connected graph using a low-diameter tree.

Theorem 2.3.2. *Let H be a connected graph with $|V(H)| \leq n$, and let T be a rooted tree of depth D, such that $V(T) = V(H)$. For a real number $0 < p < 1$, let R be a random subset of the edges of T, where each edge $e \in E(T)$ is added to R independently with probability p. Then with probability at least $1 - \frac{D}{n^{48}}$, the diameter of the graph $H \cup R$ is at most $(\frac{101 \ln n}{p})^D$.*

Theorem 2.3.1 easily follows from Theorem 2.3.2: For each $1 \leq i < k$, we construct a graph G_i as follows. Start with $G_i = T_i$ for all $1 \leq i \leq k$. Compute a random partition $E_1, \ldots, E_{k-1}$ of the edges of $E(T_k)$, by adding each edge $e \in E(T_k)$ to a set E_i chosen uniformly at random from $\{E_1, \ldots, E_{k-1}\}$ independently from other edges. Using Theorem 2.3.2 with $p = 1/(k-1)$, it is immediate to see that with high probability, the diameter of each resulting graph G_i is bounded by $O((101k \ln n)^D)$. We then let T'_i be a BFS tree of graph G_i, rooted at an arbitrary vertex. In order to complete the proof of Theorem 2.1.1, it is now enough to prove Theorem 2.3.2.

Proof of Theorem 2.3.2. Recall that we are given a connected graph H with $|V(H)| \leq n$, and a rooted tree T of depth D, such that $V(T) = V(H)$, together with a parameter $0 < p < 1$. We let R be a random subset of $E(T)$, where each edge $e \in E(T)$ is added to R independently with probability p. Our goal is to show that the diameter of the graph $H \cup R$ is at most $\left(\frac{101 \ln n}{p}\right)^D$ with probability at least $1 - \frac{D}{n^{48}}$. Denote $V = V(H) = V(T)$. For each $0 \leq i \leq D$, let V_i be the set of nodes lying at level i of the tree T (that is, at distance i from the tree root), and denote $V_{\leq i} = \bigcup_{t=0}^{i} V_t$. Let $H' = H \cup R$.

We say that a node $x \in V$ is *good* if either (i) $x \in V_{\leq D-1}$; or (ii) $x \in V_D$, and there is an edge in R connecting x to a node in V_{D-1}. We assume that $V = \{v_1, \ldots, v_{n'}\}$, where the vertices are indexed in an arbitrary order. Given an ordered pair (x, x') of vertices in H, and a path P connecting x to x', let $\sigma(P)$ be a sequence of vertices that lists all the vertices appearing on P in their natural order, starting from vertex x (so in a sense, we think of P as a directed path). For an ordered pair $(x, x') \in V$ of vertices, let $P_{x,x'}$ be shortest path connecting x to x' in H, and among all such paths P, choose the one whose sequence $\sigma(P)$ is smallest lexicographically. Observe that $P_{x,x'}$ is unique, and, moreover, if some pair u, u' of vertices lie on $P_{x,x'}$, with u lying closer to x than u' on $P_{x,x'}$, then the sub-path of $P_{x,x'}$ from u to u' is precisely $P_{u,u'}$.

Let $M = \frac{50 \ln n}{p}$. For a pair x, x' of vertices of V, we let $B(x, x')$ be the bad event that length of $P_{x,x'}$ is greater than M and there is no good internal node on $P_{x,x'}$. Notice that event $B(x, x')$ may only happen if every inner vertex on $P_{x,x'}$ lies in V_D, and for each such vertex, the unique edge of T that is incident to it was not added to R. Therefore, the probability that event $B(x, x')$ happens for a fixed pair x, x' of vertices is at most $(1-p)^M = (1-p)^{(50 \ln n)/p} \leq n^{-50}$. Let B be the bad event that $B(x, x')$ happens for some pair $x, x' \in V$ of nodes. From the union bound over all pairs of nodes in V, the probability of B is bounded by n^{-48}.

Recall that H is a subgraph of H' and $\mathsf{dist}_H(\cdot, \cdot)$ is the shortest-path distance metric on H. We use the following immediate observation.

Observation 2.3.3. *If the event B does not happen, then for every node $x \in V$, there is a good node $x' \in V$ such that $\mathsf{dist}_H(x, x') \leq M$.*

We prove Theorem 2.3.2 by induction on D. The base of the induction is when $D = 1$. In this case, T is a star graph. Let c denote the vertex that serves as the center of the star. For any pair $x_1, x_2 \in V$ of vertices, we denote by x'_1 the good node that is closest to x_1 in H, and we define x'_2 similarly for x_2. Notice that, from the definition of good vertices, either $x'_1 = c$, or it is connected to c by an edge of R, and the same holds for x'_2. Therefore, $\mathsf{dist}_{H'}(x'_1, x'_2) \leq 2$ must hold. If the event B does not happen, then, since H is a subgraph of H', $\mathsf{dist}_{H'}(x_1, x_2) \leq \mathsf{dist}_{H'}(x_1, x'_1) + \mathsf{dist}_{H'}(x'_1, x'_2) + \mathsf{dist}_{H'}(x_2, x'_2) \leq \mathsf{dist}_{H}(x_1, x'_1) + \mathsf{dist}_{H'}(x'_1, x'_2) + \mathsf{dist}_{H}(x_2, x'_2) \leq 2M + 2 \leq \frac{101 \ln n}{p}$. Therefore, with probability at least $1 - n^{-48}$, $\mathsf{dist}_{H'}(x_1, x_2) \leq \frac{101 \ln n}{p}$.

Assume now that Theorem 2.3.2 holds for every connected graph H and every tree T of depth at most $D - 1$, with $V(T) = V(H)$. Consider now some connected graph H, and a rooted tree T of depth D, with $V(T) = V(H)$. We partition the edges of $E(T)$ into two subsets: set E_1 contains all edges incident to the vertices of V_D, and set E_2 contains all remaining edges. Let $E'_1 = E_1 \cap R$, and let $E'_2 = E_2 \cap R$. Notice that the definition of good vertices only depends on the edges of E'_1, and so the event B only depends on the random choices made in selecting the edges of E'_1, and is independent from the random choices made in selecting the edges of E'_2.

Let L be a subgraph of H', obtained by starting with $L = H$, and then adding all edges of E'_1 to the graph. Finally, we define a new graph $\hat{H}$, whose vertex set is $V_{\leq D-1}$, and there is an edge between a pair of nodes w, w' in $\hat{H}$ iff the distance between w and w' in L is at most $M + 2$. We also let $\hat{T}$ be the tree obtained from T, by discarding from it all vertices of V_D and all edges incident to vertices of V_D. Observe that $V(\hat{H}) = V(\hat{T}) = V_{\leq D-1}$. The idea is to use the induction hypothesis on the graph $\hat{H}$, together with the tree $\hat{T}$. In order to do so, we need to prove that $\hat{H}$ is a connected graph, which we do next.

Observation 2.3.4. *If the event B does not happen, then graph $\hat{H}$ is connected.*

Proof. Assume that the event B does not happen, and assume for contradiction that graph $\hat{H}$ is not connected. Let $\mathcal{C} = \{C_1, \ldots, C_r\}$ be the set of all connected components of graph $\hat{H}$. For every pair C_i, C_j of distinct components of $\mathcal{C}$, consider the set $\mathcal{P}_{i,j} =$

$\left\{P_{x,x'} \mid x \in V(C_i), x' \in V(C_j)\right\}$ of paths (recall that $P_{x,x'}$ is the shortest path connecting x to x' in H with $\sigma(P_{x,x'})$ lexicographically smallest among all such paths). We let $P_{i,j}$ be a shortest path in $\mathcal{P}_{i,j}$. Choose two distinct components $C_i, C_j \in \mathcal{C}$, whose path $P_{i,j}$ has the shortest length, breaking ties arbitrarily. Assume that $P_{i,j}$ connects a vertex $v \in C_i$ to a vertex $u \in C_j$, so $P_{i,j} = P_{v,u}$. Recall that $H \subseteq L$, and so the path $P_{i,j}$ is contained in graph L. Since we did not add edge (u,v) to $\hat{H}$, the length of $P_{i,j}$ is greater than $M+2$. Since we have assumed that event B does not happen, there is at least one good inner vertex on path $P_{i,j}$. Let X be the set of all good vertices that serve as inner vertices of $P_{i,j}$.

We first show that for each $x \in X$, $x \notin V(\hat{H})$ must hold. Indeed, assume for contradiction that $x \in V(\hat{H})$, so x belongs to some connected component of $V(\hat{H})$. Assume first that $x \in V(C_i)$. Recall that the sub-path of $P_{i,j}$ from x to u is precisely $P_{x,u}$, so this path lies in $\mathcal{P}_{i,j}$. But its length is less than the length of $P_{i,j}$, contradicting the choice of $P_{i,j}$. Otherwise, x belongs to some connected component C_ℓ of $\mathcal{C}$ with $\ell \neq i$. The sub-path of $P_{i,j}$ from v to x is precisely $P_{v,x}$, so this path must lie in $\mathcal{P}_{i,\ell}$. Since its length is less than the length of $P_{i,j}$, this contradicts the choice of the components C_i, C_j. We conclude that $x \notin V(\hat{H})$.

Since $V(\hat{H})$ contains all vertices of $V_{\leq D-1}$, and every vertex in X is a good vertex, it must be the case that $X \subseteq V_D$. Consider again some vertex $x \in X$. Since x is a good vertex and $x \in V_D$, there must be an edge $e_x = (x, x') \in E_1'$, connecting x to some vertex $x' \in V_{\leq D-1}$. In particular, x' must belong to some connected component of $\mathcal{C}$, and the edge e_x lies in graph L. Assume that $X = \left\{x_1, x_2, \ldots, x_q\right\}$, where the vertices are indexed in the order of their appearance on $P_{i,j}$, from v to u. Consider the sequence $\tilde{\sigma} = (v, x_1', x_2', \ldots, x_q', u)$ of vertices. All these vertices belong to $V(\hat{H})$, and $v \in C_i$, while $u \in C_j$. For convenience, denote $v = x_0' = x_0$ and $u = x_{q+1}' = x_{q+1}$. Then there must be an index $1 \leq a \leq q$, such that x_a' and x_{a+1}' belong to distinct connected components of $\mathcal{C}$. Note that the sub-path of $P_{i,j}$ between x_a and x_{a+1} is precisely $P_{x_a, x_{a+1}}$ – the shortest path connecting x_a to x_{a+1} in H. Since no good vertices lie between x_a and x_{a+1} on this path, and since we have assumed that event B does not happen, the length of this path is at most M. Therefore, there is a path in graph L, connecting x'_a to x'_{a+1}, whose length is at most $M+2$. This path connects a pair of vertices that belong to different connected components of $\hat{H}$, contradicting the construction of $\hat{H}$. $\square$

Consider now the tree $\hat{T}$ and the graph $\hat{H}$. Recall that $\hat{T}$ is a rooted tree of depth $D-1$, $V(\hat{T}) = V(\hat{H})$, $|V(\hat{H})| \leq |V(H)| \leq n$, and, assuming the event B did not happen, $\hat{H}$ is a connected graph. Moreover, set E'_2 of edges is a subset of $E(\hat{T}) = E_2$, obtained by adding every edge of $E(\hat{T})$ to E'_2 with probability p, independently from other edges. Therefore, assuming that event B did not happen, we can use the induction hypothesis on the graph $\hat{H}$, the tree $\hat{T}$, and the set E'_2 of edges as R. Let B' be the bad event that the diameter of $\hat{H} \cup E'_2$ is greater than $(\frac{101 \ln n}{p})^{D-1}$. Note that the event B' only depends on the random choices made in selecting the edges of E'_2. From the induction hypothesis, the probability that B' happens is at most $\frac{D-1}{n^{48}}$.

Lastly, we show that, if neither of the events B, B' happens, then $\mathrm{diam}(H') \leq (\frac{101 \ln n}{p})^D$.

Observation 2.3.5. *If neither of the events B, B' happens, then* $\mathrm{diam}(H') \leq (\frac{101 \ln n}{p})^D$.

Proof. Consider any pair $x_1, x_2 \in V$ of vertices. It is sufficient to show that, if events B, B' do not happen, then $\mathsf{dist}_{H'}(x_1, x_2) \leq (\frac{101 \ln n}{p})^D$.

Let x'_1 be a good node in $V(H)$ that is closest to x_1, and define x'_2 similarly for x_2. From Observation 2.3.3, $\mathsf{dist}_H(x_1, x'_1) \leq M$. If $x'_1 \in V_{\leq D-1}$, then we define $x''_1 = x'_1$, otherwise we let x''_1 be the node of V_{D-1} that is connected to x'_1 by an edge of E'_1, and we define x''_2 similarly for x_2. Therefore, $x''_1, x''_2 \in V_{\leq D-1} = V(\hat{H})$, and, assuming event B does not happen, $\mathsf{dist}_{H'}(x_1, x''_1) \leq M+1$, and $\mathsf{dist}_{H'}(x_2, x''_2) \leq M+1$. Since we have assumed that the bad event B' does not happen, $\mathsf{dist}_{\hat{H} \cup E'_2}(x''_1, x''_2) \leq (\frac{101 \ln n}{p})^{D-1}$. Recall that for every edge $e = (u, v) \in \hat{H} \cup E'_2$, if $e \in E'_2$ then $e \in E(H')$; otherwise, $e \in E(\hat{H})$, and there is a path in graph $H \cup E'_1$ of length at most $M+2$ connecting u to v in H. Therefore, $\mathsf{dist}_{H'}(x''_1, x''_2) \leq (M+2) \cdot \mathsf{dist}_{\hat{H}}(x''_1, x''_2) \leq (\frac{101 \ln n}{p})^{D-1} \cdot (M+2)$.

Altogether, since $M = (50 \ln n)/p$,

$$
\begin{aligned}
\mathsf{dist}_{H'}(x_1, x_2) &\le \mathsf{dist}_{H'}(x_1, x_1'') + \mathsf{dist}_{H'}(x_1'', x_2'') + \mathsf{dist}_{H'}(x_2, x_2'') \\
&\le \left(\frac{101 \ln n}{p}\right)^{D-1} \cdot (M+2) + (2M+2) \\
&\le \left(\frac{101 \ln n}{p}\right)^{D}.
\end{aligned}
$$

□

The probability that either B or B' happen is bounded by $\frac{D}{n^{48}}$. Therefore, with probability at least $1 - \frac{D}{n^{48}}$, neither of the events happens, and $\mathrm{diam}(H') \le (\frac{101 \ln n}{p})^D$. This concludes the proof of Theorem 2.3.2. □

2.4 Low-Diameter Packing of Edge-Disjoint Trees: Proof of Theorem 2.1.2

In this section we provide the proof of Theorem 2.1.2. The main tool in the proof of Theorem 2.1.2 is the following theorem.

Theorem 2.4.1. *Let k, D, n be any positive integers with $k > 1000 \ln n$, let $\frac{707 \ln n}{k} \le p \le 1$ be a real number, and let G be an n-vertex k-edge-connected graph of diameter D. Let G' be a sub-graph of G with $V(G') = V(G)$, where every edge $e \in E(G)$ is added to G' with probability p independently from other edges. Then, with probability at least $1 - 1/\mathrm{poly}(n)$, G' is a connected graph, and its diameter is bounded by $k^{D(D+1)/2}$.*

Karger [22] has shown that, if G is a k-connected graph, and G' is obtained by sub-sampling the edges of G with probability $\Omega(\log n/k)$, then G' is a connected graph with high probability. Theorem 2.4.1 further shows that the diameter of G' is with high probability bounded by $k^{D(D+1)/2}$, where D is the diameter of G.

Theorem 2.1.2 easily follows from Theorem 2.4.1: Let $r = \lfloor k/(707 \ln n) \rfloor$. We partition $E(G)$ into subsets $E_1, \ldots, E_r$ by choosing, for each edge $e \in E(G)$, an index i independently and uniformly at random from $\{1, 2, \ldots, r\}$ and then adding e to E_i. For each $1 \le i \le r$, we define a graph G_i by setting $V(G_i) = V(G)$ and $E(G_i) = E_i$. Finally, for each graph G_i, we compute an arbitrary BFS tree T_i, and return the resulting collection $\mathcal{T} = \{T_1, \ldots, T_r\}$ of trees. It is immediate to verify that the graphs $G_1, \ldots, G_r$ are edge-disjoint, and so are the trees of $\mathcal{T}$. Moreover, applying Theorem 2.4.1 to each graph G_i with $p = 1/r$, we get that with probability $1 - 1/\operatorname{poly}(n)$, $\operatorname{diam}(T_i) \le 2 \operatorname{diam}(G_i) \le O(k^{D(D+1)/2})$. Using the union bound over all $1 \le i \le r$ completes the proof of Theorem 2.1.2. It now remains to prove Theorem 2.4.1.

2.4.1 Bounding the Diameter of a Random Subgraph: Proof of Theorem 2.4.1

This subsection is dedicated to proving Theorem 2.4.1. We assume that we are given an n-vertex k-edge connected graph $G = (V, E)$, with $k > 1000 \ln n$, and a parameter $\frac{707 \ln n}{k} \le p \le 1$. Our goal is to show that a random graph G', obtained by independently sub-sampling every edge of G with probability p, has diameter at most $k^{D(D+1)/2}$ with probability at least $1 - 1/\operatorname{poly}(n)$.

Let B be the bad event that the graph G' is not connected. We start by establishing that B only happens with low probability, using a well known result of Karger [22].

Claim 2.4.2. *The probability that the event B happens is at most $O(1/n^{10})$.*

Proof. For a real number $p \in [0, 1]$, let $\mathcal{D}(G, p)$ be the distribution of graphs, where the vertex set of the resulting graph is $V(G)$, and each edge of G is included in the graph with probability p independently from other edges. We use the following result of Karger [22].

Theorem 2.4.3 (Adaptation of Theorem 2.1 from [22])**.** *Let k, n be any positive integers,* *and let d, p be any positive real numbers such that $0 < p < 1$. Let G be an n-vertex k-edge connected graph. Let $G' \sim \mathcal{D}(G, p)$ be a random subgraph of G and let $\epsilon = \sqrt{\frac{3(d+2)\ln n}{kp}}$. If $\epsilon < 1$ then, with probability $1 - O(1/n^d)$, every cut in G' has value between $(1+\epsilon)$ and $(1-\epsilon)$ times its expected value.*

We apply Theorem 2.4.3 to the graph G, with the parameter p and $d = 10$. Since G is k-edge connected and $p \geq (707 \ln n)/k$, we get that:

$$\epsilon = \sqrt{\frac{3(d+2)\ln n}{kp}} \leq \sqrt{\frac{36 \ln n}{k \cdot (707 \ln n)/k}} \leq \sqrt{\frac{36}{707}} < 0.3 < 1.$$

Therefore, with probability $1 - O(1/n^{10})$, for every cut $(S, V \setminus S)$ in G', $|E_{G'}(S, V \setminus S)| \geq (1-\epsilon) \cdot p \cdot |E_G(S, V \setminus S)| \geq 0.7 \cdot pk > 0$. Therefore, with probability $1 - O(1/n^{10})$, graph G' is connected, and event B happens with probability $O(1/n^{10})$. $\square$

We now proceed to bound the diameter of G'. Denote $G = (V, E)$, and let T be a BFS tree of G, rooted at an arbitrary node of G. Since G has diameter at most D, the depth of T is at most D. For each integer $0 \leq i \leq D$, we denote by V_i the set of nodes that lie at the ith level of T (recall that the root lies at level 0), and we denote $V_{\leq i} = \bigcup_{j=0}^{i} V_j$. For each $0 \leq i \leq D-1$, let E_i be the set of edges of T connecting vertices of V_i to vertices of V_{i+1}. We also let $E_{\text{out}} = E \setminus E(T)$, so $E = E_{\text{out}} \cup \left(\bigcup_{i=0}^{D-1} E_i\right)$.

Recall that $G' \sim \mathcal{D}(G, p)$. We first define a different (but equivalent) sampling algorithm for generating a random graph G' from the distribution $\mathcal{D}(G, p)$. We will then use this algorithm to bound the diameter of G'. The algorithm consists of $D+1$ phases. For each $0 \leq i \leq D$, we compute a random subgraph G'_i of G, with $V(G'_i) = V(G)$, such that $G'_0 \subseteq G'_1 \subseteq \cdots \subseteq G'_D$. The final graph G'_D is denoted by G'. For all $0 \leq i \leq D$, we denote by $\mathcal{C}_i$ the set of all connected components of the graph G'_i. Throughout the algorithm, we maintain a set $\hat{E}$ of edges, that is initialized to $\emptyset$.

In order to execute the 0th phase, we consider the edges of E_{out}. Each such edge is added to the set $\hat{E}$ with probability p independently from other edges. Let $E'_{\mathrm{out}} \subseteq E_{\mathrm{out}}$ be the set of edges that are added to $\hat{E}$ in this phase. We then set $G'_0 = (V, E'_{\mathrm{out}})$. Observe that G'_0 may not be a connected graph. We denote by $\mathcal{C}_0$ the set of all connected components of G'_0. We refer to the connected components of $\mathcal{C}_0$ as *phase*-0 *clusters*.

For each $1 \leq i \leq D$, in order to execute the ith phase, we consider the set E_{D-i} of edges. Each such edge is added to $\hat{E}$ with probability p independently from other edges. We denote by $E'_{D-i} \subseteq E_{D-i}$ the set of edges that are added to $\hat{E}$ at phase i. Graph G'_i is obtained from the graph G'_{i-1} by adding all edges of E'_{D-i} to it. As before, we denote by $\mathcal{C}_i$ the set of all connected components of G'_i, and we call them *phase-i clusters*.

Let E' be the set $\hat{E}$ at the end of this algorithm. We denote by $G' = (V, E')$ the final graph that we obtain. Clearly, $G' = G'_D$, and it is generated from the distribution $\mathcal{D}(G, p)$, since $E = E_{\mathrm{out}} \cup \left(\bigcup_{i=0}^{D-1} E_i\right)$, and the edge sets $E_{\mathrm{out}}, E_0, \ldots, E_{D-1}$ are mutually disjoint. We denote by T' the subgraph of T with $V(T') = V(T)$ and $E(T') = \bigcup_{i=0}^{D-1} E'_i$. Observe that $T' \sim \mathcal{D}(T, p)$.

Consider a pair $u, u' \in V$ of distinct vertices. We say that u and u' are *joined at phase* 0, if they belong to the same connected component of G'_0. We say that they are *joined at phase* i for $1 \leq i \leq D$, if u and u' belong to the same connected component of G'_i but they lie in different connected components of G'_{i-1}. For all $0 \leq i \leq D$, let Π_i denote the set of all pairs of vertices that joined at phase i. Note that, if the event B does not happen, then every pair (u, u') of distinct vertices of V lies in a unique set Π_i, for some $0 \leq i \leq D$.

In order to bound the distances between pairs of nodes in G', we need the following theorem, that slightly generalizes Theorem 2.3.2. The proof is similar to that of Theorem 2.3.2 and is deferred to Section 2.4.2.

Theorem 2.4.4. *Let T be a rooted tree of depth D with $|V(T)| \leq n$, and let H be a connected graph with $V(H) \subseteq V(T)$. For a real number $0 < p < 1$, let $R \sim \mathcal{D}(T, p)$ be a random subgraph of T, so $V(R) = V(T)$, and every edge of $E(T)$ is added to $E(R)$ independently* *with probability p. Then with probability at least $1-\frac{D}{n^{48}}$, for every pair u, v of vertices of H,* $\mathsf{dist}_{R\cup H}(u,v) \le (\frac{101\ln n}{p})^D$.

We use a parameter $N = (101\ln n)/p$. Since $p \ge (707\ln n)/k$, we get that $7N \le k$. For each $0 \le i \le D$, we define a distance threshold M_i, as follows. We let $M_0 = N^D$, and for all $1 \le i \le D$, we let $M_i = 7N^{D-i}\cdot M_{i-1}$. It is easy to verify that, for all $0 \le i \le D$:

$$M_i \le 7^i N^{D+(D-1)+\cdots+D-i} \le (7N)^{D(D+1)/2} \le k^{D(D+1)/2}.$$

For each $0 \le i \le D$, we say that a bad event B_i happens, if for some pair $(u,u') \in \Pi_0\cup\cdots\cup\Pi_i$ of distinct vertices, the distance between u and u' in G' is greater than M_i. The following lemma is central to the proof of Theorem 2.4.1.

Lemma 2.4.5. *For each $0 \le i \le D$, the probability of event B_i is at most i/n^{43}.*

Observe that, if none of the events $B, B_0, \ldots, B_D$ happen, then G' is a connected graph, and in particular, every pair (u,u') of distinct vertices of G belongs to some set Π_i, for some $0 \le i \le D$, so $\mathsf{dist}_{G'}(u,u') \le k^{D(D+1)/2}$. Using the union bound, the probability that at least one of the events $B, B_0, \ldots, B_D$ happens is bounded by $O(1/n^{10})$. Therefore, with probability at least $1-O(1/n^{10})$, graph G' is connected, and $\mathrm{diam}(G') \le k^{D(D+1)/2}$. In order to complete the proof of Theorem 2.4.1, it is now enough to prove Lemma 2.4.5.

Proof of Lemma 2.4.5: The proof is by induction on i. The base case is when $i=0$. Let $(u,u') \in \Pi_0$ be any pair of vertices of G' that are joined at phase 0. Let $B_0(u,u')$ be the bad event that the distance from u to u' in G' is greater than $M_0 = N^D$. Clearly, event B_0 may only happen if event $B_0(u,u')$ happens for some pair $(u,u') \in \Pi_0$ of vertices. We now bound the probability of each such event separately.

Let $(u,u') \in \Pi_0$ be any pair of vertices joined at phase 0. Recall that u, u' lie in the same connected component of G'_0, and so there is some path Q connecting u to u' in G'_0. Consider now the graph Q, and the tree T that we have defined before, whose depth is bounded by

D. Recall that $T' \subseteq T$ is obtained from T by sub-sampling each of its edges independently with probability p. Using Theorem 2.4.4 with graph $H = Q$, the tree T, and the sampling probability p, we conclude that the probability that the distance from u to u' in $Q \cup T'$ is greater than $\left(\frac{101 \ln n}{p}\right)^D = N^D$ is bounded by D/n^{46}. Recall that $Q \subseteq G'_0$ and so $Q \cup T' \subseteq G'$. Therefore, $\mathsf{dist}_{G'}(u, u') \leq \mathsf{dist}_{Q \cup T'}(u, u')$, and so the probability that event $B_0(u, u')$ happens is bounded by D/n^{46}. Using the union bound over all pairs $(u, u') \in \Pi_0$ and the fact that $D \leq n$, we conclude that $\mathbf{Pr}[B_0] \leq 1/n^{43}$.

We now assume that the claim is true for all indices $0, \ldots, (i-1)$, and prove it for index i. As before, let $(u, u') \in \Pi_i$ be any pair of vertices of G' that are joined at phase i. Let $B_i(u, u')$ be the bad event that the distance from u to u' in G' is greater than M_i. Clearly, event B_i may only happen if event $B_i(u, u')$ happens for some pair $(u, u') \in \Pi_i$ of vertices, or one of the events $B_0, \ldots, B_{i-1}$ happens. We now bound the probability of each such event $B_i(u, u')$ separately.

Recall that G'_i is the graph that we have obtained at the end of phase i of the sampling algorithm. Note that G'_i is determined completely by the random choices made in phases $0, 1, \ldots, i$. Let $(u, u') \in \Pi_i$ be a pair of vertices that are joined at phase i. By the definition, u and u' belong to different phase-$(i-1)$ clusters but the same phase-i cluster. Therefore, there is some simple path Q in graph G'_i that connects u to u'. Recall that graph G'_i is obtained from the graph G'_{i-1} by adding the edges of E'_{D-i} to it – the edges that we have sampled in phase i. The edges of E'_{D-i} are sampled from the set E_{D-i} of edges, connecting vertices of V_{D-i} to vertices of V_{D-i+1}. For convenience, we denote the edges of E'_{D-i} by $\tilde{E}$. Let $Q_1, Q_2, \ldots, Q_t$ be the set of segments of Q, obtained by deleting all edges of $\tilde{E}$ from Q. Note that each such segment Q_j is contained in some phase-$(i-1)$ cluster, and $t \geq 2$, since u and u' lie in different phase-$(i-1)$ clusters. We assume that the segments are indexed by their natural order on path Q, and that $u \in Q_1$, while $u' \in Q_t$. For each $1 \leq j < t$, we let L_j be the sub-path of Q, connecting the last vertex of Q_j to the first vertex of Q_{j+1}. Notice that all edges in L_j belong to the set $\tilde{E}$, and so each such segment L_j is either a single edge of $\tilde{E}$, or it consists of two such edges, that share a common vertex in V_{D-i} (see Figure 2.1). In either case, each such segment L_j must contain a single vertex that belongs to V_{D-i}, which we denote by w_j.

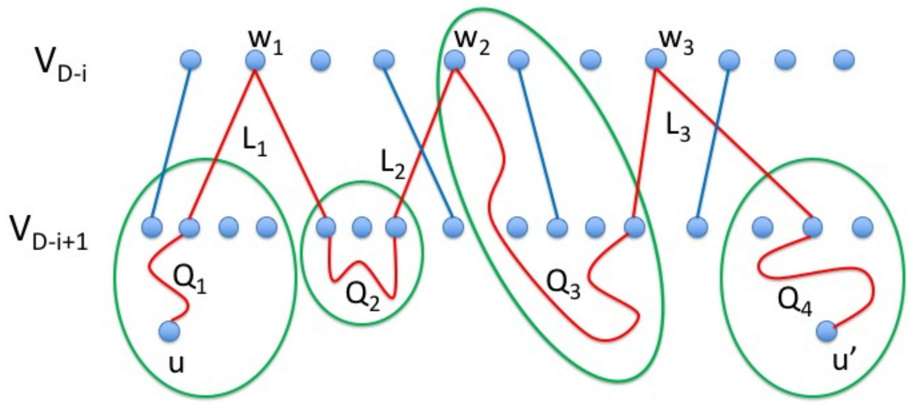


Figure 2.1: Vertices u and u' are joined at level i; the path Q is shown in red; the edges of $\tilde{E} \setminus E(Q)$ are shown in blue; the phase-$(i-1)$ clusters that share vertices with Q are shown in green.

We denote $W = \{w_1, \ldots, w_{t-1}\}$, so $W \subseteq V_{D-i}$, and we define a new graph H, whose vertex set is W, and, for each $1 \leq j \leq t-2$, there is an edge between vertex w_j and vertex w_{j+1}. Observe that H is a path, connecting the vertices of W in their natural order. Note that H is guaranteed to be a connected graph, and that it only depends on the random choices made in phases $0, \ldots, i$.

Let $\hat{T}$ be the sub-tree of T that is induced by the vertices of $V_{\leq D-i}$, and let $\hat{T}'$ be the sub-tree of $\hat{T}$ with $V(\hat{T}') = V(\hat{T})$, and $E(\hat{T}')$ containing all edges of $E'_{D-i-1} \cup \cdots \cup E'_0$. In other words, the edges of $\hat{T}'$ are all edges that were sampled in phases $(i+1), \ldots, D$ of the sampling algorithm. Observe that $\hat{T}' \sim \mathcal{D}(\hat{T}, p)$. Finally, let $H' = H \cup \hat{T}'$. We let $B'_i(u, u')$ be the bad event that the distance from w_1 to w_{t-1} in the graph H' is greater than N^{D-i}. Observe that the event $B'_i(u, u')$ only depends on random choices made in phases $(i+1), \ldots, D$. Using Theorem 2.4.4 with the graph H, the tree $\hat{T}$, and the sampling probability p, together with the fact that $N = (101 \ln n)/p$, we conclude that, the probability that the event $B'_i(u, u')$ happens is bounded by D/n^{46}. Lastly, we need the following claim.

Claim 2.4.6. *If neither of the events $B_{i-1}, B'_i(u, u')$ happens, then neither does event $B_i(u, u')$.*

Proof. Assume that neither of the events $B_{i-1}, B'_i(u, u')$ happens. We show that the distance between u and u' in G' is bounded by M_i, that is, event $B_i(u, u')$ does not happen.

Let P be the shortest path connecting w_1 to w_{t-1} in graph H'. Since we have assumed that event $B'_i(u, u')$ does not happen, $|P| \leq N^{D-i}$. We would like to turn the path P into a path P' connecting u to u' in graph G', without increasing its length by too much. Observe first that an edge $e = (v, v') \in E(P)$ must be of one of two types: either it is an edge of $\hat{T}'$, and hence it is also an edge of G'; or it is an edge of the form (w_j, w_{j+1}), in which case it may not be an edge of G'. In order to complete the proof, we show that each such edge can be replaced by a short path in G', and we show that u and u' can be connected by short paths to w_1 and w_{t-1}, respectively, in graph G'.

Observation 2.4.7. *Assume that event B_{i-1} does not happen. Then for each $1 \leq j < t-1$, there is a path P_j of length at most $2 + M_{i-1}$ in graph G', connecting vertex w_j to vertex w_{j+1}. Moreover, there is a path P_0 of length at most $1 + M_{i-1}$ in graph G' connecting u to w_1, and there is a path P_{t-1} of length at most $1 + M_{i-1}$ in graph G' connecting w_{t-1} to u'.*

Proof. From the way we have partitioned the path Q into segments, either u and w_1 lie in the same phase-$(i-1)$ cluster, or there is an edge $(v, w_1) \in \tilde{E}$, such that v lies in the same phase-$(i-1)$ cluster as u. In the former case, we also denote w_1 by v for convenience. Therefore, u and v where joined before phase i, and so $\mathsf{dist}_{G'}(u, v) \leq M_{i-1}$, by our assumption that event B_{i-1} does not happen. Therefore, there is a path in G' of length at most $M_{i-1} + 1$ that connects u to w_1. Similarly, there is a path of length at most $M_{i-1} + 1$ in graph G' connecting w_{t-1} to u'.

Consider now some index $1 \leq j < t-1$. From the definition of segments of Q, there is some phase-$(i-1)$ cluster C, and vertices $v, v' \in C$, such that: (i) either $w_j = v$, or edge $(w_j, v) \in \tilde{E}$; and (ii) either $w_{j+1} = v'$, or edge $(w_{j+1}, v') \in \tilde{E}$. In either case, $v, v' \in \Pi_{i'}$ for some $i' < i$, and, since we have assumed that event B_{i-1} does not happen, $\mathsf{dist}_{G'}(v, v') \leq M_{i-1}$. Since $\tilde{E} \subseteq E(G')$, $\mathsf{dist}_{G'}(w_j, w_{j+1}) \leq 2 + \mathsf{dist}_{G'}(v, v') \leq 2 + M_{i-1}$. $\square$

In order to obtain the desired path P', we replace each edge of the form (w_j, w_{j+1}) on path P with the corresponding path P_j, and we append P_1 and P_{t-1} to the beginning and to the end of the resulting path. It is easy to verify that $|P'| \leq |P| \cdot (M_{i-1} + 2) + 2M_{i-1} + 2 \leq |P| \cdot 7M_{i-1} \leq 7N^{D-i} M_{i-1} = M_i$. $\square$

So far we have shown that, if the events B_{i-1}, $B'_i(u, u')$ do not happen, then neither does event $B_i(u, u')$. Recall that event B_i may only happen if some event in $\{B_{i-1}\} \cup \left\{B_i(u, u') \mid (u, u') \in \Pi_i\right\}$ happens. Therefore, event B_i may only happen if some event in $\{B_{i-1}\} \cup \left\{B'_i(u, u') \mid (u, u') \in \Pi_i\right\}$ happens.

From the induction hypothesis, the probability of event B_{i-1} happening is bounded by $(i-1)/n^{43}$, and, from the previous discussion, for each $(u, u') \in \Pi_i$, the probability of the event $B'_i(u, u')$ is bounded by D/n^{46}. Taking the union bound over all these events, and using the facts that $|\Pi_i| \leq n^2$ and $D \leq n$, we conclude that the probability that any event in $\{B_{i-1}\} \cup \left\{B'_i(u, u') \mid (u, u') \in \Pi_i\right\}$ happens is bounded by i/n^{43}, and this also bounds the probability of the event B_i. $\square$

2.4.2 Proof of Theorem 2.4.4

Recall that we are given a connected graph H and a rooted tree T of depth D with $|V(T)| \leq n$ and $V(H) \subseteq V(T)$, together with a parameter p. We let R be a random subgraph of T with $V(R) = V(T)$, where every edge of $E(T)$ is added to $E(R)$ with probability p independently from other edges; in other words, $R \sim \mathcal{D}(T, p)$. Our goal is to show with probability at least

$1-\frac{D}{n^{48}}$, for every pair u, v of vertices of H, $\mathsf{dist}_{R\cup H}(u,v) \le (\frac{101\ln n}{p})^D$. The proof is a slight modification of the proof of Theorem 2.3.2. Note that the main difference between Theorem 2.4.4 and Theorem 2.3.2 is that now the tree T may contain vertices in addition to $V(H)$.

We denote $V = V(T)$. As before, for each $0 \le i \le D$, we let V_i be the set of nodes lying at level i of the tree T, and denote $V_{\le i} = \bigcup_{t=0}^{i} V_t$. We also denote $H' = H \cup R$.

We say that a node $x \in V(H)$ is *good* if either (i) $x \in V_{\le D-1} \cap V(H)$; or (ii) $x \in V_D \cap V(H)$, and there is an edge in R connecting x to a node in V_{D-1}. Let $M = \frac{50\ln n}{p}$. As before, we assume that $V(H) = \{v_1, \ldots, v_{n'}\}$, where the vertices are indexed in an arbitrary order. Given an ordered pair (x, x') of vertices in H, and a path P of H connecting x to x', let $\sigma(P)$ be a sequence of vertices that lists all the vertices appearing on P in their natural order, starting from vertex x. For an ordered pair $(x, x') \in V(H)$ of vertices, let $P_{x,x'}$ be shortest path connecting x to x' in H, and among all such paths P, choose the one whose sequence $\sigma(P)$ is smallest lexicographically. Observe that $P_{x,x'}$ is unique, and, moreover, if some pair $u, u' \in V(H)$ of vertices lie on $P_{x,x'}$, with u lying closer to x than u' on $P_{x,x'}$, then the sub-path of $P_{x,x'}$ from u to u' is precisely $P_{u,u'}$.

For a pair $x, x' \in V(H)$ of vertices of H, we let $B(x, x')$ be the bad event that length of $P_{x,x'}$ is greater than M and there is no good internal node on $P_{x,x'}$. Exactly as before, the probability that event $B(x, x')$ happens for a fixed pair x, x' of vertices is at most $(1-p)^M = (1-p)^{(50\ln n)/p} < n^{-50}$.

Let B be the bad event that $B(x, x')$ happens for some pair $x, x' \in V(H)$ of nodes. From the union bound over all pairs of distinct nodes in $V(H)$, the probability of B is bounded by n^{-48}. The following observation is an analogue of Observation 2.3.3, and its proof is identical.

Observation 2.4.8. *If the event B does not happen, then for every node $x \in V(H)$, there is a good node $x' \in V$ such that $\mathsf{dist}_H(x, x') \le M$.*

As before, we prove Theorem 2.4.4 by induction on D. The base of the induction is when

$D = 1$. In this case, T is a star graph. Let c denote the vertex that serves as the center of the star. For any pair $x_1, x_2 \in V(H)$ of vertices, we denote by x_1' the good node that is closest to x_1 in H, and we define x_2' similarly for x_2. Notice that, from the definition of good vertices, either $x_1' = c$, or it is connected to c by an edge of R, and the same holds for x_2'. Therefore, $\mathsf{dist}_{H'}(x_1', x_2') \leq 2$ must hold. If the event B does not happen, then, since H is a subgraph of H', $\mathsf{dist}_{H'}(x_1, x_2) \leq \mathsf{dist}_{H'}(x_1, x_1') + \mathsf{dist}_{H'}(x_1', x_2') + \mathsf{dist}_{H'}(x_2, x_2') \leq \mathsf{dist}_{H}(x_1, x_1') + \mathsf{dist}_{H'}(x_1', x_2') + \mathsf{dist}_{H}(x_2, x_2') \leq 2M + 2 \leq \frac{101 \ln n}{p}$. Therefore, with probability at least $1 - n^{-48}$, $\mathsf{dist}_{H'}(x_1, x_2) \leq \frac{101 \ln n}{p}$.

Assume now that Theorem 2.4.4 holds for every connected graph H and every tree T of depth at most $D - 1$, with $V(H) \subseteq V(T)$. Consider now some connected graph H, and a rooted tree T of depth D, with $V(H) \subseteq V(T)$ and $|V(T)| \leq n$. We can assume without loss of generality that every vertex of V_D lies in $V(H)$, since all other vertices of V_D can be discarded from T. We partition the edges of $E(T)$ into two subsets: set E_1 contains all edges incident to the vertices of V_D, and set E_2 contains all remaining edges. Let $R_1 \subseteq R$ be the subgraph of R containing only the edges of $E_1 \cap E(R)$ and their endpoints, and let $R_2 \subseteq R$ be obtained from R by discarding all vertices of V_D and their incident edges. Notice that the definition of good vertices only depends on the edges of R_1, and so the event B only depends on the random choices made in selecting the edges of R_1, and is independent of the random choices made in selecting the edges of R_2.

Let L be a subgraph of H', obtained by starting with $L = H$, and then adding every edge of R_1 together with their endpoints to the graph. Equivalently, $L = H \cup R_1$.

Finally, we define a new graph $\hat{H}$, whose vertex set consists of two subsets: set $U_1 = V_{\leq D-1} \cap V(H)$, and set U_2, containing all vertices $v \in V_{D-1}$, such that v is connected with an edge of R_1 to some vertex of $V_D \cap V(H) = V_D$. We set $V(\hat{H}) = U_1 \cup U_2$. Observe that $V(\hat{H}) \subseteq V(L)$. In order to define the edge set $E(\hat{H})$, we add an edge between a pair of nodes w, w' in $\hat{H}$ iff the distance between w and w' in L is at most $M + 4$. We also let $\hat{T}$ be the tree obtained from T, by discarding all vertices of V_D from it. Observe that $V(\hat{H}) \subseteq V(\hat{T}) = V_{\leq D-1}$. As before, the idea is to use the induction hypothesis on the graph $\hat{H}$, together with the tree $\hat{T}$. In order to do so, we need to prove that $\hat{H}$ is a connected graph, which we do next.

Observation 2.4.9. *If the event B does not happen, then graph $\hat{H}$ is connected.*

Proof. Assume that the event B does not happen, and assume for contradiction that graph $\hat{H}$ is not connected. Let $\mathcal{C} = \{C_1, \ldots, C_r\}$ be the set of all connected components of graph $\hat{H}$.

For every vertex $v \in V(\hat{H})$, we define a set $\Gamma(v) \subseteq V(H)$ of vertices, as follows. If $v \in V(H)$, then $\Gamma(v)$ contains a single vertex – the vertex v. Otherwise, $v \in V_{D-1} \setminus V(H)$ must hold, and it must be connected by at least one edge of R_1 to some vertex in $V_D \cap V(H) = V_D$. We then let $\Gamma(v)$ contain every vertex of V_D that is connected to v by an edge of R_1.

For an ordered pair (u, v) of vertices of $V(\hat{H})$, we define a set $\mathcal{P}(u, v)$ of paths as follows: $\mathcal{P}(u, v) = \left\{P_{x,y} \mid x \in \Gamma(u), y \in \Gamma(v)\right\}$ (recall that $P_{x,y}$ is the shortest path that starts at x and ends at y in H, with the lexicographically smallest sequence $\sigma(P_{x,y})$.) Observe that every path $P_{x,y} \in \mathcal{P}(u, v)$ can be augmented to a path connecting u to v in graph L, by appending the edge (u, x) to the beginning of the path (if $u \neq x$), and appending the edge (y, v) to the end of the path (if $y \neq v$).

For every ordered pair (C_i, C_j) of distinct components of $\mathcal{C}$, consider the set $\mathcal{P}_{i,j} = \bigcup_{u \in C_i, v \in C_j} \mathcal{P}(u, v)$ of paths. We let $P_{i,j}$ be a shortest path in $\mathcal{P}_{i,j}$. We choose two distinct components $C_i, C_j \in \mathcal{C}$ with $P_{i,j}$ having the shortest length, breaking ties arbitrarily. Assume that $P_{i,j} \in \mathcal{P}(u, v)$, for $u \in C_i$ and $v \in C_j$. Let $x \in \Gamma(u)$ and $y \in \Gamma(v)$ be the endpoints of $P_{i,j}$, so $P_{i,j} = P_{x,y}$. Let P' be the augmented path obtained from $P_{x,y}$, by appending the edge (u, x) to the beginning of the path (if $u \neq x$), and appending the edge (y, v) to the end of the path (if $y \neq v$), so P' now connects u to v. Recall that $L = H \cup R_1$, and so the path P' is contained in graph L. Since we did not add edge (u, v) to $\hat{H}$, the length of P' is greater than $M+4$. Therefore, the length of the path $P_{x,y}$ in graph H is at least $M+2$. Since we have assumed that event B does not happen, there is at least one good inner vertex on path $P_{x,y}$. Let X be the set of all good vertices that serve as inner vertices of $P_{x,y}$.

We first show that for each $z \in X$, $z \notin V(\hat{H})$ must hold. Indeed, assume otherwise, that is, $z \in V(\hat{H})$ for some $z \in X$. Then z must belong to some connected component $C_\ell \in \mathcal{C}$. Since z is a good vertex, $z \in V(H)$, and so $\Gamma(z) = \{z\}$. Therefore, the sub-path of $P_{x,y}$ from x to z lies in $\mathcal{P}(u,z)$, and the sub-path of $P_{x,y}$ from z to y lies in $\mathcal{P}(z,v)$. We denote the former path by P_1 and the latter path by P_2. The length of each of these paths is less than the length of $P_{x,y}$.

Assume first that $\ell = i$, that is, $z \in V(C_i)$. Then $P_2 \in \mathcal{P}_{i,j}$, and its length is less than the length of $P_{x,y}$, a contradiction. Otherwise, $\ell \neq i$. But then $P_1 \in \mathcal{P}_{i,\ell}$, and its length is less than the length of $P_{i,j}$, a contradiction. We conclude that for each $z \in X$, $z \notin V(\hat{H})$.

Since $V(\hat{H})$ contains all vertices of $V_{\leq D-1} \cap V(H)$, and every vertex in X is a good vertex, it must be the case that $X \subseteq V_D$. Consider again some vertex $z \in X$. Since z is a good vertex and $z \in V_D$, there must be an edge $e_z = (z, z') \in E(R_1)$, connecting z to some vertex $z' \in V_{\leq D-1}$. From the definition of graph $\hat{H}$, $z' \in V(\hat{H})$, and in particular, z' must belong to some connected component of $\mathcal{C}$, while the edge e_z lies in graph L. Assume that $X = \{z_1, z_2, \ldots, z_q\}$, where the vertices are indexed in the order of their appearance on $P_{i,j}$, from x to y. Consider the sequence $\sigma' = (u, z'_1, z'_2, \ldots, z'_q, v)$ of vertices. All these vertices belong to $V(\hat{H})$, and $u \in C_i$, while $v \in C_j$. For convenience, denote $x = z'_0$ and $y = z'_{q+1}$. Then there must be an index $1 \leq a \leq q$, such that z'_a and z'_{a+1} belong to distinct connected components of $\mathcal{C}$. Note that the sub-path of $P_{i,j}$ between z_a and z_{a+1} is precisely $P_{z_a,z_{a+1}}$ – the shortest path connecting z_a to z_{a+1} in H. Since no good vertices lie between z_a and z_{a+1} on this path, and since we have assumed that event B does not happen, the length of this path is at most M. Therefore, there is a path in graph L, connecting z'_a to z'_{a+1}, whose length is at most $M+2$. This path connects a pair of vertices that belong to different connected components of $\hat{H}$, contradicting the definition of $\hat{H}$. $\square$

Consider now the tree $\hat{T}$ and the graph $\hat{H}$. Recall that $\hat{T}$ is a rooted tree of depth $D-1$, $V(\hat{T}) = V(\hat{H})$, and, assuming the event B did not happen, $\hat{H}$ is a connected graph. Moreover, $R_2 \sim \mathcal{D}(\hat{T}, p)$. Therefore, assuming that event B did not happen, we can use the induction hypothesis on the graph $\hat{H}$, the tree $\hat{T}$, and the random sub-graph R_2 of $\hat{T}$. Let B' be the bad event that for some pair $x_1, x_2 \in V(\hat{H})$ of vertices, $\mathsf{dist}_{\hat{H} \cup R_2}(x_1, x_2) > (\frac{101 \ln n}{p})^{D-1}$. From the induction hypothesis, the probability that B' happens is at most $\frac{D-1}{n^{48}}$.

Lastly, we show that, if neither of the events B, B' happen, then for every pair $x_1, x_2 \in V(H)$ of vertices, $\mathsf{dist}_{H'}(x_1, x_2) \leq (\frac{101 \ln n}{p})^D$.

Observation 2.4.10. *If neither of the events B, B' happen, then for every pair $x_1, x_2 \in V(H)$ of vertices of H, $\mathsf{dist}_{H'}(x_1, x_2) \leq (\frac{101 \ln n}{p})^D$.*

Proof. Consider any pair $x_1, x_2 \in V(H)$ of vertices. Let x_1' be a good node in $V(H)$ that is closest to x_1 in H, and define x_2' similarly for x_2. From Observation 2.4.8, $\mathsf{dist}_H(x_1, x_1') \leq M$. If $x_1' \in V_{\leq D-1}$, then we define $x_1'' = x_1'$, otherwise we let x_1'' be the node of V_{D-1} that is connected to x_1' by an edge of E_1', and we define x_2'' similarly for x_2. Therefore, $x_1'', x_2'' \in V(\hat{H})$, and, assuming event B does not happen, $\mathsf{dist}_{H'}(x_1, x_1'') \leq M+1$, and $\mathsf{dist}_{H'}(x_2, x_2'') \leq M+1$. Since we have assumed that the bad event B' does not happen, $\mathsf{dist}_{\hat{H} \cup R_2}(x_1'', x_2'') \leq (\frac{101 \ln n}{p})^{D-1}$. Recall that for every edge $e = (u, v) \in \hat{H} \cup R_2$, if $e \in E(R_2)$, then $e \in E(H')$; otherwise, $e \in E(\hat{H})$, and there is a path in graph $H \cup R_1$ of length at most $M+4$ connecting u to v in H. Therefore, $\mathsf{dist}_H(x_1'', x_2'') \leq (M+4) \cdot \mathsf{dist}_{\hat{H}}(x_1'', x_2'') \leq (\frac{101 \ln n}{p})^{D-1} \cdot (M+4)$.

Altogether, $\mathsf{dist}_{H'}(x_1, x_2) \leq \mathsf{dist}_{H'}(x_1, x_1'') + \mathsf{dist}_{H'}(x_1'', x_2'') + \mathsf{dist}_{H'}(x_2, x_2'') \leq (\frac{101 \ln n}{p})^{D-1} \cdot (M+4) + (2M+2) \leq (\frac{101 \ln n}{p})^D$, since $M = (50 \ln n)/p$. $\square$

The probability that either B or B' happen is bounded by $\frac{D}{n^{48}}$. Therefore, with probability at least $1 - \frac{D}{n^{48}}$, neither of the events happens, for every pair $x_1, x_2 \in V(H)$ of vertices of

H, $\mathsf{dist}_{H'}(x_1, x_2) \le (\frac{101 \ln n}{p})^D$.

2.5 Lower Bound: Proof of Theorem 2.1.3

In this section we provide the proof of Theorem 2.1.3. We start by proving the following slightly weaker theorem; we then extend it to obtain the proof of Theorem 2.1.3.

Theorem 2.5.1. *For all positive integers k, D, η, α such that $k/(4D\alpha\eta)$ is an integer, there exists a k-edge connected graph G with $|V(G)| = O\left(\left(\frac{k}{2D\alpha\eta}\right)^D\right)$ and diameter at most $2D$, such that, for any collection $\mathcal{T} = \{T_1, \ldots, T_{k/\alpha}\}$ of k/α spanning trees of G that causes edge-congestion at most η, some tree $T_i \in \mathcal{T}$ has diameter at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$.*

Notice that the main difference from Theorem 2.1.3 is that the graph G is no longer required to be simple; the number of vertices of $V(G)$ is no longer fixed to be a prescribed value; and the diameter of G is $2D$ instead of $2D + 2$.

Proof. For a pair of integers $w > 1, D \ge 1$, we let $T_{w,D}$ be a tree of depth D, such that every vertex lying at levels $0, \ldots, D-1$ of $T_{w,D}$ has exactly w children. In other words, $T_{w,D}$ is the full w-ary tree of depth D. We denote $N_{w,D} = |V(T_{w,D})| = 1 + w + w^2 + \cdots + w^D \le w^{D+1}/(w-1)$. We assume that for every inner vertex $v \in V(T_{w,D})$, we have fixed an arbitrary ordering of the children of v, denoted by $a_1(v), \ldots, a_w(v)$.

A *traversal* of a tree T is an ordering of the vertices of T. A *post-order traversal* on a tree T, $\pi(T)$, is defined as follows. If the tree consists of a single node v, then $\pi(T) = (v)$. Otherwise, let r be the root of the tree and consider the sequence $(a_1(r), \ldots, a_w(r))$ of its children. For each $1 \le i \le w$, let T_i be the sub-tree of T rooted at the vertex $a_i(r)$. We then let $\pi(T)$ be the concatenation of $\pi(T_1), \pi(T_2), \ldots, \pi(T_w)$, with the vertex r appearing at the end of the sequence; see Figure 2.2 for an illustration. For simplicity, we assume that $V(T_{w,D}) = \left\{v_1, v_2, \ldots, v_{N_{w,D}}\right\}$, where the vertices are indexed in the order of their appearance in $\pi(T_{w,D})$, so the traversal visits these vertices in this order.

Next, we define a graph $G_{w,D}$, as follows. The vertex set of $G_{w,D}$ is the same as the vertex set of $T_{w,D}$, namely $V(G_{w,D}) = V(T_{w,D})$. The edge set of $G_{w,D}$ consists of two subsets: $E_1 = E(T_{w,D})$, and another set E_2 of edges that contains, for each $1 \le i < N_{w,D}$, k parallel copies of the edge (v_i, v_{i+1}). We then set $E(G_{w,D}) = E_1 \cup E_2$. For convenience, we call the edges of E_1 *blue edges*, and the edges of E_2 *red edges*; see Figures 2.2 and 2.3.

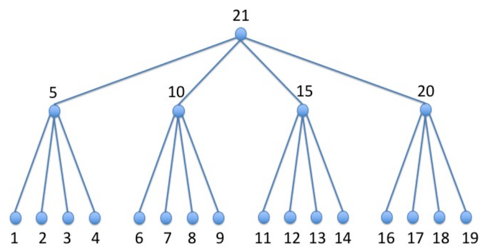


Figure 2.2: Tree $T_{4,2}$ with vertices indexed according to post-order traversal.

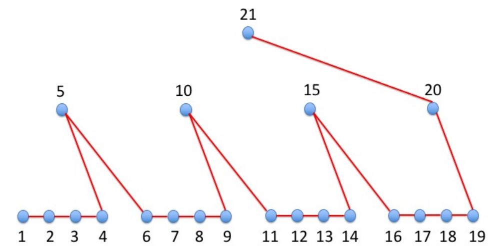


Figure 2.3: The edge set E_2 in $G_{4,2}$ (only a single copy of each edge is shown).

It is easy to verify that graph $G_{w,D}$ must be k-edge connected, since for any partition of $V(G_{w,D})$, there is some index $1 \le i < N_{w,D}$ with v_i, v_{i+1} separated by the partition, and so k parallel edges connecting v_i to v_{i+1} must cross the partition.

We now fix an integer $w = k/(2D\alpha\eta)$ (note that $w \ge 2$), and we let $T = T_{w,D}$ be the corresponding tree and $G = G_{w,D}$ the corresponding graph. For convenience, we denote $N_{w,D}$ by N. Recall that $N \le w^{D+1}/(w-1) = O\left(\left(\frac{k}{2D\alpha\eta}\right)^D\right)$. As observed before, G is k-edge connected. Since the depth of T is D, and $T \subseteq G$, it is easy to see that the diameter of G is at most $2D$.

We now consider any collection $\mathcal{T} = \{T_1, \ldots, T_{k/\alpha}\}$ of k/α spanning trees of G that causes edge-congestion at most η. Our goal is to show that some tree $T_i \in \mathcal{T}$ has diameter at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$.

For convenience, we denote $V(G) = V(T) = V$. We say that a vertex $x \in V$ is an *ancestor* of a vertex $y \in V$ if x is an ancestor of y in the tree T, that is, $x \ne y$, and x lies on the unique path connecting y to the root of T.

Let $L \subseteq V$ be the set of vertices that serve as leaves of the tree T. We denote by $u = v_1$ a vertex of L that has the lowest index, and by u' the vertex of L with the largest index. It is easy to see that $u' = v_{N-D}$, as every vertex whose index is greater than that of u' is an ancestor of u'. For each $1 \leq j \leq k/\alpha$, we denote by P_j the unique path that connects u to u' in tree T_j. Let $\mathcal{P} = \left\{P_j \mid 1 \leq j \leq k/\alpha\right\}$. It is enough to show that at least one of the paths P_j has length at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$. In order to do so, we show that $\sum_{j=1}^{k/\alpha} |E(P_j)|$ is sufficiently large. At a high level, we consider the red edges (v_i, v_{i+1}) (the edges of E_2), and show that many of the paths in $\mathcal{P}$ must contain copies of each such edge. This in turn will imply that $\sum_{P_j \in \mathcal{P}} |E(P_j)|$ is large, and that some path in $\mathcal{P}$ is long enough.

For each vertex $v_i \in L$ such that $v_i \neq u'$, we let $S_i = \{v_1, \ldots, v_i\}$, and we let $\overline{S}_i = \{v_{i+1}, \ldots, v_N\}$. Notice that, since $u \in S_i$ and $u' \in \overline{S}_i$, every path in $\mathcal{P}$ must contain an edge of $E_G(S_i, \overline{S}_i)$. Note that the only red edges in $E_G(S_i, \overline{S}_i)$ are the k parallel copies of the edge (v_i, v_{i+1}). In the next observation, we show that the number of blue edges in $E_G(S_i, \overline{S}_i)$ is bounded by Dw.

Observation 2.5.2. *For each vertex $v_i \in L$ such that $v_i \neq u'$, for every blue edge $e \in E_G(S_i, \overline{S}_i)$, at least one endpoint of e must be an ancestor of v_i.*

Proof. We consider a natural layout of the tree T, where for every inner vertex x of the tree, its children $a_1(x), \ldots, a_w(x)$ are drawn in this left-to-right order (see Figure 2.4). Consider the path Q connecting the root of T to v_i, so every vertex on Q (except for v_i) is an ancestor of v_i. All vertices lying to the left of Q in the layout are visited before v_i by $\pi(T)$. All vertices lying to the right of Q, and on Q itself (excluding v_i) are visited after v_i. It is easy to see that the vertices of Q separate the two sets in T, and so the only blue edges connecting S_i to $\overline{S}_i$ are edges incident to the vertices of $V(Q) \setminus \{v_i\}$. $\square$

Since every vertex of the tree T has at most w children, and since the depth of the tree is D, we obtain the following corollary of Observation 2.5.2.

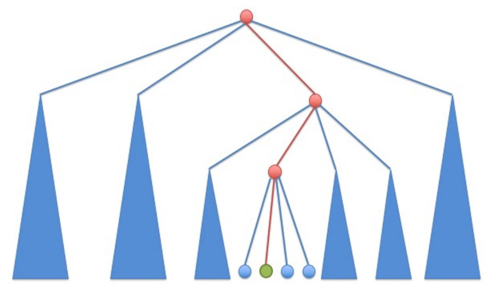


Figure 2.4: A layout of the tree T. Vertex v_i is shown in green and path Q in red. All vertices lying to the left of Q in this layout appear before v_i in $\pi(T)$, and all vertices lying to the right of Q or on Q (except for v_i) appear after v_i in $\pi(T)$.

Corollary 2.5.3. *For each vertex $v_i \in L$ such that $v_i \neq u'$, at most Dw blue edges lie in $E_G(S_i, \overline{S}_i)$.*

Since the trees in $\mathcal{T}$ cause edge-congestion η, at most $Dw\eta$ trees of $\mathcal{T}$ may contain blue edges in $E_G(S_i, \overline{S}_i)$. Each of the remaining $\frac{k}{\alpha} - Dw\eta \geq \frac{k}{2\alpha}$ trees contains a copy of the red edge (e_i, e_{i+1}) (recall that $w = k/(2D\alpha\eta)$.) Therefore, $\sum_{P_j \in \mathcal{P}} |E(P_j)| \geq |L| \cdot \frac{k}{2\alpha} \geq \frac{Nk}{4\alpha}$, since $|L| \geq |N|/2$. We conclude that at least one path $P_j \in \mathcal{P}$ must have length at least $\frac{Nk}{4\alpha}/\frac{k}{\alpha} \geq \frac{N}{4}$, and so the diameter of T_j is at least $\frac{N}{4}$. Since $N \geq w^D \geq \left(\frac{k}{2D\alpha\eta}\right)^D$, the diameter of T_j is at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$. $\square$

We are now ready to complete the proof of Theorem 2.1.3. First, we show that we can turn the graph G into a simple graph, and ensure that $|V(G)| = n$, if $n \geq 3k \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$. Let $G'_{w,D}$ be the graph obtained from $G_{w,D}$ as follows. For each $1 \leq i \leq N$, we replace the vertex v_i with a set $X_i = \{x_i^1, x_i^2 \ldots, x_i^k\}$ of k vertices that form a clique. For each $1 \leq i < N$, the k red edges connecting v_i to v_{i+1} are replaced by the perfect matching $\{(x_i^t, x_{i+1}^t)\}_{1 \leq t \leq k}$ between vertices of X_i and vertices of X_{i+1}. Each blue edge (v_i, v_j) is replaced by a new edge (x_i^1, x_j^1). Since $n \geq 3k \cdot \left(\frac{k}{2D\alpha\eta}\right)^D > k|V(G)| + k$, we add $n - k|V(G)| > k$ new vertices that form a clique, and for each newly-added vertex, we add an edge connecting it to x_N^1 (recall that the vertex v_N is the root of T). We denote $G' = G'_{w,D}$ for simplicity. It is not hard to see that G' has n vertices and it is k-edge connected. Moreover, G' has diameter at most $2D+2$, since its subgraph induced by vertices of $\{x_i^1\}_{1\le i\le N}$ has diameter $2D$, and every other vertex of G' is a neighbor of one of the vertices in $\{x_i^1\}_{1\le i\le N}$. The tree T' is defined exactly as before, except that every original vertex v_j is now replaced with its copy x_j^1. Let L denote the set of all leaf vertices in T'.

Assume that we are given a collection $\mathcal{T}=\{T_1,\ldots,T_{k/\alpha}\}$ of k/α spanning trees of G' that causes edge-congestion at most η. For each $1\le i\le k/\alpha$, we denote by Q_i the unique path that connects x_1^1 to x_{N-D}^1 in T_i and denote $\mathcal{Q}=\{Q_i \mid 1\le i\le k/\alpha\}$. For each every leaf vertex $x_j^1\in L$, we define a cut $(W_j,\overline{W}_j)$ as follows: $W_j=\bigcup_{1\le s\le j}X_s$ and $\overline{W_j}=V(G')\setminus W_j$. Using reasoning similar to that in Corollary 2.5.3, it is easy to see that for every leaf vertex $x_j^1\in L$, the set $E_{G'}(W_j,\overline{W_j})$ of edges contains at most Dw blue edges – the edges of the tree T'. Since the trees in $\mathcal{T}$ cause edge-congestion at most η, at most $Dw\eta$ trees of $\mathcal{T}$ may contain blue edges in $E_{G'}(W_j,\overline{W_j})$. Therefore, for each of the remaining $\frac{k}{\alpha}-Dw\eta\ge\frac{k}{2\alpha}$ trees T_i, path Q_i must contain a red edge from $\{(x_j^t,x_{j+1}^t)\}_{1\le t\le k}$. Therefore, the sum of lengths of all paths of $\mathcal{Q}$ is at least $\frac{Nk}{4\alpha}$, and so at least one path $Q_i\in\mathcal{Q}$ must have length at least $\frac{N}{4}$. We conclude that some tree $T_i\in\mathcal{T}$ has diameter at least $\frac{1}{4}\cdot\left(\frac{k}{2D\alpha\eta}\right)^D$.

Lastly, we extend our results to edge-independent trees. We use the same simple graph G' and the same tree T' as before, setting the congestion parameter $\eta=2$. Assume that we are given a collection $\mathcal{T}'=\{T'_1,\ldots,T'_{k/\alpha}\}$ of k/α edge-independent spanning trees of G' and let $x\in V(G')$ be their common root vertex. For each $1\le i\le k/\alpha$, we denote by Q'_i the unique path that connects vertex x_1^1 to vertex x_{N-D}^1 in tree T'_i, and we denote $\mathcal{Q}'=\{Q'_i \mid 1\le i\le k/\alpha\}$. Note that, for each $1\le i\le k/\alpha$, the path Q'_i is a sub-path of the path obtained by concatenating the path Q''_i, connecting x_1^1 to x in T'_i, with the path Q'''_i, connecting x_{N-D}^1 to x in T'_i. Since the trees in $\mathcal{T}'$ are edge-independent, the paths in $\{Q''_i\}_{1\le i\le k/\alpha}$ are edge-disjoint and so are the paths in $\{Q'''_i\}_{1\le i\le k/\alpha}$. Therefore, the paths of $\mathcal{Q}'$ cause edge-congestion at most 2. The remainder of the proof is the same as before and is omitted here.

2.6 Tree Packing for (k, D)-Connected Graphs: Proof of Theorem 2.1.4

In this section we provide the proof of Theorem 2.1.4. Recall that we are given a (k, D)-connected n-vertex graph G. Our goal is to design an efficient randomized algorithm that computes a collection $\mathcal{T} = \{T_1, \ldots, T_k\}$ of k spanning trees of G, such that, for each $1 \leq \ell \leq k$, the tree T_ℓ has diameter at most $O(D \log n)$, and with high probability each edge of G appears in $O(\log n)$ trees of $\mathcal{T}$. Note that we allow the graph G to have parallel edges. However, we can assume w.l.o.g. that for every pair (u, v) of vertices of G, there are at most k parallel edges (u, v); all remaining edges can be deleted without violating the (k, D)-connectivity property of G.

The main tool that we use in our proof is the following theorem and its corollary.

Theorem 2.6.1. *There is an efficient algorithm, that, given a (k, D)-connected graph G, a subset $U \subsetneq V(G)$ of its vertices, and an additional vertex $s \in V(G) \setminus U$, computes a flow f in G with the following properties:*

- *the endpoints of every flow-path lie in $U \cup \{s\}$;*
- *for each vertex $u \in U$, the total flow on all paths that originate or terminate at u is at least k;*
- *the total amount of flow through any edge is at most 2; and*
- *each flow-path has length at most $2D$.*

Notice that a flow-path is allowed to contain vertices of $U \cup \{s\}$ as inner vertices. We defer the proof of Theorem 2.6.1 to Section 2.6.1, after we complete the proof of Theorem 2.1.4 using it. We obtain the following useful corollary of the theorem.

Corollary 2.6.2. *There is an efficient algorithm, that, given a (k, D)-connected graph G and a subset $S \subseteq V(G)$ of its vertices, computes a bi-partition (S', S'') of S, and a flow f from vertices of S'' to vertices of S', such that the following hold:*

- *every vertex of S'' sends at least $k/2$ flow units;*
- *every flow-path has length at most $2D$;*
- *the total amount of flow through any edge is at most 3; and*
- $|S'| \leq \frac{|S|}{2} + 1$.

Proof. Let $s \in S$ be an arbitrary vertex, and set $U = S \setminus \{s\}$. We apply Theorem 2.6.1 to graph G, vertex set U and the vertex s, to obtain a flow f. Recall that every vertex of U sends or receives at least k flow units, and all flow-paths have length at most $2D$. Let $\mathcal{P}'$ be the set of all paths in G on which a non-zero amount of flow is sent. Since the algorithm in Theorem 2.6.1 is efficient, we are guaranteed that $|\mathcal{P}'| \leq n^c$ for some constant c, where $n = |V(G)|$. It will be convenient for us to ensure that for every path $P \in \mathcal{P}'$, $f(P)$ is an integral multiple of $1/n^c$. In order to achieve this, for every flow-path $P \in \mathcal{P}'$, we round $f(P)$ up to the next integral multiple of $1/n^c$. Note that this increases the total amount of flow by at most 1, so the total amount of flow through any edge is at most 3.

We now compute a bi-partition (S', S'') of S, as follows. We start from an arbitrary partition (S', S''). Consider any vertex $v \in S$. For convenience, we direct all flow-paths of $\mathcal{P}'$ for which v serves as an endpoint away from v. Let $q'(v)$ be the total amount of flow that originates at v and terminates at vertices of S', and define $q''(v)$ similarly for the total amount of flow between v and S''.

If $v \in S'$, but $q'(v) > q''(v)$, then we move v from S' to S''. Similarly, if $v \in S''$, but $q''(v) > q'(v)$, then we move v from S'' to S'. Notice that in either case, the total amount of flow between vertices of S' and vertices of S'' increases by at least $1/n^c$. We continue performing these modifications, until for every vertex $v \in S'$, $q'(v) \leq q''(v)$, and for every vertex $v \in S''$, $q''(v) \leq q'(v)$. Since the total amount of flow between S' and S'' grows by at least $1/n^c$ in every iteration, the number of such iterations is bounded by $O(|E(G)| \cdot n^c) = O(\text{poly}(n))$.

Consider the partition (S', S'') of S obtained at the end of this algorithm. Assume w.l.o.g. that $|S'| \leq |S''|$; otherwise we switch S' and S''. If the vertex s lies in S'', then we move it to S'. Notice that we are now guaranteed that for every vertex $u \in S''$, $q'(u) \geq q''(u)$, and so at least $k/2$ flow units are sent between u and the vertices of S'. In order to obtain the final flow f', we discard from f all flow-paths except those connecting the vertices of S'' to the vertices of S', and we direct these flow paths towards the vertices of S'. It is easy to verify that $|S'| \leq |S|/2 + 1$. $\square$

Our algorithm consists of two phases. In the first phase, we define a partition of the vertices of G into layers $L_1, \ldots, L_h$, where $h = O(\log n)$. Additionally, for each $1 \leq i \leq h$, we define a flow f_i in graph G between vertices of L_i and vertices of $L_1 \cup \cdots \cup L_{i-1}$. In the second phase, we use the layers and the flows in order to construct the desired set of spanning trees.

Phase 1: partitioning into layers. We use a parameter $h = \Theta(\log n)$, whose exact value will be set later. We now define the layers $L_h, \ldots, L_1$ in this order, and the corresponding flows $f_h, \ldots, f_1$. In order to define the layer L_h, we let $S = V(G)$, and we apply Corollary 2.6.2 to the graph G and the set S of its vertices, to obtain a partition (S', S'') of S, with $|S'| \leq |S|/2 + 1$, and the flow f between the vertices of S'' and the vertices of S', where every vertex of S'' sends at least $k/2$ units of flow, each flow-path has length at most $2D$, and the edge-congestion caused by f is at most 3. We then set $L_h = S''$ and $f_h = f$, and continue to the next iteration.

Assume now that we have constructed $L_h, \ldots, L_i$, we now show how to construct L_{i-1}. Let $S = V(G) \setminus (L_h \cup \cdots \cup L_i)$. We apply Corollary 2.6.2 to the graph G and the set S of its vertices, to obtain a partition (S', S'') of S, with $|S'| \leq |S''|/2 + 1$, and the corresponding flow f. We then set $L_{i-1} = S''$, $f_{i-1} = f$, and continue to the next iteration. If we reach an iteration where $|S| \leq 2$, we arbitrarily designate one of the two vertices as s, and we let U be a set of vertices containing the other vertex. We then use Theorem 2.6.1 in order to find a flow of value at least k between the two vertices, such that the edge-congestion of the flow is at most 2, and every flow-path has length at most $2D$. We then add the vertex that lies in U to the current layer, and the vertex s to the final layer L_1. If we reach an iteration where $|S| = 1$, then we add the vertex of S to the final layer L_1 and terminate the algorithm. The number h of layers is chosen to be exactly the number of iterations in this algorithm. Notice that $h \leq 2\log n$ must hold. Observe also that, for all $1 < i \leq h$, flow f_i originates at vertices of L_i, terminates at vertices of $L_1 \cup \cdots \cup L_{i-1}$, uses flow-paths of length at most $2D$, and causes edge-congestion at most 3.

Phase 2: constructing the trees. In order to construct the spanning trees $T_1, \ldots, T_k$, we start with letting each tree contain all vertices of G and no edges. We then process every vertex $v \in V(G)$ one-by-one. Assume that $v \in L_i$, for some $1 \leq i \leq h$. Consider the following experiment. Let $\mathcal{Q}(v)$ be the set of all flow-paths that carry non-zero flow in f_i, and connect v to vertices of $L_1 \cup \cdots \cup L_{i-1}$. Let $F(v)$ be the total amount of flow f_i on all paths $P \in \mathcal{Q}(v)$; recall that $F(v) \geq k/2$ must hold. We choose a path $P \in \mathcal{Q}(v)$ at random, where the probability to choose a path P is precisely $f_i(P)/F(v)$. We repeat this experiment k times, obtaining paths $P_1(v), \ldots, P_k(v)$. For each $1 \leq j \leq k$, we add all edges of $P_j(v)$ to T_j. Consider the graphs $T_1, \ldots, T_k$ at the end of this process. Notice that each such graph T_j may not be a tree. We show first that the diameter of each such graph is bounded by $O(D \log n)$.

Claim 2.6.3. *For all* $1 \leq j \leq k$*,* $\mathrm{diam}(T_j) \leq O(D \log n)$*.*

Proof. Fix an index $1 \leq j \leq k$. Let r be the unique vertex lying in L_1. We prove that for all $1 \leq i \leq h$, for every vertex $v \in L_i$, there is a path connecting v to r in T_j, of length at most $2D(i-1)$ by induction on i.

The base of the induction is when $i = 1$ and the claim is trivially true. Assume now that the claim holds for layers $L_1, \ldots, L_{i-1}$. Let v be any vertex in layer L_i. Consider the path $P_j(v)$ that we have selected. Recall that this path has length at most $2D$, and it connect v to some vertex $u \in L_1 \cup \cdots \cup L_{i-1}$. By the induction hypothesis, there is a path P in T_j of length at most $2D(i-2)$, that connects u to r. Since all edges of $P_j(v)$ are added to T_j, the path $P_j(v)$ is contained in T_j. By concatenating path $P_j(v)$ with path P, we obtain a path connecting v to r, of length at most $2D(i-1)$. $\square$

Lastly, we prove that with high probability, every edge of G belongs to $O(\log n)$ graphs $T_1, \ldots, T_k$.

Claim 2.6.4. *With probability at least* $(1 - 1/\operatorname{poly}(n))$*, every edge of* G *lies in at most* $O(\log n)$ *graphs* $T_1, \ldots, T_k$.

The proof follows the standard analysis of the Randomized Rounding technique and is delayed to Section 2.6.2.

For each $1 \leq j \leq k$, we can now let T'_j be a BFS tree of the graph T_j, rooted at the vertex r. From Claim 2.6.3, each tree T'_j has diameter at most $O(D \log n)$, and from Claim 2.6.4, the resulting set of trees cause edge-congestion $O(\log n)$.

2.6.1 Proof of Theorem 2.6.1

For every vertex $u \in U$, let $\mathcal{P}(u)$ be the set of all paths in graph G of length at most $2D$, that connect u to vertices of $(U \cup \{s\}) \setminus \{u\}$. Notice that for a pair $u, u' \in U$ of distinct vertices, each path connecting u to u' belongs to both $\mathcal{P}(u)$ and $\mathcal{P}(u')$. Let $\mathcal{P}^* = \bigcup_{u \in U} \mathcal{P}(u)$. We use the following linear program, that has no objective function; our goal will be to find a feasible solution satisfying all constraints.

$$
\begin{array}{lll}
\text{(LP-1)} & \sum_{P\in\mathcal{P}(u)} f(P)\geq k & \forall u\in U\\
 & \sum_{\substack{P\in\mathcal{P}^*:\\ e\in P}} f(P)\leq 2 & \forall e\in E(G)\\
 & f(P)\geq 0 & \forall P\in\mathcal{P}^*
\end{array}
$$

Note that, if f is a feasible solution to (LP-1), then it satisfies all requirements of Theorem 2.6.1. The following claim provides an efficient algorithm for solving (LP-1); its proof uses standard techniques and is deferred to Section 2.6.3.

Claim 2.6.5. *There is an efficient algorithm that computes a feasible solution to (LP-1), if such a solution exists.*

It now remains to prove that there is a feasible solution to (LP-1). We do so using the following lemma, that proves a stronger claim, namely that there is an integral solution to (LP-1).

Lemma 2.6.6. *Let G be a (k,D)-connected graph, let $U\subsetneq V(G)$ be any subset of its vertices, and let $s\notin U$ be any additional vertex. Then there exists a set $\mathcal{P}$ of paths in G, such that:*

- *each path $P\in\mathcal{P}$ connects a pair of distinct vertices in $U\cup\{s\}$;*
- *each node in U is the endpoint of at least k paths in $\mathcal{P}$ (but s may serve as an endpoint on fewer paths);*
- *each path $P\in\mathcal{P}$ has length at most $2D$; and*
- *each edge of G appears on at most two paths in $\mathcal{P}$.*

Notice that the lemma immediately implies that there is a feasible solution to (LP-1), as we can simply send one unit of flow on each path of $\mathcal{P}$. We now turn to prove Lemma 2.6.6.

Proof of Lemma 2.6.6. The proof relies on a theorem from [10], that needs the following definitions.

Definition 17. (Canonical Spider) *Let $\mathcal{M}$ be any collection of simple paths, such that each path $P \in \mathcal{M}$ has a distinguished endpoint $t(P)$, and the other endpoint is denoted by $v(P)$. We say that the paths in $\mathcal{M}$ form a* canonical spider *iff $|\mathcal{M}| > 1$ and there is a vertex v, such that for every path $P \in \mathcal{M}$, $v(P) = v$. Moreover, the only vertex that appears on more than one path of $\mathcal{M}$ is v (see Figure 2.5). We refer to v as the* head *of the spider, and the paths of $\mathcal{M}$ are called the* legs *of the spider.*

Definition 18. (Canonical Cycle) *Let $\mathcal{M} = \{Q_1, \ldots, Q_h\}$ be any collection of simple paths, where each path Q_i has a distinguished endpoint $t(Q_i)$ that does not appear on any other path of $\mathcal{M}$, and the other endpoint is denoted by $v(Q_i)$. We say that paths of $\mathcal{M}$ form a* canonical cycle*, iff:*

- *h is an odd integer;*
- *for each $1 \leq i \leq h$, there is a vertex $v'(Q_i) \neq v(Q_i)$ on path Q_i, such that $v'(Q_i) = v(Q_{i-1})$ (here we use the convention that $Q_0 = Q_h$); and*
- *for each $1 \leq i \leq h$, no vertex of Q_i appears on any other path of $\mathcal{M}$, except for $v'(Q_i)$ that belongs to Q_{i-1} only and $v(Q_i)$ that belongs to Q_{i+1} only (see Figure 2.5).*

Note that the definition of a canonical cycle here is slightly stronger than definition of a canonical cycle in [10], since we additionally require that, for each $1 \leq i \leq h$, the vertex $v'(Q_i) \neq v(Q_i)$.

We use the following result of Chuzhoy and Khanna (Theorem 4 in [10]). We note that the theorem appearing in [10] is slightly weaker since they used a weaker definition of a canonical cycle, but their proof immediately implies the stronger result that we state below.

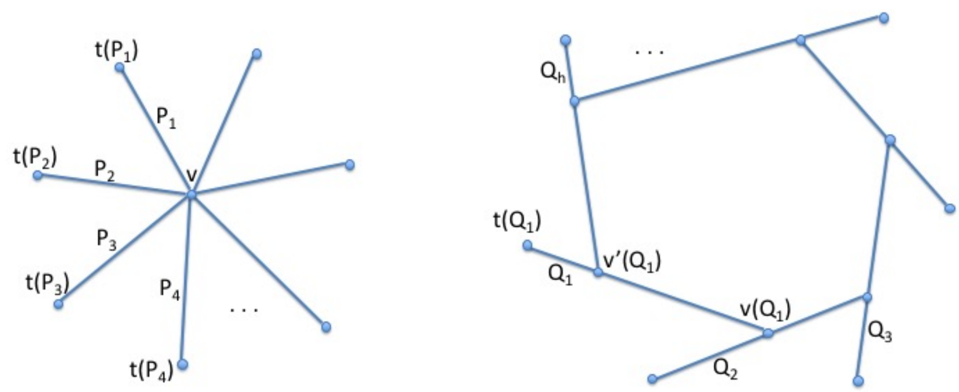


Figure 2.5: A canonical spider (left) and a canonical cycle (right).

Theorem 2.6.7. *There is an efficient algorithm, that, given any collection $\mathcal{Q}$ of paths, where every path $P \in \mathcal{Q}$ has a distinguished endpoint $t(P)$ that does not appear on any other path of $\mathcal{Q}$, computes, for each path $P \in \mathcal{Q}$, a prefix (i.e. a sub-path of P that contains $t(P)$) $q(P)$, such that, in the graph induced by $\{q(P) \mid P \in \mathcal{Q}\}$, the prefixes appearing in each connected component either form a canonical spider, a canonical cycle, or the connected component contains exactly one prefix $q(P)$, where $q(P) = P$ for some $P \in \mathcal{Q}$.*

Recall that we are given a (k, D)-connected graph G, together with a subset $U \subsetneq V(G)$ of its vertices, that we call *terminals*, and an additional vertex $s \notin U$. From the definition of (k, D)-connectivity, we are guaranteed that every vertex $u \in U$, there is a set $\mathcal{R}(u)$ of k edge-disjoint simple paths in G, of length at most D each, connecting u to s. Let $\mathcal{R} = \bigcup_{u \in U} \mathcal{R}(u)$. Intuitively, we would like to apply Theorem 2.6.7 to the set $\mathcal{R}$ of paths, where for each vertex $u \in U$, and for each path $R \in \mathcal{R}(u)$, the distinguished endpoint $t(R)$ is u. The idea is then to use the resulting canonical cycle and canonical spider structures in order to connect the vertices of U to each other and to s via short paths that are disjoint in their edges, thus constructing the collection $\mathcal{P}$ of paths. For example, if a set $\mathcal{M}$ of prefixes of the paths in $\mathcal{R}$ form a canonical spider, we can partition the legs of the spider into pairs, and each pair then defines a path connecting two vertices of U to each other, which is then added to $\mathcal{P}$. There are two problems with this approach. The first problem is that Theorem 2.6.7 requires that the distinguished endpoints $t(P)$ of the paths $P \in \mathcal{R}$ are distinct from each other, and moreover that $t(P)$ does not lie on any other path of $\mathcal{R}$. This difficulty is easy to overcome by making k copies of every terminal $u \in U$ and then modifying the paths in $\mathcal{R}(u)$ so that each of them starts from a different copy. The second difficulty is that it is possible that some resulting set $\mathcal{M}$ of prefixes that forms a canonical spider consists entirely of paths that belong to a single set $\mathcal{R}(u)$, and so the spider cannot be used to connect distinct vertices of U to each other. The reason that this may happen is that the paths in $\mathcal{R}(u)$ are only guaranteed to be edge-disjoint, and so they may share vertices. If, in contrast, they were internally vertex-disjoint, then such a problem would not arise. In order to overcome these difficulties, we slightly modify the graph G, first by replacing it with its line graph, so that any set of edge-disjoint paths in G corresponds to a set of internally node-disjoint paths in the line graph, and then creating k copies of each terminal $u \in U$. We now describe the construction of the modified graph H, in two steps.

In the first step, we construct the line graph L of G, as follows: the vertex set $V(L)$ contains a vertex v_e for each edge $e \in E(G)$. Given a pair $v_e, v_{e'}$ of vertices of L, we connect them with an edge iff e and e' share an endpoint in G.

Let H be the graph obtained from graph L by adding, for each terminal $u \in U$, a collection $\{u_1, \ldots, u_k\}$ of k vertices, that we call the *copies of* u. For each such new vertex u_i, and for every edge e that is incident to u in G, we add the edge (u_i, v_e) to the graph. Additionally, we add the vertex s to the graph, and connect it to every vertex v_e where e is an edge incident to s in G.

Recall that we have defined, for every vertex $u \in U$, a collection $\mathcal{R}(u)$ of k edge-disjoint simple paths in G of length at most D each, connecting u to s. Denote $\mathcal{R}(u) = \{R_1(u), \ldots, R_k(u)\}$. We transform the set $\mathcal{R}(u)$ of paths into a set $\mathcal{R}'(u)$ of k paths in graph H, that are internally vertex-disjoint, and each path connects a distinct copy of u to s. In order to do so, fix some $1 \le i \le k$, and consider the path $R_i(u)$. Let $e_1^i, e_2^i, \ldots, e_r^i$ be the sequence of edges on the path $R_i(u)$, with e_1^i incident to u and e_r^i incident to s. Consider the following sequence of vertices in graph H: $(u_i, v_{e_1^i}, v_{e_2^i}, \ldots, v_{e_r^i}, s)$. It is easy to verify that this vertex sequence defines a path in graph H, that we denote by $R'_i(u)$. Let $\mathcal{R}'(u) = \left\{R'_i(u) \mid 1 \le i \le k\right\}$ be the resulting set of paths. Since the paths in $\mathcal{R}(u)$ are edge-disjoint, it is immediate to verify that the paths in $\mathcal{R}'(u)$ are internally node-disjoint; in fact the only vertex that these paths share is the vertex s. The number of inner vertices on each such path is at most D. For each path $R'_i(u)$, we let its distinguished endpoint $t(R'_i(u))$ be the vertex u_i. Lastly, we let $\mathcal{Q} = \bigcup_{u \in U} \mathcal{R}'(u)$. Observe that for every path $R \in \mathcal{Q}$, the distinguished endpoint $t(R)$ does not lie on any other paths of $\mathcal{Q}$.

We apply Theorem 2.6.7 to the resulting set $\mathcal{Q}$ of paths and obtain, for each path $P \in \mathcal{Q}$, a prefix $q(P)$. Let $\hat{H}$ be the subgraph of H that is induced by all edges and vertices that appear on the paths in $\{q(P) \mid P \in \mathcal{Q}\}$. Let $\mathcal{C}$ be the set of all connected components of $\hat{H}$. For every component $C \in \mathcal{C}$, we denote by $\mathcal{Q}(C) \subseteq \mathcal{Q}$ the set of paths whose prefixes are contained in C, and we denote by $\tilde{\mathcal{Q}}(C) = \{q(P) \mid P \in \mathcal{Q}(C)\}$ the corresponding set of prefixes, so $C = \bigcup_{P' \in \tilde{\mathcal{Q}}(C)} P'$.

Next, for every component $C \in \mathcal{C}$, we define a collection $\mathcal{P}(C)$ of paths in the original graph G, with the following properties:

- P1. an edge of G may lie on at most two paths in $\mathcal{P}(C)$;
- P2. the paths in $\mathcal{P}(C)$ only contain edges $e \in E(G)$ with $v_e \in V(C)$;
- P3. for every terminal $u \in U$, the number of paths of $\mathcal{P}(C)$ for which u serves as an endpoint is at least as large as the number of paths of $\mathcal{R}'(u)$ that lie in $\mathcal{Q}(C)$; and
- P4. every path in $\mathcal{P}(C)$ has length at most $2D$;

Assume first that we have computed, for every component $C \in \mathcal{C}$, a set $\mathcal{P}(C)$ of paths in graph G with the above properties. We then set $\mathcal{P} = \bigcup_{C \in \mathcal{C}} \mathcal{P}(C)$. It is easy to verify that set $\mathcal{P}$ has all required properties. Indeed, since the components of $\mathcal{C}$ are disjoint in their vertices, Properties P1 and P2 ensure that every edge of G belongs to at most two paths of $\mathcal{P}$. Since, for every terminal $u \in U$, $|\mathcal{R}'(u)| = k$, Property P3 ensures that u serves as an endpoint of at least k paths in $\mathcal{P}$. Lastly, Property P4 ensures that the length of every path in $\mathcal{P}$ is at most $2D$.

From now on we fix a component $C \in \mathcal{C}$. It is now sufficient to show an efficient algorithm for constructing the set $\mathcal{P}(C)$ of paths in graph G with Properties P1—P4. Recall that Theorem 2.6.7 guarantees that the prefixes in $\tilde{\mathcal{Q}}(C)$ either form a canonical spider, or they form a canonical cycle, or $\tilde{\mathcal{Q}}(C)$ consists of a single path $q(P) = P$ for some path $P \in \mathcal{Q}$. We consider each of these different cases in turn; for the case of canonical spider we need to consider two sub-cases, depending on whether the head of the spider is s or not.

Case 1: This case happens if $\tilde{\mathcal{Q}}(C)$ contains a single path, or if the paths of $\tilde{\mathcal{Q}}(C)$ form a canonical spider, whose head is s. In either case, from the construction of the paths in $\mathcal{Q}$, it is easy to verify that for every path $P \in \mathcal{Q}(C)$, the prefix $q(P)$ is the path P itself. For each path $P \in \tilde{\mathcal{Q}}(C)$, we define a path P' in graph G, as follows. Assume that $P = (u_i, v_{e_1}, v_{e_2}, \ldots, v_{e_r}, s)$. We then let P' be a path in graph G, that starts at the terminal u, traverses the edges $e_1, \ldots, e_r$ in this order, and terminates at s. Let $\mathcal{P}(C) = \left\{P' \mid P \in \tilde{\mathcal{Q}}(C)\right\}$. Since the paths in $\tilde{\mathcal{Q}}(C)$ are vertex-disjoint except for sharing the vertex s, the paths in $\mathcal{P}(C)$ are all edge-disjoint. It is easy to verify that Properties P1—P4 hold for $\mathcal{P}(C)$.

Case 2: This case happens if the paths in $\tilde{\mathcal{Q}}(C)$ form a canonical spider, whose head is not s. Note that, from the definition of the paths in $\mathcal{Q}$, the head of the spider must be some vertex v_{e^*} with $e^* \in E(G)$. We denote $e^* = (x, y)$. Note that every path $P \in \mathcal{Q}(C)$ contains the vertex v_{e^*}. Therefore, each such path must belong to a different set $\mathcal{R}'(u)$, and no two paths in $\mathcal{Q}(C)$ may originate from two copies of the same terminal. For every path $P \in \tilde{\mathcal{Q}}(C)$, we define a new path P' in graph G, as follows. Assume that the sequence of vertices on P is $(u_i, v_{e_1}, v_{e_2}, \ldots, v_{e_r}, v_{e^*})$, then we let path P' start at the terminal u, and then traverse the edges $e_1, e_2, \ldots, e_r$ in this order. Note that path P' has to terminate at a vertex that serves as an endpoint of e^*. We define two sets of paths: set S_x contains all paths P' for $P \in \tilde{\mathcal{Q}}(C)$ that terminate at x, and set S_y is defined similarly for y. Therefore, $|S_x| + |S_y| = |\tilde{\mathcal{Q}}(C)|$. From the above discussion, every path in $S_x \cup S_y$ originates at a distinct terminal.

Assume first that $|S_x| > 1$ and $|S_y| > 1$. Consider the set S_x of paths. We construct a set Π_x of pairs of paths from S_x as follows. If $|S_x|$ is even, then we simply partition all paths in S_x into $|S_x|/2$ disjoint pairs. Otherwise, if $|S_x|$ is odd, then we construct $(|S_x| + 1)/2$ pairs, such that every path of S_x belongs to exactly one pair in Π_x, except for one arbitrary path that belongs to two pairs. Consider now any pair (P'_1, P'_2) of paths in Π_x. As observed before, the two paths must originate at distinct terminals. We construct a new path by concatenating P'_1 with P'_2, and add this path to $\mathcal{P}(C)$. We process the paths of S_y similarly. Notice that every prefix in $\tilde{\mathcal{Q}}(C)$ is now a sub-path of either one or two paths in $\mathcal{P}(C)$. Since the paths in $\tilde{\mathcal{Q}}(C)$ are internally vertex disjoint, and since the edge e^* is not included in any of the paths in $S_x \cup S_y$, every edge of G may belong to at most two paths of $\mathcal{P}(C)$. It is immediate to verify that Properties P1—P4 hold in $\mathcal{P}(C)$.

Assume now that $|S_x| = 1$ or $|S_y| = 1$ (or both). We assume w.l.o.g. that $|S_y| = 1$. We construct the set Π_x of pairs of paths in S_x exactly as before (if $|S_x| = 1$ then $\Pi_x = \emptyset$). For every pair (P'_1, P'_2) of paths in Π_x, we construct a new path that is added to $\mathcal{P}(C)$ exactly as before. Additionally, we choose an arbitrary path $P'_i \in S_x$ that participates in at most one pair in Π_x (notice that such a path has to exist). Let P' be the unique path in S_y. As observed before, the two paths must originate from distinct terminals. We construct a new path in graph G, by concatenating the path P'_i, the edge e^*, and the path P'. We add the resulting path to $\mathcal{P}(C)$. It is easy to verify that the resulting set $\mathcal{P}(C)$ of paths satisfy

Properties P1—P4.

Case 3: This case happens if the paths in $\tilde{\mathcal{Q}}(C)$ form a canonical cycle. We denote the paths of $\tilde{\mathcal{Q}}(C)$ by $Q_1, \ldots, Q_h$ in the order of their appearance on the cycle. We define the following set of pairs of these paths: $\Pi = \{(Q_1, Q_2), (Q_3, Q_4), \ldots, (Q_{h-2}, Q_{h-1}), (Q_{h-1}, Q_h)\}$ (recall that h is an odd integer). Notice that every path appears in exactly one pair of Π, except for the path Q_{h-1}, that appears in two pairs.

Consider now some pair $(Q_i, Q_{i+1}) \in \Pi$. We construct a two-legged spider S_i, that consists of the path Q_i, and the sub-path of Q_{i+1}, from $t(Q_{i+1})$ to $v'(Q_{i+1}) = v(Q_i)$. In the resulting collection $S_1, S_3, \ldots, S_{h-2}, S_{h-1}$ of spiders, every pair of spiders are mutually vertex-disjoint, except for the vertices of Q_{h-1} that may appear in two spiders. We process each one of these spiders as in Case 2, to obtain a collection $\mathcal{P}(C)$ of $(h+1)/2$ paths in graph G that cause edge-congestion at most 2, and that satisfy Properties P1—P4. $\square$

2.6.2 Proof of Claim 2.6.4

Let f be the flow obtained by taking the union of the flows $f_1, \ldots, f_h$. It is easy to verify that flow f causes edge-congestion at most $4h \leq 8 \log n$. For every edge $e \in E(G)$, we say that a bad event $B(e)$ happens if e lies in more than $120 \log n$ graphs $T_1, \ldots, T_k$. It is enough to show that for each edge $e \in E(G)$, the probability of the event $B(e)$ is bounded by $1/n^6$; from the union bound over all edges e, it then follows that with probability at least $(1 - 1/n^3)$, the graphs in $\{T_1, \ldots, T_k\}$ cause edge-congestion at most $120 \log n$ (we have used the fact that for every pair (u, v) of vertices of G, there are at most k parallel edges (u, v) in G, and that $k \leq n$).

For the remainder of the proof, we fix an edge $e \in E(G)$, and we prove that the probability of event $B(e)$ is at most $1/n^6$.

For every vertex $v \in V(G)$, and index $1 \leq j \leq k$, we let $X(v, j)$ be a random variable whose value is 1 if the path $P_j(v)$ contains the edge e, and it is 0 otherwise. Notice that, if we denote $S = \sum_{v\in V(G)}\sum_{j=1}^{k} X(v,j)$, then the number of graphs $T_1,\ldots,T_k$ to which edge e belongs is exactly S. Moreover, the random variables in $\{X(v,j) \mid v \in V(G), 1 \leq j \leq k\}$ are independent from each other. Consider some vertex $v \in V(G)$, and let $F(v,e)$ be the total amount of flow that f sends on all flow-paths that originate from v and contain the edge e. Notice that for each $1 \leq j \leq k$, the probability that $X(v,j) = 1$ is $F(v,e)/F(v)$. Therefore, the expectation of $\sum_{1\leq j\leq k} X(v,j) = k \cdot F(v,e)/F(v) \leq 2F(v,e)$, since $F(v) \geq k/2$. Altogether, the expectation of $S = \sum_{v\in V(G)}\sum_{1\leq j\leq k} X(v,j)$ is at most $2\sum_{v\in V(G)} F(v,e)$, which is precisely the total amount of flow traversing e in f times 2, and is bounded by $8h \leq 16\log n$. To summarize, we are given a collection $\{X(v,j) \mid v \in V(G), 1 \leq j \leq k\}$ of independent 0/1 random variables. The expectation of their sum is at most $16\log n$. We need to bound the probability that $S > 120\log n$.

We use the following standard Chernoff bound (see e.g. [15]).

Theorem 2.6.8. *Let $\{Y_1,\ldots,Y_r\}$ be a collection of independent random variables taking values in $[0,1]$, and let $Y = \sum_i Y_i$. Assume that $\mathbf{E}[Y] \leq \mu$ for some value μ. Then for all $0 < \epsilon < 1$:*

$$\mathbf{Pr}\left[Y > (1+\epsilon)\mu\right] \leq e^{-\epsilon^2\mu/3}.$$

Using the above bound with $\epsilon = 1/2$ and $\mu = 80\log n$, we get that the probability that $S > 120\log n$ is bounded by $e^{-80\log n/12} < 1/n^6$.

2.6.3 Proof of Claim 2.6.5

We rename (LP-1) by (LP-Primal-1). Consider the following LP.

$$
\begin{array}{lll}
\text{(LP-Primal-2)} & \text{maximize } 0 & \\
\text{s.t.} & & \\
& \sum_{P\in\mathcal{P}(u)} f(P) \geq k & \forall u \in U \\
& \sum_{\substack{P\in\mathcal{P}^*:\\ e\in P}} f(P) \leq 2 & \forall e \in E(G) \\
& f(P) \geq 0 & \forall P \in \mathcal{P}^*
\end{array}
$$

It is clear that any feasible solution to (LP-Primal-1) is also a feasible solution to (LP-Primal-2), and vice versa. It is therefore sufficient to show that (LP-Primal-2) can be solved efficiently, if it has a feasible solution. Below is the Dual LP for (LP-Primal-2).

$$
\begin{array}{lll}
\text{(LP-Dual-1)} & \text{minimize } 2\cdot\sum_{e\in E(G)} \ell_e - k\cdot\sum_{u\in U} z_u & \\
\text{s.t.} & & \\
& \sum_{e\in P} \ell_e \geq z_u + z_{u'} & \forall u, u' \in U : u \neq u', \forall P \in \mathcal{P}(u)\cap\mathcal{P}(u') \\
& \sum_{e\in P} \ell_e \geq z_u & \forall u \in U, \forall P \in \mathcal{P}(u)\cap\mathcal{P}(s) \\
& z_u \geq 0 & \forall u \in U \\
& \ell_e \geq 0 & \forall e \in E(G)
\end{array}
$$

Recall that the number of vertices in G is n. Note that for (LP-Primal-2), the number of variables is exponential in n and the number of constraints is polynomial in n, while for (LP-Dual-1), the number of variables is polynomial in n and the number of constraints can be exponential in n. From the strong duality, the optimal objective value of (LP-Dual-1) is 0 if (LP-Primal-2) is feasible. We make a change to (LP-Dual-1) by replacing the objective function with a constraint that $2 \cdot \sum_{e \in E(G)} \ell_e - k \cdot \sum_{u \in U} z_u = 0$ to get the following LP.

(LP-Dual-2)

$$
\begin{aligned}
& 2 \cdot \textstyle\sum_{e \in E(G)} \ell_e - k \cdot \sum_{u \in U} z_u = 0 & & \\
& \textstyle\sum_{e \in P} \ell_e \geq z_u + z_{u'} & & \forall u, u' \in U : u \neq u', \forall P \in \mathcal{P}(u) \cap \mathcal{P}(u') \\
& \textstyle\sum_{e \in P} \ell_e \geq z_u & & \forall u \in U, \forall P \in \mathcal{P}(u) \cap \mathcal{P}(s) \\
& z_u \geq 0 & & \forall u \in U \\
& \ell_e \geq 0 & & \forall e \in E(G)
\end{aligned}
$$

Claim 2.6.9. *There exists an efficient separation oracle to (LP-Dual-2).*

We provide the proof of Claim 2.6.9 below, after we show that there is an efficient algorithm that solves (LP-Primal-2) using it. We run the Ellipsoid Algorithm on (LP-Dual-2) using the separation oracle, and let $\mathcal{C}$ be the set of all violated constraints that the oracle returns. Note that, since the running time of the Ellipsoid Algorithm is polynomial in the number of variables, when we run the Ellipsoid Algorithm on (LP-Dual-2), the size of $\mathcal{C}$, which is the number of violated constraints returned by the separation oracle, is at most polynomial in n. Let (LP-Dual-3) be a linear program whose set of constraints is precisely $\mathcal{C}$. Note that the linear program (LP-Dual-3) is feasible iff the linear program (LP-Dual-2) is feasible. This is because, if we run the Ellipsoid Algorithm on (LP-Dual-3), then the separation oracle will return the same set of constraints and the algorithm will return the same solution or report infeasible (if it reports infeasible on (LP-Dual-2)). We now compute the dual of (LP-Dual-3) and obtain a linear program that we denote by (LP-Primal-3). It is not hard to see that (LP-Primal-3) contains a subset (whose size is polynomial in n) of variables of (LP-Primal-2), and that for every constraint of (LP-Primal-2), there is a constraint in (LP-Primal-3), with the variables which are not in that subset omitted. From the strong duality, (LP-Primal-3) is feasible if (LP-Primal-2) is feasible. We can now solve (LP-Primal-3) efficiently, and the resulting solution is a feasible solution to (LP-Primal-2), as this is the same as setting all variables that do not correspond to the constraints in $\mathcal{C}$ to 0. This finishes the proof of Claim 2.6.5.

Proof of Claim 2.6.9: We now show that there exists a separation oracle to (LP-Dual-2). Given a suggested solution to (LP-Dual-2), the separation oracle needs to check if it satisfies all the constraints of (LP-Dual-2), and if not, return a violated constraint.

Let $\{z_u\}_{u\in U}, \{\ell_e\}_{e\in E(G)}$ be the suggested solution in an iteration. It is immediate to check whether the constraints $z_u \geq 0\ \forall u \in U$, the constraints $\ell_e \geq 0\ \forall e \in E(G)$ and the constraint $2 \cdot \sum_{e\in E(G)} \ell_e - k \cdot \sum_{u\in U} z_u = 0$ are satisfied. We will now show an efficient algorithm that checks whether the suggested solution satisfies the constraints $\sum_{e\in P} \ell_e \geq z_u + z_{u'}\ \forall u, u' \in U : u \neq u', \forall P \in \mathcal{P}(u) \cap \mathcal{P}(u')$ and the constraints $\sum_{e\in P} \ell_e \geq z_u\ \forall u \in U, \forall P \in \mathcal{P}(u) \cap \mathcal{P}(s)$ efficiently.

We assign each edge $e \in E(G)$ length ℓ_e. For any path P of G, we denote $\ell(P) = \sum_{e\in P} \ell_e$. Note that $U \subseteq V(G)$. We show an algorithm, that, given the suggested solution $\{z_u\}_{u\in U}, \{\ell_e\}_{e\in E(G)}$, either claims (correctly) that all constraints $\sum_{e\in P} \ell_e \geq z_u + z_{u'}\ \forall u, u' \in U : u \neq u', \forall P \in \mathcal{P}(u) \cap \mathcal{P}(u')$ and all constraints $\sum_{e\in P} \ell_e \geq z_u\ \forall u \in U, \forall P \in \mathcal{P}(u) \cap \mathcal{P}(s)$ are satisfied, or returns a pair u, u' of distinct vertices of U and a path $\hat{P}_{u,u'} \in \mathcal{P}(u) \cap \mathcal{P}(u')$, such that $\ell(\hat{P}_{u,u'}) < z_u + z_{u'}$ (which means that the constraint $\sum_{e\in \hat{P}_{u,u'}} \ell_e \geq z_u + z_{u'}$ is not satisfied by the suggested solution), or returns a vertex $u \in U$ and a path $\hat{P}_{u,s} \in \mathcal{P}(u) \cap \mathcal{P}(s)$, such that $\ell(\hat{P}_{u,s}) < z_u$ (which means that the constraint $\sum_{e\in \hat{P}_{u,s}} \ell_e \geq z_u$ is not satisfied by the suggested solution).

Claim 2.6.10. *There is an efficient algorithm, that, given any pair v, v' of vertices of G, computes the shortest path (with respect to edge lengths $\{\ell_e\}_{e\in E(G)}$) connecting v to v' that contains at most $2D$ edges.*

We will prove Claim 2.6.10 below, after we complete the proof of Claim 2.6.9 using it. For every pair $u, u' \in U$ of distinct vertices of U, let $\hat{P}_{u,u'}$ be the path returned by the algorithm in Claim 2.6.10, we check if $\ell(\hat{P}_{u,u'}) < z_u + z_{u'}$. For every vertex $u \in U$, let $\hat{P}_{u,s}$ be the path returned by the algorithm in Claim 2.6.10, we check if $\ell(\hat{P}_{u,s}) < z_u$. If there exists a pair $u, u' \in U$ of distinct vertices of U such that $\ell(\hat{P}_{u,u'}) < z_u + z_{u'}$, by definition, $\sum_{e \in \hat{P}_{u,u'}} \ell_e = \ell(\hat{P}_{u,u'}) < z_u + z_{u'}$. In this case, we claim that the constraint $\sum_{e \in \hat{P}_{u,u'}} \ell_e \geq z_u + z_{u'}$ is violated, and return this constraint as a violated constraint. If there does not exist a pair $u, u' \in U$ of distinct vertices of U such that $\ell(\hat{P}_{u,u'}) < z_u + z_{u'}$, then from Claim 2.6.10, for any pair u, u' of distinct vertices of U, for any path $P \in \mathcal{P}(u) \cap \mathcal{P}(u')$, we have $\sum_{e \in P} \ell_e \geq z_u + z_{u'}$. In this case, we know that all constraints $\sum_{e \in P} \ell_e \geq z_u + z_{u'}\ \forall u, u' \in U : u \neq u', \forall P \in \mathcal{P}(u) \cap \mathcal{P}(u')$ are satisfied, so we then proceed to check if there exists a vertex $u \in U$ such that $\ell(\hat{P}_{u,s}) < z_u$. If there does exists such a vertex u, by definition, $\sum_{e \in \hat{P}_{u,s}} \ell_e = \ell(\hat{P}_{u,s}) < z_u$. In this case, we claim that the constraint $\sum_{e \in \hat{P}_{u,s}} \ell_e \geq z_u$ is violated, and return this constraint as a violated constraint. If there does not exist a vertex $u \in U$ such that $\ell(\hat{P}_{u,s}) < z_u$, then from Claim 2.6.10, for vertex $u \in U$, for any path $P \in \mathcal{P}(u) \cap \mathcal{P}(s)$, we have $\sum_{e \in P} \ell_e \geq z_u$. We then claim that all constraints $\sum_{e \in P} \ell_e \geq z_u + z_{u'}\ \forall u, u' \in U : u \neq u', \forall P \in \mathcal{P}(u) \cap \mathcal{P}(u')$ and all constraints $\sum_{e \in P} \ell_e \geq z_u\ \forall u \in U, \forall P \in \mathcal{P}(u) \cap \mathcal{P}(s)$ are satisfied.

This finishes the description of the separation oracle to (LP-Dual-2). Since it is clear that the running time of the separation oracle is polynomial in n, this finishes the proof of Claim 2.6.9.

Proof of Claim 2.6.10: The algorithm employs dynamic programming. It is convenient to view the algorithm as constructing $2D + 1$ dynamic programming tables $\{\Pi_i\}_{0 \leq i \leq 2D}$. For each $0 \leq i \leq 2D$ and each pair v, v' of vertices of G, the table Π_i contains an entry $\Pi_i(v, v')$, that stores the shortest path $P^i_{v,v'}$ (with respect to edge lengths $\{\ell_e\}_{e \in E(G)}$) among all paths in G that connects v to v' and contains at most i edges, together with its length $\ell(P^i_{v,v'})$. So each entry $\Pi_i(v, v')$ has the form $\Pi_i(v, v') = (P^i_{v,v'}, L^i_{v,v'})$ where $L^i_{v,v'} = \ell(P^i_{v,v'})$. When such a path does not exist, we set $P^i_{v,v'}$ to be a default value $\perp$ and set $L^i_{v,v'} = +\infty$.

We now describe how to compute the entries of dynamic programming tables. First we initialize the entries in Π_0. For each vertex v, we set $P^0_{v,v}$ to be the path that contains a single node v, and we set $L^0_{v,v} = 0$. For each pair v, v' of distinct vertices of G, we set $P^0_{v,v} =\perp$ and $L^0_{v,v'} = +\infty$. For each $1 \leq i \leq 2D$, the table Π_i is computed based on G and the table Π_{i-1} as follows. For each vertex $v \in V(G)$, we denote $N(v) \subseteq V(G)$ to be the set of neighbors of v in G. For each pair $v, v' \in V(G)$, we set

$$L^i_{v,v'} = \min\{L^{i-1}_{v,v'}, \min_{w \in N(v)}\{\ell_{(v,w)} + L^{i-1}_{w,v'}\}\}.$$

For $P^i_{v,v'}$, we set it to be $\perp$ if $L^i_{v,v'} = +\infty$; we set it to be the same path as $P^{i-1}_{v,v'}$ if $L^{i-1}_{v,v'} \leq \min_{w \in N(v)}\{\ell_{(v,w)} + L^{i-1}_{w,v'}\}$; and if $w' = \arg\min\{L^{i-1}_{v,v'}, \min_{w \in N(v)}\{\ell_{(v,w)} + L^{i-1}_{w,v'}\}\}$ and $\ell_{(v,w')} + L^{i-1}_{w',v'} < L^{i-1}_{v,v'}$, we set it to be the concatenation of the edge (v, w') and the path $P^{i-1}(w', v')$.

Finally, given a pair v, v' of vertices of G, we return the path $P^{2D}_{v,v'}$ if $P^{2D}_{v,v'} \neq\perp$, and we claim that such a path does not exist if $P^{2D}_{v,v'} =\perp$. □ □

2.7 Applications to Distributed Computation

In this section we present several applications of low-diameter tree packing in the standard CONGEST model of distributed computation [32]. The proofs of all results in this section can be found in [13]. First, by the proof of Theorem 2.1.2 and the $O(\log n)$-approximation algorithm for edge connectivity by [19], we obtain the following result.

Theorem 2.7.1. *There is a randomized distributed algorithm, that, given an n-vertex graph G of constant diameter $D = O(1)$ and an integer λ, with high probability solves the problem of $O(\log n)$-approximate verification of λ-edge connectivity in G in* $\mathrm{poly}(\lambda \cdot \log n)$ *rounds.*

This improves upon the state of the art bound of $O(\sqrt{n})$ for graphs with constant diameter $D \geq 3$, and $\lambda \leq n^c$ for some positive constant $c < 1/(2D^2)$. From now on, we restrict our attention to k-edge connected graphs with a constant diameter $D = O(1)$. We employ the modular approach for distributed optimization introduced by Ghaffari and Haeupler in [18] which is based on the notion of *low-congestion shortcuts*. Roughly speaking, these shortcuts augment vertex-disjoint connected subgraphs by adding nearly-edge disjoint subsets of "shortcut" edges (that is, edges that reduce the diameter of each subgraph). Using our tree packing construction, we provide improved shortcuts for highly connected graphs of small diameter. This immediately leads to $o(\sqrt{n})$-round algorithms for several classical graph problems. For example, we prove the following:

Theorem 2.7.2. *There is a randomized distributed algorithm, that, given a k-edge connected weighted n-vertex graph G of diameter D, such that the nodes know an $O(\log n)$ approximation of k, computes an MST of G in $\widetilde{O}(\min\{\sqrt{n/k}+n^{D/(2D+1)}, n/k\})$ rounds with high probability.*

If the nodes do not know an $O(\log n)$-approximation of the value of k, then such an approximation can be computed in $\mathrm{poly}(k \log n)$ rounds for $D = O(1)$ using Theorem 2.7.1, w.h.p. For general graphs (of an arbitrary connectivity) with diameter $D = 3, 4$, Kitamura et al. [25] showed nearly optimal constructions of MST's (based on shortcuts) with round complexities of $\widetilde{O}(n^{1/4})$ and $\widetilde{O}(n^{1/3})$ respectively. Turning to lower bounds, we slightly modify the construction of Lotker et al. [28] to obtain a lower bound of $\Omega((n/k)^{1/3})$ rounds for computing an MST in k-edge connected graphs of diameter 4, assuming that $k = O(n^{1/4})$.

Finally, we consider the basic task of *information dissemination*, where a given source vertex s is required to send N bits of information to the designated target vertex t in a k-edge connected n-vertex graph. This problem was first addressed in [19], who showed a lower bound of $\Omega(\min\{N/\log^2 n, n/k\})$ rounds, provided that the diameter of the graph is $\Theta(\log n)$. Using our low-diameter tree packing we obtain the first improved upper bounds for sublogarithmic diameter. We also show a new lower bound for simple store-and-forward algorithms, for the regime where $D = o(\log n)$.

Theorem 2.7.3. *There is a randomized distributed algorithm, that, given any k-edge con-*

nected n-vertex graph G of diameter D with a source vertex s and a destination vertex t, sends an input sequence of N bits from s to t. The number of rounds is bounded by $\widetilde{O}(N^{1-1/(D+1)} + N/k)$ with high probability. In addition, for all integers n, N, D and $k \leq n$, there exists a k-edge connected n-vertex graph $G = (V, E)$ of diameter $2D$, and a pair s, t of its vertices, such that sending N bits from s to t in a store-and-forward manner requires at least $\Omega(\min\{(N/(D \log n))^{1-1/(D+1)}, n/k\} + N/k + D)$ rounds.

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] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2), 2009.

[4] Sandeep N Bhatt and Frank Thomson Leighton. A framework for solving vlsi graph layout problems. *Journal of Computer and System Sciences*, 28(2):300–343, 1984.

[5] Sergio Cabello. Hardness of approximation for crossing number. *Discrete & Computational Geometry*, 49(2):348–358, 2013.

[6] Chandra Chekuri, Sanjeev Khanna, and F Bruce Shepherd. Edge-disjoint paths in planar graphs. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 71–80. IEEE, 2004.

[7] Chandra Chekuri and Anastasios Sidiropoulos. Approximation algorithms for euler genus and related problems. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 167–176. IEEE, 2013.

[8] Markus Chimani and Petr Hliněnỳ. A tighter insertion-based approximation of the crossing number. In *International Colloquium on Automata, Languages, and Programming*, pages 122–134. Springer, 2011.

[9] Julia Chuzhoy. An algorithm for the graph crossing number problem. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 303–312. ACM, 2011.

[10] Julia Chuzhoy and Sanjeev Khanna. Algorithms for single-source vertex connectivity. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 105–114. IEEE, 2008.

[11] Julia Chuzhoy, Vivek Madan, and Sepideh Mahabadi. In *Private Communication*, 2016.

[12] Julia Chuzhoy, Yury Makarychev, and Anastasios Sidiropoulos. On graph crossing number and edge planarization. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms*, pages 1050–1069. SIAM, 2011.

[13] Julia Chuzhoy, Merav Parter, and Zihan Tan. On packing low-diameter spanning trees. *arXiv preprint arXiv:2006.07486*, 2020.

[14] Julia Chuzhoy and Zihan Tan. A subpolynomial approximation algorithm for graph crossing number in low-degree graphs. *arXiv preprint arXiv:2202.06827*, 2022.

[15] Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[16] Guy Even, Sudipto Guha, and Baruch Schieber. Improved approximations of crossings in graph drawings and vlsi layout areas. *SIAM Journal on Computing*, 32(1):231–252, 2002.

[17] M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM J. Algebraic Discrete Methods*, 4(3):312–316, 1983.

[18] Mohsen Ghaffari and Bernhard Haeupler. Distributed algorithms for planar networks

ii: Low-congestion shortcuts, mst, and min-cut. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 202–219. SIAM, 2016.

[19] Mohsen Ghaffari and Fabian Kuhn. Distributed minimum cut approximation. In *International Symposium on Distributed Computing*, pages 1–15. Springer, 2013.

[20] P. Hlinený. Crossing number is hard for cubic graphs. *J. Comb. Theory, Ser. B*, 96(4):455–471, 2006.

[21] Tomáš Kaiser. A short proof of the tree-packing theorem. *Discrete Mathematics*, 312(10):1689–1691, 2012.

[22] David R Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.

[23] Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Polylogarithmic approximation for minimum planarization (almost). In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 779–788, 2017.

[24] Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Polylogarithmic approximation for euler genus on bounded degree graphs. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 164–175. ACM, 2019.

[25] Naoki Kitamura, Hirotaka Kitagawa, Yota Otachi, and Taisuke Izumi. Low-congestion shortcut and graph parameters. In *33rd International Symposium on Distributed Computing (DISC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[26] F. T. Leighton. *Complexity issues in VLSI: optimal layouts for the shuffle-exchange graph and other networks.* MIT Press, 1983.

[27] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their

use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999.

[28] Zvi Lotker, Boaz Patt-Shamir, and David Peleg. Distributed MST for constant diameter graphs. *Distributed Computing*, 18(6):453–460, 2006.

[29] J. Matoušek. *Lectures on discrete geometry*. Springer-Verlag, 2002.

[30] CSJA Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, 1(1):445–450, 1961.

[31] J. Pach and G. Tóth. Thirteen problems on crossing numbers. *Geombinatorics*, 9(4):194–207, 2000.

[32] David Peleg. *Distributed Computing: A Locality-sensitive Approach*. SIAM, 2000.

[33] 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.

[34] Neil Robertson and Paul D Seymour. Graph minors. vi. disjoint paths across a disc. *Journal of Combinatorial Theory, Series B*, 41(1):115–138, 1986.

[35] P. Turán. A note of welcome. *J. Graph Theory*, 1:1–5, 1977.

[36] William Thomas Tutte. On the problem of decomposing a graph into n connected factors. *Journal of the London Mathematical Society*, 1(1):221–230, 1961.

[37] Imrich Vrto. Crossing numbers of graphs: A bibliography. `http://www.ifi.savba.sk/~imrich`.

[38] Hassler Whitney. Congruent graphs and the connectivity of graphs. In *Hassler Whitney Collected Papers*, pages 61–79. Springer, 1992.