Routing Symmetric Demands in Directed Minor-Free Graphs with Constant Congestion

Timothy Carpenter
Dept. of Computer Science & Engineering, The Ohio State University, Columbus, OH, USA
carpenter.454@osu.edu

Ario Salmasi
Dept. of Computer Science & Engineering, The Ohio State University, Columbus, OH, USA
salmasi.1@osu.edu

Anastasios Sidiropoulos
Dept. of Computer Science, University of Illinois at Chicago, USA
sidiropo@uic.edu

Abstract

The problem of routing in graphs using node-disjoint paths has received a lot of attention and a polylogarithmic approximation algorithm with constant congestion is known for undirected graphs [Chuzhoy and Li 2016] and [Chekuri and Ene 2013]. However, the problem is hard to approximate within polynomial factors on directed graphs, for any constant congestion [Chuzhoy, Kim and Li 2016].

Recently, [Chekuri, Ene and Pilipczuk 2016] have obtained a polylogarithmic approximation with constant congestion on directed planar graphs, for the special case of symmetric demands. We extend their result by obtaining a polylogarithmic approximation with constant congestion on arbitrary directed minor-free graphs, for the case of symmetric demands.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Routing, Node-disjoint, Symmetric demands, Minor-free graphs

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2019.14

Category APPROX

Funding This work was supported by NSF under CAREER award 1453472 and grant CCF 1815145.

1 Introduction

Routing in graphs along disjoint paths is a fundamental problem with numerous applications in various domains [1, 2, 3, 23, 24]. Disjoint path problems have been well-studied in both the directed and undirected setting, and it is known that the directed formulations of these problems are generally more difficult to approximate [14, 11]. The recent work of [5, 6] has brought to light a more tractable formulation of the directed version of these problems by considering routing symmetric demand pairs with constant congestion.

Two of the most well-known and studied disjoint path problems are the node-disjoint paths problem (NDP) and the edge-disjoint paths problems (EDP). In these problems, the goal is to connect a set of node pairs through node- or edge-disjoint paths in an undirected graph. It is known that the decision version of NDP is NP-complete [20], and it has been shown to be fixed parameter tractable [26]. But there remain gaps in our understanding of their approximability. For both EDP and NDP on n-node graphs, the state of the art is an $O(\sqrt{n})$-approximation [9], [22]. For planar graphs, a slightly better bound of $\tilde{O}(n^{9/19})$-approximation is known [13]. Even for the case of the grid, only a $\tilde{O}(n^{1/4})$-approximation for NDP is known [12]. For hardness of approximation, it is known that both NDP and EDP are $2^{\Omega\left(\sqrt{\log n}\right)}$-hard to approximate, unless all problems in NP have algorithms with running time $n^{\log n}$ [14].

Progress has been made on relaxed versions of these problems. One such relaxation is the all-or-nothing flow problem (ANF), where a subset $\mathcal{M}' \subseteq \mathcal{M}$ is routed if there is a feasible multicommodity flow routing one unit of flow for each pair in $\mathcal{M}'$. Poly-logarithmic approximations are known for ANF [8, 7]. Another relaxation is to allow some small constant congestion on the nodes or edges. For this relaxation, poly-logarithmic approximations have been obtained for EDP with congestion 2 [15], and for NDP with congestion $O(1)$ [4].

It is natural to extend the study of disjoint path problems to directed graphs. However, these problems are known to be significantly harder on directed graphs. Even the case of ANF with constant congestion c allowed has an $n^{\Omega(1/c)}$ inapproximability bound [11]. However, a more tractable case is found by considering symmetric demand pairs. The study of maximum throughput routing problems in directed graphs with symmetric demand pairs began in [5]. In this setting the graph G is directed, and routing a source-destination pair (s_i, t_i) requires finding a path from s_i to t_i and a path from t_i to s_i. We let Sym-Dir-ANF be the analogue of ANF, and Sym-Dir-NDP be the analogue of NDP in this setting. A poly-logarithmic approximation for Sym-Dir-ANF is obtained in [5]. Subsequently, in [6] a randomized poly-logarithmic approximation with constant congestion on planar graphs for Sym-Dir-NDP is obtained.

1.1 Our contribution

We consider the problem of routing symmetric demands along node-disjoint paths in directed graphs. We refer to this problem as Sym-Dir-NDP. Letting $G = (V, E)$ be a directed graph with unit node capacities and $\mathcal{M} = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$ be a set of source-destination pairs, we say that $(G, \mathcal{M})$ is an instance of Sym-Dir-NDP. Routing a pair (s_i, t_i) requires finding a path from s_i to t_i, and from t_i to s_i. A solution to an instance of Sym-Dir-NDP is a routing strategy maximizing the number of pairs routed through disjoint paths. We refer to a solution having congestion ζ, if no vertex is used in more than ζ paths. Our contribution generalizes the result from [6] from the class of directed planar graphs to arbitrary directed minor-free graphs. We now formally state our results and briefly highlight the methods used. Our main result is the following.

▶ **Theorem 6.** *Let G be an H-minor free graph. There is a polynomial time randomized algorithm that, with high probability, achieves an $\Omega\left(\frac{1}{h^7\sqrt{h}\log^{5/2}(n)}\right)$-approximation with congestion 5 for Sym-Dir-NDP instances in G, where h is an integer dependent only on H.*

The approximation algorithm in this theorem is obtained by extending the algorithm of [6]. For an instance $(G, \{(s_1, t_1), \ldots, (s_k, t_k)\})$ of Sym-Dir-NDP, we say that the set $\mathcal{T} = \{s_1, \ldots, s_k\} \cup \{t_1, \ldots, t_k\}$ is the set of terminals. Speaking broadly, the algorithm obtained in Theorem 6 consists of the following steps.

1. Using a multicommodity flow based LP relaxation and the well-linked decomposition of [6], reduce to an instance in which the terminals $\mathcal{T}$ are α-well-linked for a fixed constant α.
2. Find a large routing structure connected to a large proportion of the terminals.
3. Use the routing structure to connect a large number of the source-destination pairs.

From here on, we shall refer to the routing structure as the crossbar. The reduction we use in Step 1 allows us to reduce an instance of Sym-Dir-NDP to an instance on an Eulerian graph of small maximum degree, and where the terminals are α-well-linked. This comes at the cost of then having a randomized algorithm for the original instance. This reduction comes from [6], and while there it is used for planar graphs, we were fortunate in that it can also be used for general graphs. The routing scheme of Step 3 is also thanks to [6], and relies on a similar crossbar construction. Our main contribution to this line of research is in finding an appropriate crossbar construction for Step 2.

To build our crossbar, we would ideally find a “flat” grid minor so that some constant fraction of the terminal pairs can be routed along node-disjoint paths to the interface of the grid minor (a “flat” grid minor is one in which the grid minor is connected with the rest of the graph only through the outer face). Then we would have the following sets of node-disjoint paths along which to route the terminal pairs: the paths from the terminals to the interface, the paths from terminals to terminals implied by the node-well-linked property of the terminals, the concentric cycles of the grid minor, and the paths connecting the outermost and innermost cycles of the grid minor. From these, just as in [6] we can construct a routing scheme with congestion 5. To find a suitable flat grid minor, we combine results of [10] and [28] to show that flat grid minors of a suitable size can be found. We then show that if for the flat grid minor produced we cannot route a large enough fraction of the terminals to the interface then there exists some vertex which can be eliminated from the graph without destroying a potential solution to the problem. Thus, we find and test flat grid minors until one suitable to be used in the crossbar is found.

2 Notation and Preliminaries

We now introduce some notation and definitions that are used throughout the paper.

Directed and undirected graphs

From any directed graph G we can obtain an undirected graph G^{UN} as follows. We set $V(G^{\mathsf{UN}}) = V(G)$ and $E(G^{\mathsf{UN}}) = \{\{u, v\} : (u, v) \in E(G) \vee (v, u) \in E(G)\}$. We refer to G^{UN} as the *underlying undirected graph* of G.

Flat subgraphs

We say that a planar subgraph H of an undirected graph G is *flat* if there exists a planar drawing Φ of H such that for any $\{u, v\} \in E(G)$, with $u \in V(H)$ and $v \in V(G) \setminus V(H)$, we have that u is on the outer face of Φ.

Well-linked sets

Let G be a directed (resp. undirected) graph. A set $X \subseteq V(G)$ is node-well linked in G if for any two disjoint subsets $Y, Z \subset X$ of equal size, there exist $|Y|$ node-disjoint directed (resp. undirected) paths from Y to Z, such that each vertex in Y is the start of exactly one path, and each vertex in Z is the end of exactly one path. For some $\alpha \in (0, 1)$, we say that X is α-node well-linked if for any two disjoint subsets $Y, Z \subset X$ of equal size, there exist $|Y|$ directed (resp. undirected) paths from Y to Z such that no vertex is in more than $1/\alpha$ of these paths; In other words, we allow a node congestion of $1/\alpha$ for these paths.

Directed and undirected treewidth

For a directed graph G, we will denote by $\mathsf{dtw}(G)$ the *directed treewidth of* G, and we will denote by $\mathsf{tw}(G^{\mathsf{UN}})$ the (undirected) *treewidth of* G^{UN}. Directed treewidth is a global connectivity measure introduced in [19, 25], and just as undirected treewidth is defined by the minimum width tree decomposition, directed treewidth is defined by the minimum size of what is termed an arboreal decomposition. Instead of providing the full definitions of directed and undirected treewidth here, we only ask the reader to make a note of the following two important facts:

- If G is an Eulerian directed graph with max degree Δ, then $\mathsf{tw}(G^{\mathsf{UN}}) \leq \mathsf{dtw}(G) = O(\Delta \cdot \mathsf{tw}(G^{\mathsf{UN}}))$ [19].
- If a directed graph G contains an α-well-linked set X, then $\mathsf{dtw}(G) = \Omega(\alpha|X|)$ [25].

Clique-sums

Let G_1 and G_2 be two graphs. A *clique-sum* of G_1 and G_2 is any graph that is obtained by identifying a clique in G_1 with a clique of the same size in G_2, and then possibly removing some edges in the resulting shared clique. An h*-clique-sum*, or h*-sum* for short, is a clique-sum where the identified cliques have at most h vertices.

Nearly-embeddable and minor-free graphs

We say that a graph is (a, g, k, p)*-nearly embeddable* if it is obtained from a graph of Euler genus g by adding a apices and k vortices of pathwidth p. We say that a graph is h-nearly embeddable if it is (a, g, k, p)-nearly embeddable for some $a, g, k, p \leq h$. The following is implicit in [27].

▶ **Theorem 1** (Robertson and Seymour [27]). *Let H be any graph. Every H-minor-free graph can be obtained by at most h-clique-sums of graphs that are h-nearly embeddable graphs, where h is a non-negative integer dependent on H.*

Note that the result of the above theorem is existential. Demaine, Hajiaghayi, and Kawarabayashi in [16] provide an algorithm to compute this decomposition in polynomial time, for any fixed minor H.

3 The Algorithm for Minor-Free Graphs

We first use the following result of [6] to reduce the problem to the case of Eulerian graphs with small degrees. Note that this result is stated for planar graphs in [6], but the proof does not use planarity, and thus can be stated for general graphs.

▶ **Lemma 2** (Chekuri, Ene & Pilipczuk [6]). *Suppose that there is a polynomial time algorithm for $\Omega(1)$-node-well-linked instances of Sym-Dir-NDP in directed Eulerian graphs of maximum degree Δ that achieves a $\beta(\Delta)$-approximation with congestion c. Then there is a polynomial time randomized algorithm that, with high probability, achieves a $\beta(O(\log^2 k)) \cdot O(\log^6 k)$-approximation with congestion c for arbitrary instances of Sym-Dir-NDP in directed graphs, where k is the number of pairs in the instance.*

Now we describe how to construct the crossbar in minor-free graphs, assuming that we are given a $m \times m$ flat grid minor Γ, for some large enough m, and a family of λm node-disjoint paths connecting the set of terminals and the interface of Γ, for some constant λ. The following is our main technical result, which is similar to the one in [6] for planar graphs. We use here a generalized notion of *enclosed* for flat grids in non-planar graphs. Let H be a directed graph with a flat grid minor η. Let u^{out} be an arbitrary vertex not contained in η. Let C be some cycle contained within η. We say a vertex u is *enclosed* by C if all paths in H^{UN} from u to u^{out} intersect C. We now find the desired concentric cycles in G. The proof is deferred to Section 4.

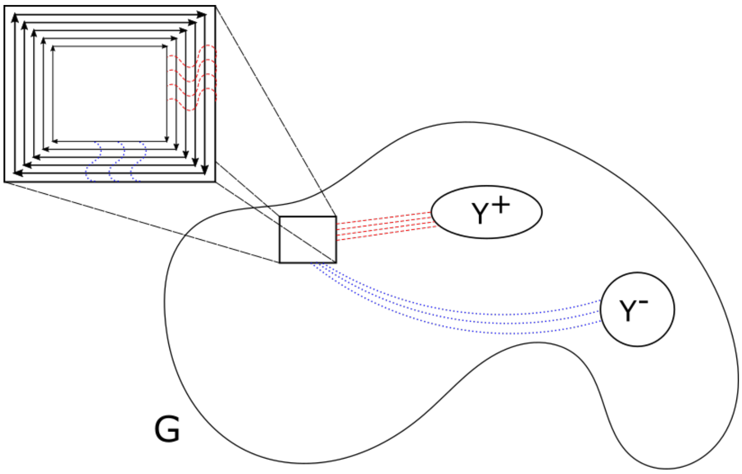

Figure 1 An example for case (1) of Theorem 3. The red paths are the node disjoint paths in P^+, going from Y^+ to the innermost of the concentric cycles, and the blue paths correspond to the node disjoint paths in P^-, going from Y^- to the innermost of the concentric cycles.

▶ **Theorem 3.** *Let G be a directed minor-free graph of maximum in-degree of at most Δ. Let X be an α-node-well-linked set in G with $|X| = \Omega\left(\frac{\Delta^2}{\alpha}\right)$. Let $m = \Omega\left(\frac{\alpha|X|}{\beta}\right)$, where β is a non-negative number dependent on G. Suppose that we can find a $m \times m$ flat grid minor Γ of G^{UN}, and a family of λm node-disjoint paths connecting X and the interface of Γ in G^{UN}, for some $0 < \lambda \leq 1$. One can in polynomial time find a set of $\Omega\left(\frac{\alpha|X|}{\beta\Delta}\right)$ concentric directed cycles going in the same direction w.r.t. a planar embedding of Γ (all clockwise or counter-clockwise), sets $Y^+, Y^- \in X$ with $|Y^+| = |Y^-| = \Omega\left(\frac{\alpha^2|X|}{\beta\Delta^2}\right)$, and families P^+ and P^- of node-disjoint paths such that one of the following holds.*

(1) *None of the cycles enclose any vertex of $Y^+ \cup Y^-$, the family P^+ consists of $|Y^+|$ node-disjoint paths from Y^+ to the innermost cycle, and the family P^- consists of $|Y^-|$ node-disjoint paths from the innermost cycle to Y^- (See Figure 1).*

(2) *All cycles enclose $Y^+ \cup Y^-$, the family P^+ consists of $|Y^+|$ node-disjoint paths from $|Y^+|$ to the outermost cycle, and the family P^- consists of $|Y^-|$ node-disjoint paths from the outermost cycle to Y^-.*

In order to obtain such a crossbar, we need to find a flat grid minor of large enough size. The following Lemma provides us the desired flat grid minor, and the proof is deferred to Section 6.2.

▶ **Lemma 4.** *Let H be any graph and let G be an H-minor-free directed graph with treewidth t. Let X be an α-node-well-linked set in G with $|X| = \Omega\left(\frac{\Delta^2}{\alpha}\right)$. One can in polynomial time find a $r \times r$ flat grid minor Γ in G^{UN}, with $r = \Omega\left(\frac{t}{h^7\sqrt{h}\log^{5/2}(n)}\right)$, and a family of r node-disjoint paths connecting X and the interface of Γ, where h is an integer dependent only on the structure of H.*

Once we obtain a crossbar as described above, we can route a large subset of terminal pairs.

▶ **Lemma 5.** *Given the crossbar described in Theorem 3, one can get an $O\left(\frac{\Delta^2}{\beta\alpha^3}\right)$-approximation algorithm with congestion 5 for Sym-Dir-NDP in instances for which the input graph is minor-free and Eulerian with maximum in-degree Δ, the set of terminals is α-node-well-linked for some $\alpha \leq 1$, and β is dependent only on H.*

Proof. By applying the same routing scheme as in the one in [6], we get the desired result. ◀

Now we are ready to state the main result of this paper.

▶ **Theorem 6.** *Let G be a H-minor-free graph. There is a polynomial time randomized algorithm that, with high probability, achieves an $\Omega\left(\frac{1}{h^7\sqrt{h}\log^{5/2}(n)}\right)$-approximation with congestion 5 for Sym-Dir-NDP instances in G, where h is an integer dependent only on H.*

Proof. This is immediate by Lemmas 2, 4, 5, and Theorem 3. ◀

4 The Crossbar Construction

In this section we discuss the construction of the crossbar stated in Theorem 3. Before we give the proof of this Theorem we establish some auxiliary facts. Throughout this subsection, we assume that we are given the input of Theorem 3.

▶ **Lemma 7.** *One can in polynomial time find an integer $r = \Omega\left(\frac{\alpha|X|}{\beta}\right)$ and a sequence of node-disjoint concentric undirected cycles $C_1, C_2, \ldots, C_r$ in G^{UN}, with C_1 being the outermost and C_r being the innermost cycle.*

Proof. Let t be the treewidth of G^{UN}. Since X is α-node-well-linked in G, X is also α-node-well-linked in G^{UN}. Thus, $t = \Omega(\alpha|X|)$. Let Γ be a flat $m \times m$ grid minor of G^{UN}, as given in the input of Theorem 3. By losing a constant factor, we can construct a flat sub-divided $r \times r$ wall in G^{UN}, with $r = \Omega\left(\frac{\alpha|X|}{\beta}\right)$. Let C_1 be the outermost cycle of Γ, and for each $i \in \{2, \ldots, r\}$, let C_i be the outermost cycle of $\Gamma \setminus \cup_{1 \leq j < i} V(C_i)$. ◀

As in [6], for a vertex set $Q \subseteq V(G^{\mathsf{UN}})$, a vertex $v \notin Q$, and an integer $\ell \geq 2\Delta$, we say that a vertex set S is a (v, Q, ℓ)-*isle* if $v \in S$, $G^{\mathsf{UN}}[S]$ is connected, $S \cap Q = \emptyset$, and $|N_{G^{\mathsf{UN}}}(S)| \leq \ell$. Let $C_1, \ldots, C_r$ be the sequence of node-disjoint concentric undirected cycles in G^{UN} obtained from Lemma 7. We set isles S^{out} and S^{in} by choosing an arbitrary vertex v^{out} in C_1, and an arbitrary vertex v^{in} in C_r. Letting $\ell = \lfloor r/(4\Delta + 2) \rfloor$, then S^{out} is the $(v^{\mathsf{out}}, X, \ell)$-isle and S^{in} is the $(v^{\mathsf{in}}, X, \ell)$-isle obtained. We also need that S^{out} and S^{in} are separated by many cycles. For this, we use the following Lemma of [6], the proof of which is slightly modified.

▶ **Lemma 8.** *The isle S^{out} does not contain any vertex that is enclosed by $C_{\ell+1}$, and the isle S^{in} does not contain any vertex that is not strictly enclosed by $C_{r-\ell}$.*

Proof. The proofs for S^{in} and S^{out} are symmetrical, so we focus on the case of S^{out}. Assume that S^{out} contains a vertex enclosed by $C_{\ell+1}$, and we will find a contradiction. Since $v^{out} \in S^{\mathsf{out}}$, S^{out} is connected in G^{UN}, and Γ is a flat wall, it must be that S^{out} contains a vertex from every cycle C_i, $1 \leq i \leq \ell + 1$. Since $|N_{G^{\mathsf{UN}}}(S^{\mathsf{out}})| \leq \ell$, for some $1 \leq i \leq \ell + 1$ we have that $V(C_i)$ is completely contained in S^{out}. However, there are $r > \ell$ vertex-disjoint paths in G^{UN} connecting C_i with X. Thus, either $S^{\mathsf{out}} \cap X \neq \emptyset$ or $|N_{G^{\mathsf{UN}}}(S^{\mathsf{out}})| > \ell$, both of which are contradictions. ◀

We are almost ready to prove the main result of this section. We will make use of the following Lemma, which is implicit in [6]. Note that sets S'^{in} and S'^{out}, the concentric cycles $C'_1, \ldots, C'_{r'}$, and integers r' and Δ' in the next Lemma are defined for a planar graph G' as described in [6].

▶ **Lemma 9.** *Let G' be an Eulerian, planar directed graph, with sets $S'^{\mathsf{in}}, S'^{\mathsf{out}}$ separated by concentric cycles $C'_1, \ldots, C'_{r'}$, and let $\ell' = \lfloor r'/(4\Delta' + 2) \rfloor$, where Δ' is the the maximum in-degree of G'. Then one can in polynomial time find $\lceil \ell'/2 \rceil$ node-disjoint directed concentric*

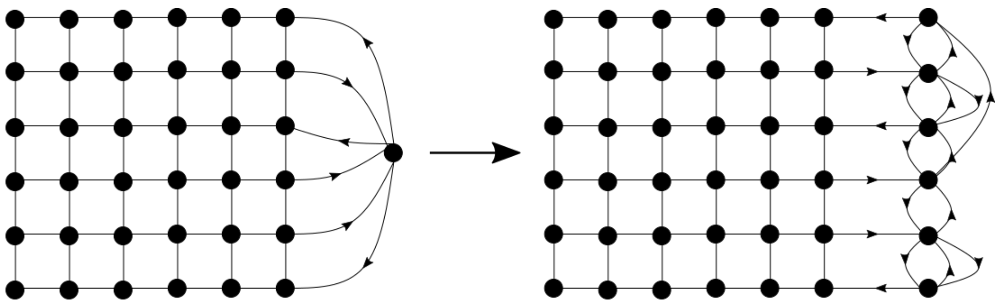

Figure 2 Maintaining an Eulerian graph with bounded degree.

cycles, all going in the same direction (all clockwise or all counter-clockwise), such that all vertices of S'^{in} are strictly enclosed by the innermost cycle, and all vertices of S'^{out} are not enclosed by the outermost cycle, or vice versa, with the roles of S'^{in} and S'^{out} swapped.

We will use Lemma 9 to find concentric cycles in minor-free graphs. We first generalize the notion of *enclosed* for flat grids in non-planar graphs. Let H be a directed graph with a flat grid minor η. Let u^{out} an arbitrary vertex not contained in η. Let C be some cycle contained within η. We say a vertex u is *enclosed* by C if all paths in H^{UN} from u to u^{out} intersect C. We now find the desired concentric cycles in G.

▶ **Lemma 10.** *One can in polynomial time find $\lceil \ell/2 \rceil$ node-disjoint directed concentric cycles in G, all going in the same direction (all clockwise or all counter-clockwise), such that all vertices of S^{in} are enclosed by the innermost cycle, and all vertices of S^{out} are not enclosed by the outermost cycle, or vice versa, with the roles of S^{in} and S^{out} swapped.*

Proof. We proceed by creating G' from G as follows. Let

$$Z = \{v \in V(G) : v \in V(C_1) \text{ or } v \text{ is not in the component of } G \setminus V(C_1) \text{ containing } C_2\}.$$

Let $G' = G/Z$, i.e. G' is the graph created by identifying all vertices in Z to a single vertex z. Since G is Eulerian, G' is also Eulerian. Furthermore, we can delete any self-loops on z, and G' is still Eulerian. Since $C_1, \ldots, C_r$ are contained within a flat grid minor of G, G' is therefore a planar graph. The only impediment to directly applying Lemma 9 is that the in-degree δ of z might be greater than Δ. We can eliminate this by replacing z with a path P of length δ, with edges directed both ways between adjacent vertices. We then connect the vertices formerly connected to z to vertices in P, maintaining the planarity of G'. Then, to restore G' as an Eulerian graph, for the vertices in P with an imbalance between in- and out-degree we can create a new edge (See Figure 2).

After these modifications, G' is an Eulerian, planar digraph with maximum in-degree Δ. Let $S'_{\mathsf{in}} = S^{\mathsf{in}}$ and $S'_{\mathsf{out}} = (S^{\mathsf{out}} \cap V(G')) \cup \{z\}$. We now apply Lemma 9 using G', S'_{in}, and S'_{out} to find $\lceil \ell/2 \rceil$ node-disjoint directed concentric cycles, all going in the same direction, and all vertices of S'_{in} are strictly enclosed by the innermost cycle, and all vertices of S'_{out} are not enclosed by the outermost cycle. Clearly, each of these cycles exists in G, all vertices of S^{in} are strictly enclosed by the innermost cycle, and all vertices of S^{out} are not enclosed by the outermost cycle. ◀

We are now ready to obtain the proof of Theorem 3.

Proof of Theorem 3. By Lemma 10, we can finish the construction of the crossbar with the same argument as in [6]. ◀

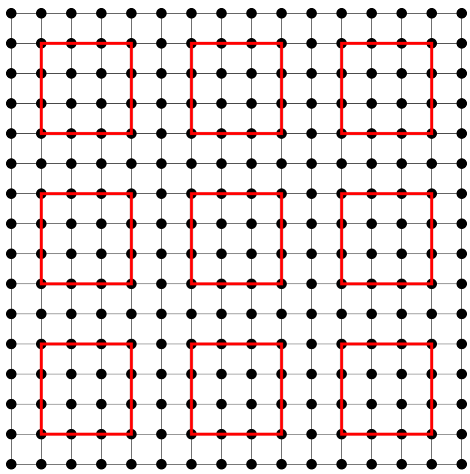

Figure 3 Decomposition of the grid minor.

5 Graphs of Bounded Genus

In this section we describe an algorithm to construct a flat grid minor Γ of large enough size in graphs of bounded genus. The following is implicit in the work of Chekuri and Sidiropoulos [10].

▶ **Lemma 11.** *Let G be an undirected graph of Euler genus $g \geq 1$, with treewidth $t \geq 1$. There is a polynomial time algorithm that computes a $r' \times r'$-grid as a minor, with $r' = \Omega\left(\frac{t}{g^3 \log^{5/2} n}\right)$. Furthermore, the algorithm does not require a drawing of G as part of the input.*

We need to find a flat grid minor for our purpose. Thomassen in [28] shows that if a graph of genus g contains a $m \times m$-grid as a minor, then it contains a $k \times k$ flat grid minor, where $m > 100k\sqrt{g}$. With some minor modifications, we can use this result to obtain the following.

▶ **Lemma 12.** *Let G be an undirected graph of Euler genus $g \geq 1$, and let H be a $m \times m$ grid minor of G. Let $k < \frac{m}{100\sqrt{g}}$ be an integer. Then one can compute a $k \times k$ flat grid minor of G in polynomial time.*

Proof. Thomassen in [28] shows that in order to find the desired flat grid minor, it is enough to construct a family of pairwise disjoint subgraphs $Q_1, Q_2, \cdots, Q_{2g+2}$ of H, satisfying the following conditions.

(1) Each Q_i is a $k \times k$ sub-grid of H.

(2) For any i, j with $1 \leq i < j \leq 2g+2$, we have the following. If x_i and x_j are on the outer cycles of Q_i and Q_j respectively, and they have neighbors $y_i \in V(H) \setminus V(Q_i)$ and $y_j \in V(H) \setminus V(Q_j)$ respectively, then H has a path P_{ij} from x_i to x_j such that $V(P_{ij}) \cap \left(\bigcup_{r=1}^{2g+2} V(Q_r)\right) = \{x_i, x_j\}$.

Since we have $m > 100k\sqrt{g}$, this construction can be easily done as shown in Figure 3, and thus one of the Q_i's is flat, as desired. ◀

▶ **Lemma 13.** *Let G be an undirected graph of Euler genus $g \geq 1$, with treewidth $t \geq 1$. There exists a polynomial time algorithm that computes a $r \times r$-grid as a minor, with $r = \Omega\left(\frac{t}{g^3\sqrt{g}\log^{5/2} n}\right)$. Moreover, the algorithm does not require a drawing of G as part of the input.*

Proof. This is immediate by Lemmas 11 and 12. ◀

Note that computing a large grid minor in the graph is not enough. We need to make sure that a large number of terminals can reach the interface of the grid minor. The following Lemma will provide for us the desired grid minor. The proof of this Lemma is deferred to Appendix A.

▶ **Lemma 14.** *Let $\mathcal{F}$ be some minor-closed family of graphs, let $\alpha \leq 1$, and $\beta > 0$. Suppose that there exists a polynomial-time algorithm which given, some $G' \in \mathcal{F}$ and some α-node-well-linked set X' in G', outputs some $r' \times r'$ flat grid minor Γ' in G', for some $r' = \Omega(\alpha|X'|/\beta)$. Then there exists a polynomial-time algorithm which, given some $G \in \mathcal{F}$ and some α-node-well-linked set X in G, outputs some $r \times r$ flat grid minor Γ in G, for some integer $r = \Omega(\alpha|X|/\beta)$, and a family of λr node-disjoint paths in G connecting X to the interface of Γ, for some constant $0 < \lambda < 1$.*

▶ **Lemma 15.** *Let G be an undirected graph of genus g, and let $\alpha \leq 1$. Let X be an α-node-well-linked set in G. One can, in polynomial time, find some $r \times r$ flat grid minor Γ in G, for some integer $r = \Omega\left(\frac{\alpha|X|}{g^3\sqrt{g}\log^{5/2} n}\right)$, and a family of λr node-disjoint paths connecting X and the interface of Γ, for some $0 < \lambda \leq 1$.*

Proof. This is immediate by combining Lemmas 13 and 14. ◀

Now by Lemmas 2, 5, and 15 we get the following result.

▶ **Theorem 16.** *Let G be a graph of genus g. There is a polynomial time randomized algorithm that, with high probability, achieves an $\Omega\left(\frac{1}{g^3\sqrt{g}\log^{5/2}(n)}\right)$-approximation with congestion 5 for Sym-Dir-NDP instances in G.*

6 Minor Free Graphs

In this section we present the flat grid minor construction for minor-free graphs. We first consider the problem on nearly embeddable graphs, and we extend our solution to arbitrary minor-free graphs by dealing with sums of constant size.

6.1 Nearly Embeddable Graphs

In this subsection we work on nearly embeddable graphs. First we reduce the problem to the case of zero apices.

▶ **Lemma 17** (Reduction to $(0, g, k, p)$-nearly embeddable graphs)**.** *Suppose that there is a polynomial time algorithm for Sym-Dir-NDP in $(0, g, k, p)$-nearly embeddable graphs that achieves a β-approximation with congestion c. Then there is a polynomial time algorithm for Sym-Dir-NDP in (a, g, k, p)-nearly embeddable graphs that achieves a β/a-approximation with congestion c.*

Proof. Let G be an (a, g, k, p)-nearly embeddable graph, and suppose that we are given a Sym-Dir-NDP instance $M = \{s_1t_1, \cdots, s_mt_m\}$ in G. Let $A \subseteq V(G)$ be the set of apices in G. Let $G' = G \setminus A$. Clearly, G' is a $(0, g, k, p)$-nearly embeddable graph. Let $M' \subseteq M$ be the subset of source-terminal pairs that do not intersect A. M' forms a Sym-Dir-NDP instance in G', and thus we can get a β-approximation solution S' with congestion c. Since $|M| \leq |M'| + a$, we have that S' is a β/a-approximation solution with congestion c for M in G, as desired. ◀

Next we provide an algorithm for Sym-Dir-NDP in $(0, g, k, p)$-nearly embeddable graphs. Let G be an $(0, g, k, p)$-nearly embeddable graph, and let S be the bounded genus subgraph of G on the surface; that is, S is obtained from G by deleting all vortices. Let $X \subseteq V(G)$ be the set of terminals. Note that by using Lemma 2 we can reduce the problem to the case where X is α-well-linked for some $\alpha \leq 1$. The following is implicit in [17].

▶ **Lemma 18** (Demaine and Hajiaghayi [17]). *Let $t \geq 1$ be the treewidth of G^{UN}, and let t' be the treewidth of S^{UN}. Then we have $t' \geq \frac{t}{(p+k)^3}$.*

▶ **Lemma 19.** *One can in polynomial time find a $r \times r$ flat grid minor Γ in G^{UN},with $r = \Omega\left(\frac{t}{g^3\sqrt{g}(p+k)^3 \log^{5/2} n}\right)$.*

Proof. By Lemma 18 we have that the treewidth of S^{UN} is at least $\frac{t}{(p+k)^3}$. S^{UN} is a graph of Euler genus g, and thus by Lemma 13 we get the desired result. ◀

▶ **Lemma 20.** *One can in polynomial time find some $r \times r$ flat grid minor Γ in G^{UN}, for some integer $r = \Omega\left(\frac{t}{g^3\sqrt{g}(p+k)^3 \log^{5/2} n}\right)$, and a family of r node-disjoint paths connecting X and the interface of Γ.*

Proof. This is immediate by Lemmas 19 and 14. ◀

Now by combining Lemmas 2, 3, 20, the crossbar construction and routing scheme in Section 4, we get the following result.

▶ **Lemma 21.** *Let G be a $(0, g, k, p)$-nearly embeddable graph. There is a polynomial time randomized algorithm that, with high probability, achieves an $\Omega\left(\frac{1}{g^3\sqrt{g}(p+k)^3 \log^{5/2} n}\right)$-approximation with congestion 5 for Sym-Dir-NDP instances in G.*

▶ **Theorem 22.** *Let G be a (a, g, k, p)-nearly embeddable graph. There is a polynomial time randomized algorithm that, with high probability, achieves an $\Omega\left(\frac{1}{ag^3\sqrt{g}(p+k)^3 \log^{5/2} n}\right)$-approximation with congestion 5 for Sym-Dir-NDP instances in G.*

Proof. This follows immediately by Lemmas 21 and 17. ◀

6.2 Dealing with h-sums

In this subsection we are going to prove Lemma 4. Let G be a minor-free graph, with treewidth t. Let $X \subseteq V(G)$ be the set of terminals. The following is implicit in [18].

▶ **Lemma 23** ([18]). *Let G_1, G_2 be two undirected graphs, and let G_3 be an h-sum of G_1 and G_2 for some integer $h > 0$. Let t_1, t_2, and t_3 be the treewidth of G_1, G_2, and G_3 respectively. Then we have $t_3 \leq \max\{t_1, t_2\}$.*

We are now ready to prove our result for computing flat grid minors in minor-free graphs.

Proof of Lemma 4. By using Theorem 1, we get a decomposition of G^{UN} into h-sums of h-nearly-embeddable graphs. By Lemma 23, we have that at least one summand G' has treewidth at least t. Now G' is a h-nearly-embeddable graph with treewidth t, and thus by Lemma 20 we get the desired flat grid minor. ◀

References

1 Alok Aggarwal, Amotz Bar-Noy, Don Coppersmith, Rajiv Ramaswami, Baruch Schieber, and Madhu Sudan. Efficient Routing and Scheduling Algorithms for Optical Networks. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 412–423, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics.

2 B. Awerbuch, R. Gawlick, T. Leighton, and Y. Rabani. On-line Admission Control and Circuit Routing for High Performance Computing and Communication. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, FOCS '94, pages 412–423, Washington, DC, USA, 1994. IEEE Computer Society.

3 Andrei Z. Broder, Alan M. Frieze, and Eli Upfal. Existence and Construction of Edge-Disjoint Paths on Expander Graphs. *SIAM Journal on Computing*, 23(5):976–989, 1994.

4 Chandra Chekuri and Alina Ene. Poly-logarithmic Approximation for Maximum Node Disjoint Paths with Constant Congestion. In *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 326–341, Philadelphia, PA, USA, 2013. Society for Industrial and Applied Mathematics.

5 Chandra Chekuri and Alina Ene. The all-or-nothing flow problem in directed graphs with symmetric demand pairs. *Mathematical Programming*, 154(1):249–272, December 2015.

6 Chandra Chekuri, Alina Ene, and Marcin Pilipczuk. Constant Congestion Routing of Symmetric Demands in Planar Directed Graphs. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 7:1–7:14, 2016.

7 Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. The All-or-nothing Multicommodity Flow Problem. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 156–165, New York, NY, USA, 2004. ACM.

8 Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. Multicommodity Flow, Well-linked Terminals, and Routing Problems. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 183–192, New York, NY, USA, 2005. ACM.

9 Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory of Computing*, 2:2006, 2006.

10 Chandra Chekuri and Anastasios Sidiropoulos. Approximation algorithms for Euler genus and related problems. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 167–176. IEEE, 2013.

11 Julia Chuzhoy, Venkatesan Guruswami, Sanjeev Khanna, and Kunal Talwar. Hardness of Routing with Congestion in Directed Graphs. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 165–178, New York, NY, USA, 2007. ACM.

12 Julia Chuzhoy and David H. K. Kim. On Approximating Node-Disjoint Paths in Grids. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 187–211, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

13 Julia Chuzhoy, David HK Kim, and Shi Li. Improved approximation for node-disjoint paths in planar graphs. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 556–569. ACM, 2016.

14 Julia Chuzhoy, David HK Kim, and Rachit Nimavat. New hardness results for routing on disjoint paths. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 86–99. ACM, 2017.

15 Julia Chuzhoy and Shi Li. A polylogarithmic approximation algorithm for edge-disjoint paths with congestion 2. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 233–242. IEEE, 2012.

16 Erik D Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 637–646. IEEE, 2005.

17 Erik D Demaine and MohammadTaghi Hajiaghayi. Linearity of grid minors in treewidth with applications through bidimensionality. *Combinatorica*, 28(1):19–36, 2008.

18 Erik D Demaine, MohammadTaghi Hajiaghayi, Naomi Nishimura, Prabhakar Ragde, and Dimitrios M Thilikos. Approximation algorithms for classes of graphs excluding single-crossing graphs as minors. *Journal of Computer and System Sciences*, 69(2):166–195, 2004.

19 Thor Johnson, Neil Robertson, P.D. Seymour, and Robin Thomas. Directed Tree-Width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.

20 R M Karp. On the complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

21 Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Polylogarithmic approximation for minimum planarization (almost). In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 779–788. IEEE, 2017.

22 Stavros G Kolliopoulos and Clifford Stein. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 153–168. Springer, 1998.

23 D. Peleg and E. Upfal. Constructing disjoint paths on expander graphs. *Combinatorica*, 9(3):289–313, September 1989.

24 Prabhakar Raghavan and Eli Upfal. Efficient Routing in All-optical Networks. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 134–143, New York, NY, USA, 1994. ACM.

25 B. Reed. Introducing Directed Tree Width. *Electronic Notes in Discrete Mathematics*, 3(Supplement C):222–229, 1999. 6th Twente Workshop on Graphs and Combinatorial Optimization.

26 N. Robertson and P.D. Seymour. Graph Minors. XIII. The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.

27 Neil Robertson and Paul D Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003.

28 Carsten Thomassen. A simpler proof of the excluded minor theorem for higher surfaces. *Journal of Combinatorial Theory, Series B*, 70(2):306–311, 1997.

A Missing Proofs

Proof of Lemma 14. Let t be the treewidth of G. Since X is α-node-well-linked in G, we have that $t = \Omega(\alpha|X|)$. Let Γ_0 be an $r' \times r'$ flat grid minor in G, for some $r' = \Omega(\alpha|X|/\beta)$. If there is a family of λr_0 node-disjoint paths connecting X and the the interface of Γ_0, then we are done. Otherwise, we will find an *irrelevant* vertex; that is a vertex $v \in V(G)$ such that deleting v from G does not affect the well-linkedness of X. Therefore, we can delete v from G, and recursively call the process for finding flat grid minors, until we get the desired one.

Suppose that there is not a family of λr_0 node-disjoint paths connecting X and the interface of Γ_0. First we find a $r'_0 \times r'_0$ sub-grid Γ'_0 of Γ_0 such that $r'_0 = O(r_0)$ and Γ'_0 contains at most $\frac{\lambda r_0}{\alpha}$ terminals. For any minor H of G, and for every $v \in V(H)$, let $\eta(v) \subseteq V(G)$ be the subset of vertices in G corresponding to v. Let also $X_H = X \cap \eta(H)$. Since there is not a family of λr_0 node-disjoint paths connecting X and the interface of Γ_0, we can find a cut $C \subseteq E(G)$ in G, separating X_{Γ_0} and the interface of Γ_0, with $|C| < \lambda r_0$. Now let $A_1, A_2, \ldots, A_m$ be the connected components of $G \setminus C$ that contain vertices of X_{Γ_0} (See Figure 4). We may assume w.l.o.g. that $|V(A_1)| \geq |V(A_2)| \geq \ldots \geq |V(A_m)|$. Now let $Y, Z \subset X$ be two disjoint subsets of X of equal size such that $X_{A_1} \subset Y$ and $X_{A_i} \subset Z$ for any $i \in \{2, 3, \ldots, m\}$. Since X is α-node-well-linked, there exist a family $\mathcal{P}$ of $|Y|$ paths from Y

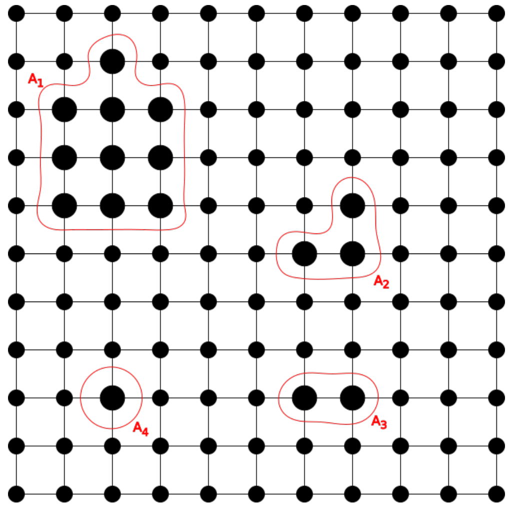

Figure 4 The connected components of $G \setminus C$ in Γ_0'.

to Z such that no vertex is in more than $1/\alpha$ of these paths. However, we have $X_{A_1} \subset Y$ and $X_{A_i} \subset Z$ for any $i \in \{2, 3, \ldots, m\}$, and thus we have $|V(X_{A_2}) \cup \ldots \cup V(X_{A_m})| \leq |C|\frac{1}{\alpha} < \frac{\lambda r_0}{\alpha}$. Therefore, we can find a $\frac{r_0}{4} \times \frac{r_0}{4}$ sub-grid Γ_0' of Γ_0 such that Γ_0' does not intersect X_{A_1}, and moreover there are at most $\frac{\lambda r_0}{\alpha}$ number of terminals in $\eta(\Gamma_0')$.

If there is a family of $\lambda r_0'$ node-disjoint paths connecting X and the interface of Γ_0', then we are done. Otherwise, we find an irrelevant vertex. We use a similar technique as in [21]. Let Γ_0'' be the $r_0'' \times r_0''$ sub-grid of Γ_0' obtained by deleting the first and last $r_0'/4$ rows and columns of Γ_0'. By the construction, we know that Γ_0'' contains at most $\frac{\lambda r_0}{\alpha}$ terminals. We may assume w.l.o.g. that r_0' is a power of 2, and thus r_0'' is a power of 2 as well. We construct a hierarchical partitioning of Γ_0'' into smaller sub-grids as follows. For every $i, j \in \{1, 2, \ldots, r_0''\}$, let $v_{i,j}$ be the vertex in the i'th row and j'th column of Γ_0''. For any $i, j, h \in \{1, 2, \ldots, r_0''\}$, let

$$H_{i,j,h} = \bigcup_{a=\max\{1,i-h-1\}}^{\min\{i+h,r_0''\}} \bigcup_{b=\max\{1,j-h-1\}}^{\min\{j+h,r_0''\}} \{v_{a,b}\}.$$

We also define $\ell(H_{i,j,h}) = 2\text{h}$. For every $q \in \{0, 1, \ldots, \log r_0''\}$, we define two partitions of Γ_0'' into $q \times q$ sub-grids as follows. Let

$$\mathcal{H}_{q,1} = \bigcup_{i=0}^{r_0''/2^{q+1}} \bigcup_{j=0}^{r_0''/2^{q+1}} \{H(i2^{q+1}, j2^{q+1}, 2^q)\},$$

and

$$\mathcal{H}_{q,2} = \bigcup_{i=0}^{r_0''/2^{q+1}} \bigcup_{j=0}^{r_0''/2^{q+1}} \{H(i2^{q+1} + 2^q, j2^{q+1} + 2^q, 2^q)\}.$$

Let $\mathcal{H} = \bigcup_{q=0}^{\log r_0''} \bigcup_{i=1}^{2} \mathcal{H}_{q,i}$. For every $H \in \mathcal{H}$, let $w(H)$ be the number of terminals in $\eta(H)$. Let also $w(\Gamma_0'')$ be the number of terminals in $\eta(\Gamma_0'')$. We say that some $H \in \mathcal{H}$ is *dense* if $w(H) \geq \ell(H)/100$. Let $\delta(\Gamma_0'')$ be the interface of Γ_0''. We say that some $v \in V(\Gamma_0'')$ is *good* if v is not contained in any dense $H \in \mathcal{H}$, and there is no terminals in $\eta(v)$. First we

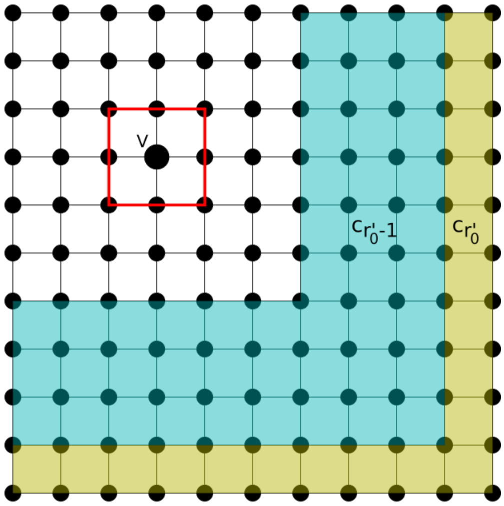

Figure 5 Sets C_q.

show that there exists a good vertex in Γ''_0. We count the number of vertices in Γ''_0 that are contained in at least one dense $H \in \mathcal{H}$. Let $\mathcal{H}_{q,j} \in \mathcal{H}$ for some $q \in \{0,\ldots,\log r''_0\}$ and $j \in \{1,2\}$, and let $H \in \mathcal{H}_{q,j}$. H is dense if and only if $w(H) \geq \ell(H)/100 = 2^{q+1}/100$. We know that $w(\Gamma''_0) \leq r''_0/10000$, and thus if $2^{q+1} > r''_0/100$, then there are no dense $H \in \mathcal{H}_{q,j}$. Now suppose that $2^{q+1} \leq r''_0/100$, and thus $q < \log r''_0 - 7$. Let $i \in \{8,\ldots,\log r''_0\}$, and let $q = \log r''_0 - i$. Let $H' \in \mathcal{H}_{q,1}$. We have that $\ell(H') = 2^{q+1} = r''_0/2^{i-1}$. In order for H' to be dense it must be that $w(H') \geq \frac{\ell(H)}{100} = \frac{r''_0}{100\cdot 2^{i-1}}$. Note that we have $w(\Gamma''_0) \leq r''_0/10000$, and therefore there can be at most $2^{i-1}/100$ dense $H' \in \mathcal{H}_{q,1}$. With a similar argument, we can show that there can be at most $2^{i-1}/100$ dense $H' \in \mathcal{H}_{q,2}$. Now we have

$$\left|\bigcup_{H\in\mathcal{H}:H \text{ is dense}} H\right| \leq 2\cdot\sum_{i=8}^{\log r''_0}\left(\frac{r''_0}{2^{i-1}}\right)^2\cdot\frac{2^{i-1}}{100}$$
$$= \frac{(r''_0)^2}{50}\cdot\sum_{i=8}^{\log r''_0}\frac{1}{2^{i-1}}$$
$$< \frac{(r''_0)^2}{50}.$$

This means that there exist at least $\frac{49(r''_0)^2}{50}$ vertices in Γ''_0 that are not contained in any dense $H \in \mathcal{H}$, and since there are at most $r''_0/10000$ terminals in $\eta(\Gamma''_0)$, there must exist a good vertex in Γ''_0, as desired. Furthermore, this vertex can be found in polynomial time. Let $v \in V(\Gamma''_0)$ be a good vertex.

We claim that vertices in $\eta(v)$ are irrelevant. For every $q \in \{0,1,\ldots,\log r''_0\}$ and $i \in \{1,2\}$, let $H_{q,i} \in \mathcal{H}_{q,i}$ be a sub-grid that contains v. By the construction, for every $q \in \{0,1,\ldots,\log r''_0\}$, we have that either $d_{\Gamma''_0}(v,\delta(H_{q,1})) \geq 2^{q-1}$ or $d_{\Gamma''_0}(v,\delta(H_{q,2})) \geq 2^{q-1}$. Let $B_q \in \{H_{q,1},H_{q,2}\}$ be such that $d_{\Gamma''_0}(v,\delta(B_q)) \geq 2^{q-1}$. For every $q \in \{1,\ldots,\log r''_0\}$, let $C_q = B_q \setminus B_{q-1}$, and let also $C_{\log r''_0+1} = V(\Gamma'_0) \setminus V(\Gamma''_0)$ (See Figure 5).

Let $Y, Z \subset X$ be two disjoint subsets of X of equal size. Since X is α-node well-linked, we know that there exists a family $\mathcal{P}$ of $|Y|$ paths from Y to Z such that no vertex is in more than $1/\alpha$ of these paths. If none of these paths use v, then we are done. Otherwise, we try to re-route these paths to obtain a new family $\mathcal{P}'$ of paths, such that no path is using

v, and no vertex is in more than $1/\alpha$ of the paths in $\mathcal{P}'$. First we look at the paths $P \in \mathcal{P}$ with both endpoints outside of Γ_0'; that is the endpoints of P do not belong to $\eta(\Gamma_0')$. Let $\mathcal{P}^* \subseteq \mathcal{P}$ be the set of all such paths. We re-route them in a way such that they do not intersect $\eta(\Gamma_0'')$. Note that by the construction, at most $\lambda r_0'$ of paths in $\mathcal{P}^*$ can intersect $\eta(\Gamma_0')$. For these paths, we can re-route their intersection with $\eta(\Gamma_0')$ in $\eta(\Gamma_0') \setminus \eta(\Gamma_0'')$, and thus they will not intersect $\eta(\Gamma_0'')$. Now let $\mathcal{P}^{**} \subseteq \mathcal{P}$ be the set of paths with one endpoint outside of $\eta(\Gamma_0')$, and one endpoint inside of $\eta(\Gamma_0')$. Let $P = (a_1, a_2, \ldots, a_p) \in \mathcal{P}^{**}$, where $a_1 \notin \eta(\Gamma_0')$ and $a_p \in \eta(\Gamma_0')$. Let $a_f \in V(P)$ be the first intersection of P and $\eta(\Gamma_0')$; that is $f \in \{1, 2, \ldots, p\}$ is the minimum number such that $a_f \in \eta(\Gamma_0')$. Let $P' = (a_f, \ldots, a_p)$. We replace P with P' in $\mathcal{P}$. Note that again there are at most $\lambda r_0'$ such paths in $\mathcal{P}$. Now we are only dealing with paths with both endpoints in $\eta(\Gamma_0')$. For all such paths, we use an inductive argument to re-route them. For any $i, j \in \{1, 2, \ldots, \log r_0'' + 1\}$, let $\mathcal{P}_{i,j} \subseteq \mathcal{P}$ be the paths with one endpoint in $\eta(C_i)$, and the other endpoint in $\eta(C_j)$. By the construction, for any $i \in \{1, 2, \ldots, \log r_0'' + 1\}$, we know that there are at most $2^i/20$ terminals in $\eta(C_i)$, and thus $|\mathcal{P}_{i,i}| \le 2^i/20$. For all such paths, we can re-route them such that they stay inside C_i. We start with $\mathcal{P}_{\log r_0''+1, \log r_0''+1}$, and re-route all these paths such that they only use vertices in $C_{\log r_0''+1}$. Again, by the construction, we have that $\left| \bigcup_{j=1}^{\log r_0''} \mathcal{P}_{\log r_0''+1, j} \right| \le r_0''/10$. For all $P \in \bigcup_{j=1}^{\log r_0''} \mathcal{P}_{\log r_0''+1, j}$, similar to the paths in $\mathcal{P}^{**}$, we can replace them with paths with one endpoint on the boundary of $C_{\log r_0''}$, and recursively follow the same argument for paths with both endpoints in $\eta\left(\bigcup_{j=1}^{\log r_0''} C_j\right)$ and so on. Therefore, by applying the same re-routing pattern, we can get a new set of paths $\mathcal{P}'$ such that no path uses vertex v, as desired.

Now let $G_1 = G \setminus v$. Since v is an irrelevant vertex in G, we have that X is α-node-well-linked in G_1, and thus we have that the treewidth of G_1 is $\Omega(\alpha|X|)$. Therefore, we can find a $r_1' \times r_1'$ flat grid minor Γ_1 in G_1, for some $r_1' = \Omega\left(\frac{\alpha|X|}{\beta}\right)$. If there exists a family of $\lambda r_1'$ node-disjoint paths connecting X and the interface of Γ_1, we are done. Otherwise, we recursively follow the same approach to find an irrelevant vertex v_1 in G_1, and let $G_2 = G_1 \setminus v_1$ and so on. This recursive call stops in $O(n)$ steps, because for each $i \ge 1$, G_i is a graph of treewidth $\alpha|X|$. Therefore, for some $j \ge 1$, we can find a $r_j \times r_j$ flat grid minor Γ_j of G_j, for some $r_j = \Omega\left(\frac{\alpha|X|}{\beta}\right)$, such that there exists a family of λr_j node-disjoint paths connecting X and the interface of Γ_j. Note that Γ_j is also a flat grid minor of G, and this completes the proof. ◀