Improved Approximation for Node-Disjoint Paths in Planar Graphs*

Julia Chuzhoy[†] David H. K. Kim[‡] Shi Li[§]

March 17, 2016

Abstract

We study the classical Node-Disjoint Paths (NDP) problem: given an n-vertex graph G and a collection $\mathcal{M} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of pairs of vertices of G called *demand pairs*, find a maximum-cardinality set of node-disjoint paths connecting the demand pairs. NDP is one of the most basic routing problems, that has been studied extensively. Despite this, there are still wide gaps in our understanding of its approximability: the best currently known upper bound of $O(\sqrt{n})$ on its approximation ratio is achieved via a simple greedy algorithm, while the best current negative result shows that the problem does not have a better than $\Omega(\log^{1/2-\delta} n)$-approximation for any constant δ, under standard complexity assumptions. Even for planar graphs no better approximation algorithms are known, and to the best of our knowledge, the best negative bound is APX-hardness. Perhaps the biggest obstacle to obtaining better approximation algorithms for NDP is that most currently known approximation algorithms for this type of problems rely on the standard multi-commodity flow relaxation, whose integrality gap is $\Omega(\sqrt{n})$ for NDP, even in planar graphs. In this paper, we break the barrier of $O(\sqrt{n})$ on the approximability of the NDP problem in planar graphs and obtain an $\tilde{O}(n^{9/19})$-approximation. We introduce a new linear programming relaxation of the problem, and a number of new techniques, that we hope will be helpful in designing more powerful algorithms for this and related problems.

*An extended abstract is to appear in STOC 2016

[†]Toyota Technological Institute at Chicago. Email: cjulia@ttic.edu. Supported in part by NSF grant CCF-1318242.

[‡]Computer Science Department, University of Chicago. Email: hongk@cs.uchicago.edu. Supported in part by NSF grant CCF-1318242.

[§]Department of Computer Science and Engineering, University at Buffalo. Part of the work done while the author was at the Toyota Technological Institute at Chicago. Email: shil@buffalo.edu.

1 Introduction

In the Node-Disjoint Paths (NDP) problem, we are given an n-vertex graph G, and a collection $\mathcal{M} = \{(s_1,t_1),\ldots,(s_k,t_k)\}$ of pairs of vertices of G, called *source-destination*, or *demand*, pairs. The goal is to route as many of the demand pairs as possible, by connecting each routed pair with a path, so that the resulting paths are node-disjoint. We denote by NDP-Planar the special case of the problem where the input graph G is planar, and by NDP-Grid the special case where G is the $(\sqrt{n} \times \sqrt{n})$-grid. NDP is one of the most basic problems in the area of graph routing, and it was initially introduced to the area in the context of VLSI design. In addition to being extensively studied in the area of approximation algorithms, this problem has played a central role in Robertson and Seymour's Graph Minor series. When the number of the demand pairs k is bounded by a constant, Robertson and Seymour [RS90, RS95] have shown an efficient algorithm for the problem, as part of the series. When k is a part of input, the problem becomes NP-hard [Kar72, EIS76], even on planar graphs [Lyn75], and even on grid graphs [KvL84]. Despite the importance of this problem, and many efforts, its approximability is still poorly understood. The following simple greedy algorithm achieves an $O(\sqrt{n})$-approximation [KS04]: while G contains any path connecting any demand pair, choose the shortest such path P, add P to the solution, and delete all vertices of P from G. Surprisingly, this elementary algorithm is the best currently known approximation algorithm for NDP, even for planar graphs. Until recently, this was also the best approximation algorithm for NDP-Grid. On the negative side, it is known that there is no $O(\log^{1/2-\delta} n)$-approximation algorithm for NDP for any constant δ, unless $\mathsf{NP} \subseteq \mathsf{ZPTIME}(n^{\operatorname{poly}\log n})$ [AZ05, ACG+10]. To the best of our knowledge, the best negative result for NDP-Planar and for NDP-Grid is APX-hardness [CK15]. Perhaps the biggest obstacle to obtaining better upper bounds on the approximability of NDP is that the common approach to designing approximation algorithms for this type of problems is to use the multicommodity flow relaxation, where instead of connecting the demand pairs with paths, we send a (possibly fractional) multicommodity flow between them. The integrality gap of this relaxation is known to be $\Omega(\sqrt{n})$, even for planar graphs, and even for grid graphs. In a recent work, Chuzhoy and Kim [CK15] showed an $\tilde{O}(n^{1/4})$-approximation algorithm for NDP-Grid, thus bypassing the integrality gap obstacle for this restricted family of graphs. The main result of this paper is an $\tilde{O}(n^{9/19})$-approximation algorithm for NDP-Planar. We also show that, if the value of the optimal solution to the NDP-Planar instance is OPT, then we can efficiently route $\Omega\left(\frac{\mathsf{OPT}^{1/19}}{\operatorname{poly}\log n}\right)$ demand pairs. Our algorithm is motivated by the work of [CK15] on NDP-Grid, and it relies on approximation algorithms for the NDP problem on a disc and on a cylinder, that we discuss next.

We start with the NDP problem on a disc, that we denote by NDP-Disc. In this problem, we are given a planar graph G, together with a set $\mathcal{M}$ of demand pairs as before, but we now assume that G can be drawn in a disc, so that all vertices participating in the demand pairs lie on its boundary. The NDP problem on a cylinder, NDP-Cylinder, is defined similarly, except that now we assume that we are given a cylinder Σ, obtained from the sphere, by removing two disjoint open discs (caps) from it. We denote the boundaries of the discs by Γ_1 and Γ_2 respectively, and we call them the *cuffs* of Σ. We assume that G can be drawn on Σ, so that all source vertices participating in the demand pairs in $\mathcal{M}$ lie on Γ_1, and all destination vertices lie on Γ_2. Robertson and Seymour [RS86] showed an algorithm, that, given an instance of the NDP-Disc or the NDP-Cylinder problem, decides whether all demand pairs in $\mathcal{M}$ can be routed simultaneously via node-disjoint paths, and if so, finds the routing efficiently. Moreover, for each of the two problems, they give an exact characterization of instances for which all pairs in $\mathcal{M}$ can be routed in G. Several other very efficient algorithms are known for both problems [RLWW96, SAN90]. However, for our purposes, we need to consider the optimization version of both problems, where we are no longer guaranteed that all demand pairs in $\mathcal{M}$ can be routed, and would like to route the largest possible subset of the demand pairs. We are not aware of any results for these two special cases of the NDP problem. In this paper, we provide $O(\log k)$-approximation algorithms for both problems.

Other Related Work. A problem closely related to NDP is the Edge-Disjoint Paths (EDP) problem. It is defined similarly, except that now the paths chosen to the solution are allowed to share vertices, and are only required to be edge-disjoint. It is easy to show, by using a line graph of the EDP instance, that NDP is more general than EDP (though this transformation inflates the number of the graph vertices, so it may not preserve approximation factors that depend on n). This relationship breaks down in planar graphs, since the resulting NDP instance may no longer be planar. The approximability status of EDP is very similar to that of NDP: there is an $O(\sqrt{n})$-approximation algorithm [CKS06], and it is known that there is no $O(\log^{1/2-\delta} n)$-approximation algorithm for any constant δ, unless $\mathsf{NP} \subseteq \mathsf{ZPTIME}(n^{\text{poly}\log n})$ [AZ05, ACG+10]. We do not know whether our techniques can be used to obtain improved approximation algorithms for EDP on planar graphs. As in the NDP problem, we can use the standard multicommodity flow LP-relaxation of the problem, in order to obtain an $O(\sqrt{n})$-approximation algorithm, and the integrality gap of the LP-relaxation is $\Omega(\sqrt{n})$ even in planar graphs. For several special cases of the problem better algorithms are known: Kleinberg [Kle05], building on the work of Chekuri, Khanna and Shepherd [CKS05, CKS04], has shown an $O(\log^2 n)$-approximation LP-rounding algorithm for even-degree planar graphs. Aumann and Rabani [AR95] showed an $O(\log^2 n)$-approximation algorithm for EDP on grid graphs, and Kleinberg and Tardos [KT98, KT95] showed $O(\log n)$-approximation algorithms for broader classes of nearly-Eulerian uniformly high-diameter planar graphs, and nearly-Eulerian densely embedded graphs. Recently, Kawarabayashi and Kobayashi [KK13] gave an $O(\log n)$-approximation algorithm for EDP when the input graph is either 4-edge-connected planar or Eulerian planar. It appears that the restriction of the graph G to be Eulerian, or nearly-Eulerian, makes the EDP problem significantly simpler, and in particular improves the integrality gap of the LP-relaxation. The analogue of the grid graph for the EDP problem is the wall graph: the integrality gap of the standard LP-relaxation for EDP on wall graphs is $\Omega(\sqrt{n})$, and until recently, no better than $O(\sqrt{n})$-approximation algorithms for EDP on walls were known. The work of [CK15] gives an $\tilde{O}(n^{1/4})$-approximation algorithm for EDP on wall graphs.

A variation of the NDP and EDP problems, where small congestion is allowed, has been a subject of extensive study. In the NDP with congestion (NDPwC) problem, the input is the same as in the NDP problem, and we are additionally given a non-negative integer c. The goal is to route as many of the demand pairs as possible with congestion at most c: that is, every vertex may participate in at most c paths in the solution. EDP with Congestion (EDPwC) is defined similarly, except that now the congestion bound is imposed on edges and not vertices. The classical randomized rounding technique of Raghavan and Thompson [RT87] gives a constant-factor approximation for both problems, if the congestion c is allowed to be as high as $\Theta(\log n/\log\log n)$. A recent line of work [CKS05, Räc02, And10, RZ10, Chu12, CL12, CE13, CC] has lead to an $O(\text{poly}\log k)$-approximation for both NDPwC and EDPwC problems, with congestion $c = 2$. For planar graphs, a constant-factor approximation with congestion 2 is known for EDP [SCS11]. All these algorithms perform LP-rounding of the standard multicommodity flow LP-relaxation of the problem and so it is unlikely that they can be extended to routing with no congestion.

Our Results and Techniques. Given an instance $(G, \mathcal{M})$ of the NDP problem, we denote by $\mathsf{OPT}(G, \mathcal{M})$ the value of the optimal solution to it. Our first result is an approximation algorithm for NDP-Disc and NDP-Cylinder.

Theorem 1.1 *There is an efficient $O(\log k)$-approximation algorithm for the* NDP-Disc *and the* NDP-Cylinder *problems, where k is the number of the demand pairs in the instance.*

We provide a brief high-level overview of the techniques we use in the proof of Theorem 1.1. We define a new intermediate problem, called Demand Pair Selection Problem (DPSP). In this problem, we are given two disjoint directed paths σ and σ', and a set $\mathcal{M} = \{(s_1,t_1),\ldots,(s_k,t_k)\}$ of pairs of vertices that we call demand pairs, such that all vertices in $\{s_1,\ldots,s_k\}$ lie on σ, and all vertices in $\{t_1,\ldots,t_k\}$ lie on σ'. For any pair of vertices $v,v' \in V(\sigma)$, we denote $v \prec v'$ if v lies before v' on σ, and we denote $v \preceq v'$ if $v = v'$ or $v \prec v'$. For every pair $v,v' \in V(\sigma')$ of vertices, we define the relationships $v \prec v'$ and $v \preceq v'$ similarly. We say that two demand pairs $(s,t),(s',t') \in \mathcal{M}$ *cross*, if either (i) $\{s,t\} \cap \{s',t'\} \neq \emptyset$, or (ii) $s \prec s'$ and $t' \prec t$, or (iii) $s' \prec s$ and $t \prec t'$. We are also given a set $\mathcal{K}$ of constraints that we describe below. The goal of the DPSP problem is to find a maximum-cardinality subset $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs, such that no two pairs in $\mathcal{M}'$ cross, and all constraints in $\mathcal{K}$ are satisfied. There are four types of constraints in $\mathcal{K}$, and each constraint is defined by a quadruple (i,x,y,w), where $i \in \{1,2,3,4\}$ is the constraint type, $x,y \in V(\sigma) \cup V(\sigma')$ are vertices, and w is an integer. In a type-1 constraint $(1,x,y,w)$, we have $x,y \in V(\sigma)$, and the constraint requires that the number of the demand pairs $(s,t) \in \mathcal{M}'$ with $x \preceq s \preceq y$ is at most w. Similarly, in a type-2 constraint $(2,x,y,w)$, we have $x,y \in V(\sigma')$, and the constraint requires that the number of the demand pairs $(s,t) \in \mathcal{M}'$ with $x \preceq t \preceq y$ is at most w. If $(3,x,y,w) \in \mathcal{K}$ is a type-3 constraint, then $x \in V(\sigma)$ and $y \in V(\sigma')$ must hold. The constraint requires that the number of the demand pairs $(s,t) \in \mathcal{M}'$ with $s \preceq x$ and $y \preceq t$ is at most w. Similarly, if $(4,x,y,w) \in \mathcal{K}$ is a type-4 constraint, then $x \in V(\sigma)$ and $y \in V(\sigma')$ must hold, and the constraint requires that the number of the demand pairs $(s,t) \in \mathcal{M}'$ with $x \preceq s$ and $t \preceq y$ is at most w. We show that both NDP-Disc and NDP-Cylinder reduce to DPSP with an $O(\log k)$ loss in the approximation factor. The reduction from NDP-Disc to DPSP uses the characterization of routable instances of NDP-Disc due to Robertson and Seymour [RS86]. Finally, we show a factor-8 approximation algorithm for the DPSP problem.

The main result of our paper is summarized in the following two theorems.

Theorem 1.2 *There is an efficient $O(n^{9/19} \cdot \text{poly}\log n)$-approximation algorithm for the* NDP-Planar *problem.*

Theorem 1.3 *There is an efficient algorithm, that, given an instance $(G,\mathcal{M})$ of* NDP-Planar*, computes a routing of $\Omega\left(\frac{(\mathsf{OPT}(G,\mathcal{M}))^{1/19}}{\text{poly}\log n}\right)$ demand pairs of $\mathcal{M}$ via node-disjoint paths in G.*

Notice that when $\mathsf{OPT}(G,\mathcal{M})$ is small, Theorem 1.3 gives a much better that $\tilde{O}(n^{9/19})$-approximation.

We now give a high-level intuitive overview of the proof of Theorem 1.2. Given an instance $(G,\mathcal{M})$ of the NDP problem, we denote by $\mathcal{T}$ the set of vertices participating in the demand pairs in $\mathcal{M}$, and we refer to them as *terminals*. We start with a quick overview of the $\tilde{O}(n^{1/4})$-approximation algorithm of [CK15] for the NDP-Grid problem, since their algorithm was the motivation for this work. The main observation of [CK15] is that the instances of NDP-Grid, for which the multicommodity flow relaxation exhibits the $\Omega(\sqrt{n})$ integrality gap, have terminals close to the grid boundary. When all terminals are at a distance of at least $\Omega(n^{1/4})$ from the boundary of the grid, one can find an $\tilde{O}(n^{1/4})$-approximation via LP-rounding (but unfortunately the integrality gap remains polynomial in n even in this case). When the terminals are close to the grid boundary, the integrality gap of the LP-relaxation becomes $\Omega(\sqrt{n})$. However, this special case of NDP-Grid can be easily approximated via simple dynamic programming. For example, when all terminals lie on the grid boundary, the integrality gap of the LP-relaxation is $\Omega(\sqrt{n})$, but a constant-factor approximation can be achieved via standard dynamic programming. More generally, when all terminals are within distance at most $O(n^{1/4})$ from the grid boundary, we can obtain an $O(n^{1/4})$-approximation via dynamic programming. Overall, we partition the demand pairs of $\mathcal{M}$ into two subsets, depending on whether the terminals lie close to or far from the grid boundary, and obtain an $\tilde{O}(n^{1/4})$-approximation for each of the two resulting problem instances separately, selecting the better of the two solutions as our output.

This idea is much more difficult to implement in general planar graphs. For one thing, the notion of the "boundary" of a planar graph is meaningless - any face in the drawing of the planar graph can be chosen as the outer face. We note that the standard multicommodity flow LP-relaxation performs poorly not only when all terminals are close to the boundary of a single face (a case somewhat similar to NDP-Disc), but also when there are two faces F and F', and for every demand pair $(s,t) \in \mathcal{M}$, s is close to the boundary of F and t is close to the boundary of F' (this setting is somewhat similar to NDP-Cylinder). The notion of "distance", when deciding whether the terminals lie close to or far from a face boundary is also not well-defined, since we can subdivide edges and artificially modify the graph in various ways in order to manipulate the distances without significantly affecting routings. Intuitively, we would like to define the distances between the terminals in such a way that, on the one hand, whenever we find a set $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs, such that all terminals participating in the pairs in $\mathcal{M}'$ are far enough from each other, then we can route a large subset of the demand pairs in $\mathcal{M}'$. On the other hand, if we find a set $\mathcal{M}'' \subseteq \mathcal{M}$ of demand pairs, with all terminals participating in the pairs in $\mathcal{M}''$ being close to the boundary of some face (or a pair of faces), then we can find a good approximate solution to instance $(G, \mathcal{M}'')$ (for example, by reducing the problem to NDP-Disc or NDP-Cylinder). Since we do not know beforehand which face (or faces) will be chosen as the "boundary" of the graph, we cannot partition the problem into two sub-problems and employ different techniques to solve each sub-problem as we did for NDP-Grid. Instead, we need a single framework in which both cases can be handled.

We assume that every terminal participates in exactly one demand pair, and that the degree of every terminal is 1. This can be achieved via a standard transformation, where we create several copies of each terminal, and connect them to the original terminal. This transformation may introduce up to $O(n^2)$ new vertices. Since we are interested in obtaining an $\tilde{O}(n^{9/19})$-approximation for NDP-Planar, we denote by N the number of the non-terminal vertices in the new graph G. Abusing the notation, we denote the total number of vertices in the new problem instance by n. It is now enough to obtain an $\tilde{O}(N^{9/19})$-approximation for the new problem instance.

Our first step is to define a new LP-relaxation of the problem. We assume that we have guessed correctly the value OPT of the optimal solution. We start with the standard multicommodity flow LP-relaxation, where we try to send OPT flow units between the demand pairs, so that the maximum amount of flow through any vertex is bounded by 1. We then add the following new set of constraints to the LP: for every subset $\mathcal{M}' \subseteq \mathcal{M}$ of the demand pairs, for every value $\mathsf{OPT}(G, \mathcal{M}') \leq z \leq k$, the total amount of flow routed between the demand pairs in $\mathcal{M}'$ is no more than z. Adding this type of constraints may seem counter-intuitive. We effectively require that the LP solves the problem exactly, and naturally we cannot expect to be able to do so efficiently. Since the number of the resulting constraints is exponential in k, and since we do not know the values $\mathsf{OPT}(G, \mathcal{M}')$, we indeed cannot solve this LP efficiently. In fact, our algorithm does not attempt to solve the LP exactly. Instead, we employ the Ellipsoid algorithm, that in every iteration produces a potential solution to the LP-relaxation. We then show an algorithm that, given such a potential solution, either finds an integral solution routing $\tilde{\Omega}(\mathsf{OPT}/N^{9/19})$ demand pairs, or it returns some subset $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs, whose corresponding LP-constraint is violated. Therefore, we use our approximation algorithm as the separation oracle for the Ellipsoid algorithm. We are then guaranteed that after $\text{poly}(n)$ iterations, we will obtain a solution routing the desired number of demand pairs, as only $\text{poly}(n)$ iterations are required for the Ellipsoid algorithm in order to find a feasible LP-solution.

The heart of the proof of Theorem 1.2 is then an algorithm that, given a potential (possibly infeasible) solution to the LP-relaxation, either finds an integral solution routing $\tilde{\Omega}(\mathsf{OPT}/N^{9/19})$ demand pairs, or returns some subset $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs, whose corresponding LP-constraint is violated. We can assume without loss of generality that the fractional solution we are given satisfies all the standard multicommodity flow constraints, as this can be verified efficiently. For simplicity of exposition, we assume that every demand pair in $\mathcal{M}$ sends the same amount of w^* flow units to each other.

We assume for now that the set $\mathcal{T}$ of terminals is α_{WL}-well-linked, for $\alpha_{\text{WL}} = \Theta(w^*/\log n)$ - that is, for every pair $(\mathcal{T}', \mathcal{T}'')$ of disjoint equal-sized subsets of vertices of $\mathcal{T}$, we can connect vertices of $\mathcal{T}'$ to vertices of $\mathcal{T}''$ by at least $\alpha_{\text{WL}} \cdot |\mathcal{T}'|$ node-disjoint paths. We discuss this assumption in more detail below. We assume that we are given a drawing of G on the sphere. Our first step is to define the notion of distances between the terminals. In order to do so, we first construct *enclosures* around them. Throughout the proof, we use a parameter $\Delta = \mathsf{OPT}^{2/19}$. We say that a curve γ on the sphere is a *G-normal curve* iff it intersects the drawing of G only at its vertices. The length of such a curve is the number of vertices of G it contains. An enclosure around a terminal t is a disc D_t containing t, whose boundary, that we denote by C_t, is a G-normal curve of length exactly Δ, so that at most $O(\Delta/\alpha_{\text{WL}})$ terminals lie in D_t. We show an efficient algorithm to construct the enclosures D_t around the terminals t, so that the following additional conditions hold: (i) if $D_t \subseteq D_{t'}$ for any pair $t, t' \in \mathcal{T}$ of terminals, then $D_t = D_{t'}$; and (ii) if $D_t \cap D_{t'} = \emptyset$, then there are Δ node-disjoint paths connecting the vertices of C_t to the vertices of $C_{t'}$. We now define the distances between pairs of terminals. For every pair $(t, t') \in \mathcal{T}$, distance $d(t, t')$ is the length of the shortest G-normal curve, connecting a vertex of C_t to a vertex of $C_{t'}$.

Next, we show that one of the following has to happen: either there is a large collection $\tilde{\mathcal{M}} \subseteq \mathcal{M}$ of demand pairs, such that all terminals participating in the pairs in $\tilde{\mathcal{M}}$ are at a distance at least $\Omega(\Delta)$ from each other; or there is a large collection $\tilde{\mathcal{M}}' \subseteq \mathcal{M}$ of demand pairs, and two faces F, F' in the drawing of G (with possibly $F = F'$), such that for every demand pair in $\tilde{\mathcal{M}}'$, one of its terminals is within distance at most $\tilde{O}(\Delta)$ from the boundary of F, and the other is within distance at most $\tilde{O}(\Delta)$ from the boundary of F'. In the former case, we show that we can route a large subset of the demand pairs in $\tilde{\mathcal{M}}$ via node-disjoint paths, by constructing a special routing structure called a crossbar (this construction exploits well-linkedness of the terminals and the paths connecting the enclosures). In the latter case, we reduce the problem to $\mathsf{NDP\text{-}Disc}$ or $\mathsf{NDP\text{-}Cylinder}$, depending on the distance between the faces F and F', and employ the approximation algorithms for these problems to route $\tilde{\Omega}\left(\frac{\mathsf{OPT}(G,\tilde{\mathcal{M}}')}{\text{poly}(\Delta)}\right)$ demand pairs from $\tilde{\mathcal{M}}'$ in G. If the resulting number of demand pairs routed is close enough to OPT, then we return this as our final solution. Otherwise, we show that the LP-constraint corresponding to the set $\tilde{\mathcal{M}}'$ of demand pairs is violated, or equivalently, the amount of flow sent by the LP solution between the demand pairs in $\tilde{\mathcal{M}}'$ is greater than $\mathsf{OPT}(G, \tilde{\mathcal{M}}')$.

So far we have assumed that the terminals participating in the demand pairs in $\mathcal{M}$ are α_{WL}-well-linked. In general this may not be the case. Using standard techniques, we can perform a well-linked decomposition: that is, compute a subset $U \subseteq V(G)$ of at most $\mathsf{OPT}/64$ vertices, such that, if we denote the set of all connected components of $G \setminus U$ by $\{G_1, \ldots, G_r\}$, and for each $1 \leq i \leq r$, we denote by $\mathcal{M}_i \subseteq G_i$ the set of the demand pairs contained in G_i, then the terminals participating in the demand pairs in $\mathcal{M}_i$ are α_{WL}-well-linked in G_i. We are then guaranteed that $\sum_{i=1}^{r} \mathsf{OPT}(G_i, \mathcal{M}_i) \geq \frac{63}{64}\mathsf{OPT}$. It is then tempting to apply the algorithm described above to each of the graphs separately. Indeed, if, for each $1 \leq i \leq r$, we find a set $\mathcal{P}_i$ of node-disjoint paths, routing $\Omega\left(\frac{\mathsf{OPT}(G_i,\mathcal{M}_i)}{N_i^{9/19} \cdot \text{poly} \log n}\right)$ demand pairs of $\mathcal{M}_i$ in G_i (where N_i denotes the number of the non-terminal vertices in G_i), then we obtain an $O(N^{9/19} \cdot \text{poly} \log n)$-approximate solution overall. Assume now that for some $1 \leq i \leq r$, we find a subset $\mathcal{M}'_i \subseteq \mathcal{M}_i$ of demand pairs, such that $\mathsf{OPT}(G_i, \mathcal{M}'_i) < w^*|\mathcal{M}'_i|/16$. Unfortunately, the set $\mathcal{M}'_i$ of demand pairs does not necessarily define a violated LP-constraint, since it is possible that $\mathsf{OPT}(G, \mathcal{M}'_i) >> \mathsf{OPT}(G_i, \mathcal{M}'_i)$, if the optimal routing uses many vertices of U (and possibly from some other graphs G_j). In general, the number of vertices in set U is relatively small compared to OPT, so in the global accounting across all instances $(G_{i'}, \mathcal{M}_{i'})$, only a small number of paths can use the vertices of U. But for any specific instance $(G_i, \mathcal{M}_i)$, it is possible that most paths in the optimal solution to instance $(G, \mathcal{M}_i)$ use the vertices of U. In order to overcome this difficulty, we need to perform a careful global accounting across all resulting instances $(G_i, \mathcal{M}_i)$.

Organization We start with preliminaries in Section 2. Section 3 is devoted to the proof of Theorem 1.1. Since this is not our main result, and the proof is somewhat long (though not very difficult), most of the proof appears in Section B of the Appendix. Sections 4–7 are devoted to the proof of Theorem 1.2: Section 4 provides an overview of the algorithm and some initial steps; Section 5 introduces the main technical tools that we use: enclosures, shells, and a partition of the terminals into subsets; and Sections 6 and 7 deal with Case 1 (when many terminals are far from each other) and Case 2 (when many terminals are close to the boundaries of at most two faces), respectively. We prove Theorem 1.3 in Section 8, and provide conclusions in Section 9. For convenience, we include in Section D of the Appendix a table of the main parameters used in the proof of Theorem 1.2.

2 Preliminaries

Given a graph G and a subset U of its vertices, we denote by $N(U)$ the set of all neighbors of U, that is, all vertices $v \in V(G) \setminus U$, such that there is an edge $(u, v) \in E(G)$ for some $u \in U$. We say that two paths P and P' are *internally disjoint* iff for every vertex $v \in P \cap P'$, v is an endpoint of both P and P'. Given a path P and a subset U of vertices of G, we say that P is internally disjoint from U iff every vertex in $P \cap U$ is an endpoint of P. Similarly, P is internally disjoint from a subgraph G' of G iff P is internally disjoint from $V(G')$. Given a graph G and a set $\mathcal{M}$ of demand pairs in G, for every subset $\mathcal{M}' \subseteq \mathcal{M}$ of the demand pairs, we denote by $\mathcal{T}(\mathcal{M}')$ the set of all vertices participating in the demand pairs in $\mathcal{M}'$. For a subset $\mathcal{M}' \subseteq \mathcal{M}$ of the demand pairs, and a sub-graph $H \subseteq G$, let $\mathsf{OPT}(H, \mathcal{M}')$ denote the value of the optimal solution to instance $(H, \mathcal{M}')$.

Given a drawing of any planar graph H in the plane, and given any cycle C in H, we denote by $D(C)$ the unique disc in the plane whose boundary is C. Similarly, if C is a closed simple curve in the plane, $D(C)$ is the unique disc whose boundary is C. When the graph H is drawn on the sphere, there are two discs whose boundaries are C. In such cases we will explicitly specify which of the two discs we refer to. Given any disc D (in the plane or on the sphere), we use D° to denote the disc D without its boundary. We say that a vertex of H belongs to disc D, and denote $v \in D$, if v is drawn inside D or on its boundary.

Given a planar graph G, drawn on a surface Σ, we say that a curve C in Σ is *G-normal*, iff it intersects the drawing of G at vertices only. The set of vertices of G lying on C is denoted by $V(C)$, and the length of C is $\ell(C) = |V(C)|$. For any disc D, whose boundary is a G-normal curve, we denote by $V(D)$ the set of all vertices of G lying inside D or on its boundary.

Definition 2.1 *Let γ, γ' be two curves in the plane or on the sphere. We say that γ and γ'* cross*, iff there is a disc D, whose boundary is a simple closed curve that we denote by β, such that:*

- *$\gamma \cap D$ is a simple open curve, whose endpoints we denote by a and b;*
- *$\gamma' \cap D$ is a simple open curve, whose endpoints we denote by a' and b'; and*
- *$a, a', b, b' \in \beta$, and they appear on β in this circular order.*

Given a graph G embedded in the plane or on the sphere, we say that two paths P, P' in G cross iff their images cross. Similarly, we say that a path P crosses a curve γ iff the image of P crosses γ.

Sparsest Cut. In this paper we use the node version of the sparsest cut problem, defined as follows. Suppose we are given a graph $G = (V, E)$ with a subset $\mathcal{T} \subseteq V$ of its vertices called terminals. A vertex cut is a tri-partition (A, C, B) of V, such that there are no edges in G with one endpoint in A and another in B. If $(A \cup C) \cap \mathcal{T}, (B \cup C) \cap \mathcal{T} \neq \emptyset$, then the sparsity of the cut (A, C, B) is $\frac{|C|}{\min\{|A\cap\mathcal{T}|,|B\cap\mathcal{T}|\}+|C\cap\mathcal{T}|}$. The sparsest cut in G with respect to the set $\mathcal{T}$ of terminals is a vertex cut (A, C, B) with $(A\cup C)\cap\mathcal{T}, (B\cup C)\cap\mathcal{T} \neq \emptyset$, whose sparsity is the smallest among all such cuts. Amir, Krauthgamer and Rao [AKR03] showed an efficient algorithm, that, given any planar graph G with a set $\mathcal{T} \subseteq V(G)$ of terminal vertices, computes a vertex cut (A, C, B) in G, whose sparsity with respect to $\mathcal{T}$ is within a constant factor of the optimal one. We denote this algorithm by $\mathcal{A}_{\text{AKR}}$, and the approximation factor it achieves by α_{AKR}, so α_{AKR} is a universal constant.

In the special case of the sparsest cut problem that we consider in our paper, all terminals have degree 1, and no edge of G connects any pair of terminals. We show that in this case we can compute a near-optimal solution (A, C, B) to the sparsest cut problem with $C\cap\mathcal{T} = \emptyset$. The proof of the following observation uses standard techniques and is deferred to the Appendix.

Observation 2.1 *Let G be a planar graph and let $\mathcal{T} \subseteq V(G)$ be a subset of its vertices called terminals, with $|\mathcal{T}| \geq 3$. Assume that the degree of every terminal is 1, and no edge of G connects any pair of terminals. Then there is an efficient algorithm to compute a vertex cut (A, C, B) in G, whose sparsity is within a factor α_{AKR} of the optimal one, and $C \cap \mathcal{T} = \emptyset$.*

Nested Segments. Suppose we are given a graph G, a cycle C in G, and a collection Σ of (not necessarily disjoint) segments of C, where each segment is either C itself, or a sub-path of C. We say that Σ is a *nested set of segments of C* iff for all $\sigma, \sigma' \in \Sigma$, either $\sigma \subseteq \sigma'$, or $\sigma' \subseteq \sigma$, or σ and σ' are internally disjoint - that is, every vertex in $\sigma \cap \sigma'$ is an endpoint of both segments. We define a set of nested segments of a closed curve C, and of a path P in G similarly.

Decomposition of Forests. A directed forest F is a disjoint union of arborescences $\tau_1, \ldots, \tau_r$ for some $r \geq 1$, where in each arborescence τ_i, all edges are directed towards the root. We use the following simple claim about partitioning directed forests into collections of paths. Similar decompositions were used in previous work, see e.g. Lemma 3.5 in [Kle05]. The proof is included in Appendix for completeness.

Claim 2.2 *There is an efficient algorithm, that, given a directed forest F with n vertices, computes a partition $\mathcal{R} = \left\{R_1, \ldots, R_{\lceil \log n \rceil}\right\}$ of $V(F)$ into subsets, such that for each $1 \leq j \leq \lceil \log n \rceil$, $F[R_j]$ is a collection of disjoint directed paths, that we denote by $\mathcal{P}_j$. Moreover, for all $v, v' \in R_j$, if there is a directed path from v to v' in F, then they both lie on the same path in $\mathcal{P}_j$.*

Routing on a Disc. Assume that we are given an instance $(G, \mathcal{M})$ of the NDP-Disc problem, where G is drawn in a disc D whose boundary is denoted by C. We need the following two definitions.

Definition 2.2 *We say that two demand pairs $(s, t), (s', t') \in \mathcal{M}$ cross iff either $\{s, t\}\cap\{s', t'\} \neq \emptyset$, or (s, s', t, t') appear on C in this circular order. We say that the set $\mathcal{M}$ of demand pairs is non-crossing if no two demand pairs in $\mathcal{M}$ cross.*

Definition 2.3 *Let C be a closed simple curve and $\mathcal{M}$ a set of demand pairs with all vertices of $\mathcal{T}(\mathcal{M})$ lying on C. We say that $\mathcal{M}$ is an r-split* collection of demand pairs *with respect to C, iff there is a partition $\mathcal{M}_1, \ldots, \mathcal{M}_r$ of the demand pairs in $\mathcal{M}$, and there is a partition $\{\sigma_1, \sigma_2, \ldots, \sigma_{2r}\}$ of C into* *disjoint segments, such that $\sigma_1, \ldots, \sigma_{2r}$ appear on C in this circular order, and for each $1 \leq i \leq r$, for every demand pair $(s,t) \in \mathcal{M}_i$, either $s \in \sigma_{2i-1}$ and $t \in \sigma_{2i}$, or vice versa.*

Finally, the following lemma allows us to partition any set of demand pairs into a small collection of split sets. The proof appears in the Appendix.

Lemma 2.3 *There is an efficient algorithm, that, given a closed simple curve C in the plane and a set $\mathcal{M}$ of κ demand pairs, whose corresponding terminals lie on C, computes a partition $\mathcal{M}^1, \ldots, \mathcal{M}^{4\lceil \log \kappa \rceil}$ of $\mathcal{M}$, such that for each $1 \leq i \leq 4\lceil \log \kappa \rceil$, set $\mathcal{M}^i$ is r_i-split with respect to C for some integer $r_i \geq 0$.*

Routing on a Cylinder. Assume that we are given an instance $(G, \mathcal{M})$ of the NDP-Cylinder problem, where Γ_1 and Γ_2 are the cuffs of the cylinder.

Definition 2.4 *We say that a set $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs is* non-crossing *if there is an ordering $(s_{i_1}, t_{i_1}), \ldots, (s_{i_r}, t_{i_r})$ of the demand pairs in $\mathcal{M}'$, such that $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$ are all distinct and appear in this counter-clock-wise order on Γ_1, and $t_{i_1}, t_{i_2}, \ldots, t_{i_r}$ are all distinct and appear in this counter-clock-wise order on Γ_2.*

It is immediate to verify that if we are given any instance $(G, \mathcal{M})$ of NDP-Cylinder, and any set $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs that can all be routed via node-disjoint paths in G, then set $\mathcal{M}'$ is non-crossing.

Tight Concentric Cycles. We start with the following definition.

Definition 2.5 *Given a planar graph H drawn in the plane and a vertex $v \in V(H)$ that is not incident to the infinite face,* min-cycle(H, v) *is the cycle C in H, such that: (i) $v \in D^\circ(C)$; and (ii) among all cycles satisfying (i), C is the one for which $D(C)$ is minimal inclusion-wise.*

It is easy to see that min-cycle(H, v) is uniquely defined. Indeed, consider the graph $H \setminus v$, and the face F in the drawing of $H \setminus v$ where v used to reside. Then the boundary of F contains exactly one cycle C with $D(C)$ containing v, and $C =$ min-cycle(H, v). We next define a family of tight concentric cycles.

Definition 2.6 *Suppose we are given a planar graph H, an embedding of H in the plane, a simple closed H-normal curve C, and an integral parameter $r \geq 1$. A family of r tight concentric cycles around C is a sequence $Z_1, Z_2, \ldots, Z_r$ of disjoint simple cycles in H, with the following properties:*

- *$D(C) \subsetneq D(Z_1) \subsetneq D(Z_2) \subsetneq \cdots \subsetneq D(Z_r)$;*
- *if H' is the graph obtained from H by contracting all vertices lying in $D(C)$ into a super-node a, then $Z_1 =$ min-cycle(H', a); and*
- *for every $1 < h \leq r$, if H' is the graph obtained from H by contracting all vertices lying in $D(Z_{h-1})$ into a super-node a, then $Z_h =$ min-cycle(H', a).*

We will sometimes allow C to be a simple cycle in H. The family of tight concentric cycles around C is then defined similarly.

Monotonicity of Paths and Cycles. Suppose we are given a planar graph H, embedded into the plane, a simple H-normal curve C in H, and a family $(Z_1, \ldots, Z_r)$ of tight concentric cycles around C. Assume further that we are given a set $\mathcal{P}$ of κ node-disjoint paths, originating at the vertices of C, and terminating at some vertices lying outside of $D(Z_r)$. We would like to re-route these paths to ensure that they are monotone with respect to the cycles, that is, for all $1 \leq h \leq r$, and for all $P \in \mathcal{P}$, $P \cap Z_h$ is a path. We first discuss re-routing to ensure monotonicity with respect to a single cycle, and then extend it to monotonicity with respect to a family of concentric cycles.

Definition 2.7 *Given a graph H, a cycle C and a path P in H, we say that P is monotone with respect to C, iff $P \cap C$ is a path.*

The proof of the following lemma is deferred to the Appendix.

Lemma 2.4 *Let H be a planar graph embedded into the plane, C a simple cycle in H, and $\mathcal{P}$ a collection of κ simple internally node-disjoint paths between two vertices: vertex s lying in $D^{\circ}(C)$, and vertex $t \notin D(C)$, that is incident on the outer face. Assume further that H is the union of C and the paths in $\mathcal{P}$, and that $C = \text{min-cycle}(H, s)$. Then there is an efficient algorithm to compute a set $\mathcal{P}'$ of κ internally node-disjoint paths connecting s to t in H, such that every path in $\mathcal{P}'$ is monotone with respect to C.*

We now define monotonicity with respect to a family of cycles.

Definition 2.8 *Let H be a graph, $\mathcal{Z} = (Z_1, \ldots, Z_r)$ a collection of r disjoint cycles, and $\mathcal{P}$ a collection of node-disjoint paths in H. We say that the paths in $\mathcal{P}$ are* monotone *with respect to $\mathcal{Z}$, iff for every $1 \leq h \leq r$, every path in $\mathcal{P}$ is monotone with respect to Z_h.*

The following theorem allows us to re-route sets of paths so they become monotone with respect to a given family of tight concentric cycles. Its proof is a simple application of Lemma 2.4 and is deferred to the Appendix.

Theorem 2.5 *Let H be a planar graph embedded in the plane, C any simple closed H-normal curve or a simple cycle in H, and $\mathcal{Z} = (Z_1, \ldots, Z_r)$ a family of r tight concentric cycles in H around C. Let $Y \subsetneq H$ be any connected subgraph of H lying completely outside of $D(Z_r)$, and let $\mathcal{P}$ be a set of κ node-disjoint paths, connecting a subset $A \subseteq V(C)$ of κ vertices to a subset $B \subseteq V(Y)$ of κ vertices, so that the paths of $\mathcal{P}$ are internally disjoint from $V(C) \cup V(Y)$. Let $H' = (\bigcup_{h=1}^{r} V(Z_h)) \cup \mathcal{P}$. Then there is an efficient algorithm to compute a collection $\mathcal{P}'$ of κ node-disjoint paths in H', connecting the vertices of A to the vertices of B, so that the paths in $\mathcal{P}'$ are monotone with respect to $\mathcal{Z}$, and they are internally node-disjoint from $V(C) \cup V(Y)$.*

3 Routing on a Disc and on a Cylinder

In this section we prove Theorem 1.1. In order to do so, we define a new problem, called Demand Pair Selection Problem (DPSP), and show an 8-approximation algorithm for it. We then show that both NDP-Disc and NDP-Cylinder reduce to DPSP.

Demand Pair Selection Problem We assume that we are given two disjoint directed paths, σ and σ', and a collection $\mathcal{M} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of pairs of vertices of $\sigma \cup \sigma'$ that are called demand pairs, where all vertices of $S = \{s_1, \ldots, s_k\}$ lie on σ, and all vertices of $T = \{t_1, \ldots, t_k\}$ lie on σ'

(not necessarily in this order). We refer to the vertices of S and T as the *source* and the *destination* vertices, respectively. Note that the same vertex of σ may participate in several demand pairs, and the same is true for the vertices of σ'. Given any pair a, a' of vertices of σ, with a lying before a' on σ, we sometimes denote by (a, a') the sub-path of σ between a and a' (that includes both these vertices), and we will sometimes refer to it as an interval. We define intervals of σ' similarly.

For every pair $v, v' \in V(\sigma)$ of vertices, we denote $v \prec v'$ if v lies strictly before v' on σ, and we denote $v \preceq v'$, if $v \prec v'$ or $v = v'$ hold. Similarly, for every pair $v, v' \in V(\sigma')$ of vertices, we denote $v \prec v'$ if v lies strictly before v' on σ', and we denote $v \preceq v'$, if $v \prec v'$ or $v = v'$ hold. We need the following definitions.

Definition 3.1 *Suppose we are given two pairs (a, b) and (a', b') of vertices of $\sigma \cup \sigma'$, with $a, a' \in \sigma$ and $b, b' \in \sigma'$. We say that (a, b) and (a', b')* cross *iff one of the following holds: either (i) $a = a'$; or (ii) $b = b'$; or (iii) $a \prec a'$ and $b' \prec b$; or (iv) $a' \prec a$ and $b \prec b'$.*

Definition 3.2 *We say that a subset $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs is* non-crossing *iff for all distinct pairs $(s, t), (s', t') \in \mathcal{M}'$, (s, t) and (s', t') do not cross.*

Our goal is to select the largest-cardinality non-crossing subset $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs, satisfying a collection $\mathcal{K}$ of constraints. Set $\mathcal{K}$ of constraints is given as part of the problem input, and consists of four subsets, $\mathcal{K}_1, \ldots, \mathcal{K}_4$, where constraints in set $\mathcal{K}_i$ are called *type-i constraints*. Every constraint $K \in \mathcal{K}$ is specified by a quadruple (i, a, b, w), where $i \in \{1, 2, 3, 4\}$ is the constraint type, $a, b \in V(\sigma \cup \sigma')$, and $1 \leq w \leq |\mathcal{M}|$ is an integer.

For every type-1 constraint $K = (1, a, b, w) \in \mathcal{K}_1$, we have $a, b \in V(\sigma)$ with $a \prec b$. The constraint is associated with the sub-path $I = (a, b)$ of σ. We say that a subset $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs *satisfies* K iff the total number of the source vertices participating in the demand pairs of $\mathcal{M}'$ that lie on I is at most w.

Similarly, for every type-2 constraint $K = (2, a, b, w) \in \mathcal{K}_2$, we have $a, b \in V(\sigma')$ with $a \prec b$, and the constraint is associated with the sub-path $I = (a, b)$ of σ'. A set $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs satisfies K iff the total number of the destination vertices participating in the demand pairs in $\mathcal{M}'$ that lie on I is at most w.

For each type-3 constraint $K = (3, a, b, w) \in \mathcal{K}_3$, we have $a \in V(\sigma)$ and $b \in V(\sigma')$. The constraint is associated with the sub-path L_a of σ between the first vertex of σ and a (including both these vertices), and the sub-path R_b of σ' between b and the last vertex of σ' (including both these vertices). We say that a demand pair $(s, t) \in \mathcal{M}$ *crosses* K iff $s \in L_a$ and $t \in R_b$. A set $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs satisfies K iff the total number of pairs $(s, t) \in \mathcal{M}'$ that cross K is bounded by w.

Finally, for each type-4 constraint $K = (4, a, b, w) \in \mathcal{K}_4$, we also have $a \in V(\sigma)$ and $b \in V(\sigma')$. The constraint is associated with the sub-path R_a of σ between a and the last vertex of σ (including both these vertices), and the sub-path L_b of σ' between the first vertex of σ' and b (including both these vertices). We say that a demand pair $(s, t) \in \mathcal{M}$ crosses K iff $s \in R_a$ and $t \in L_b$. A set $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs satisfies K iff the total number of pairs $(s, t) \in \mathcal{M}'$ that cross K is bounded by w.

Given the paths σ, σ', the set $\mathcal{M}$ of the demand pairs, and the set $\mathcal{K}$ of constraints as above, the goal in the DPSP problem is to select a maximum-cardinality non-crossing subset $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs, such that all constraints in $\mathcal{K}$ are satisfied by $\mathcal{M}'$. The proof of the following theorem is deferred to the Appendix.

Theorem 3.1 *There is an efficient 8-approximation algorithm for* DPSP.

We then use Theorem 3.1 in order to design $O(\log k)$-approximation algorithms for NDP-Disc and NDP-Cylinder. The remainder of the proof of Theorem 1.1 appears in Section B of the Appendix. The algorithm for NDP-Disc exploits the exact characterization of routable instances of the problem given by Robertson and Seymour [RS86], in order to reduce the problem to DPSP. The algorithm for NDP-Cylinder reduces the problem to NDP-Disc and DPSP.

4 Algorithm Setup

The rest of this paper mostly focuses on proving Theorem 1.2; we prove Theorem 1.3 using the techniques we employ for the proof of Theorem 1.2 in Section 8.

We assume without loss of generality that the input graph G is connected - otherwise we can solve the problem separately on each connected component of G. Let $\mathcal{T} = \mathcal{T}(\mathcal{M})$. It is convenient for us to assume that every terminal participates in exactly one demand pair, and that the degree of every terminal is 1. This can be achieved via a standard transformation of the input instance, where we add a new collection of terminals, connecting them to the original terminals. This transformation preserves planarity, but unfortunately it can increase the number of the graph vertices. If the original graph G contained n vertices, then $|\mathcal{M}|$ can be as large as n^2, and so the new graph may contain up to $n^2 + n$ vertices, while our goal is to obtain an $\tilde{O}(n^{9/19})$-approximation. In order to overcome this difficulty, we denote by N the number of the non-terminal vertices in the new graph G, so N is bounded by the total number of vertices in the original graph, and by n the total number of all vertices in the new graph, so $n = O(N^2)$. Our goal is then to obtain an efficient $O(N^{9/19} \cdot \text{poly} \log n)$-approximation for the new problem instance. From now on we assume that every terminal participates in exactly one demand pair, and the degree of every terminal is 1. If (s,t) is a demand pair, then we say that s is the mate of t, and t is the mate of s. We denote $|\mathcal{M}| = k$. Throughout the algorithm, we define a number of sub-instances of the instance $(G, \mathcal{M})$, but we always use k to denote the number of the demand pairs in this initial instance. We can assume that $k > 100$, as otherwise we can return a routing of a single demand pair.

We assume that we are given a drawing of G on the sphere. Throughout the algorithm, we will sometimes select some face of G as the outer face, and consider the resulting planar drawing of G.

4.1 LP-Relaxations

Let us start with the standard multicommodity flow LP-relaxation of the problem. Let G' be the directed graph, obtained from G by bi-directing its edges. For every edge $e \in E(G')$, for each $1 \le i \le k$, there is an LP-variable $f_i(e)$, whose value is the amount of the commodity-i flow through edge e. We denote by x_i the total amount of commodity-i flow sent from s_i to t_i. For every vertex v, let $\delta^+(v)$ and $\delta^-(v)$ denote the sets of its out-going and in-coming edges, respectively. We denote $[k] = \{1, \ldots, k\}$. The standard LP-relaxation of the NDP problem is as follows.

(LP-flow1)

$$
\begin{array}{lll}
\max & \sum_{i=1}^{k} x_i & \\
\text{s.t.} & & \\
& \sum_{e\in\delta^+(s_i)} f_i(e) = x_i & \forall i \in [k] \\
& \sum_{e\in\delta^+(v)} f_i(e) = \sum_{e\in\delta^-(v)} f_i(e) & \forall i \in [k], \forall v \in V(G') \setminus \{s_i, t_i\} \quad \text{(flow conservation)} \\
& \sum_{e\in\delta^+(v)} \sum_{i=1}^{k} f_i(e) \le 1 & \forall v \in V(G') \quad \text{(vertex capacity constraints)} \\
& f_i(e) \ge 0 & \forall i \in [k], \forall e \in E(G')
\end{array}
$$

We will make two changes to (LP-flow1). First, we will assume that we know the value X^* of the optimal solution, and instead of the objective function, we will add the constraint $\sum_{i=1}^{k} x_i \ge X^*$. We can do so using standard methods, by repeatedly guessing the value X^* and running the algorithm for each such value. It is enough to show that the algorithm routes $\Omega\left(\frac{X^*}{N^{9/19}\cdot \text{poly}\log n}\right)$ demand pairs, when the value X^* is guessed correctly.

Recall that for a subset $\mathcal{M}' \subseteq \mathcal{M}$ of the demand pairs, and a sub-graph $H \subseteq G$, $\mathsf{OPT}(H, \mathcal{M}')$ denotes the value of the optimal solution to instance $(H, \mathcal{M}')$. For every subset $\mathcal{M}' \subseteq \mathcal{M}$ of the demand pairs, we add the constraint that the total flow between all pairs in $\mathcal{M}'$ is no more than z, for all integers z between $\mathsf{OPT}(G, \mathcal{M}')$ and k. We now obtain the following linear program that has no objective function, so we are only interested in finding a feasible solution.

(LP-flow2)

$$
\begin{array}{llr}
\sum_{i=1}^{k} x_i \ge X^* & & (1) \\
\sum_{e\in\delta^+(s_i)} f_i(e) = x_i & \forall i \in [k] & (2) \\
\sum_{e\in\delta^+(v)} f_i(e) = \sum_{e\in\delta^-(v)} f_i(e) & \forall i \in [k], \forall v \in V(G') \setminus \{s_i, t_i\} \quad \text{(flow conservation)} & (3) \\
\sum_{e\in\delta^+(v)} \sum_{i=1}^{k} f_i(e) \le 1 & \forall v \in V(G') \quad \text{(vertex capacity constraints)} & (4) \\
\sum_{(s_i,t_i)\in\mathcal{M}'} x_i \le z & \forall \mathcal{M}' \subseteq \mathcal{M}, \forall z \in \mathbb{Z} : \mathsf{OPT}(G, \mathcal{M}') \le z \le k & (5) \\
f_i(e) \ge 0 & \forall i \in [k], \forall e \in E(G') & (6)
\end{array}
$$

We say that a solution to (LP-flow2) is *semi-feasible* iff all constraints of types (1)–(4) and (6) are satisfied. Notice that the number of the constraints in (LP-flow2) is exponential in k. In order to solve it, we will use the Ellipsoid Algorithm with a separation oracle, where our approximation algorithm itself will serve as the separation oracle. This is done via the following theorem, which is our main technical result.

Theorem 4.1 *There is an efficient algorithm, that, given any semi-feasible solution to (LP-flow2), either computes a routing of at least* $\Omega\left(\frac{X^*}{N^{9/19}\cdot \text{poly}\log n}\right)$ *demand pairs of* $\mathcal{M}$ *via node-disjoint paths, or returns a constraint of type (5), that is violated by the current solution.*

We can now obtain an $O\left(N^{9/19} \cdot \text{poly}\log n\right)$-approximation algorithm for NDP-Planar via the Ellipsoid algorithm. In every iteration, we start with some semi-feasible solution to (LP-flow2), and apply the algorithm from Theorem 4.1 to it. If the outcome is a solution routing at least $\Omega\left(\frac{X^*}{N^{9/19}\cdot \text{poly}\log n}\right)$

demand pairs in $\mathcal{M}$, then we obtain the desired approximate solution to the problem, assuming that X^* was guessed correctly. Otherwise, we obtain a violated constraint of type (5), and continue to the next iteration of the Ellipsoid Algorithm. Since the Ellipsoid Algorithm is guaranteed to terminate with a feasible solution after a number of iterations that is polynomial in the number of the LP-variables, this gives an algorithm that is guaranteed to return a solution of value $\Omega\left(\frac{X^*}{N^{9/19}\cdot \text{poly}\log n}\right)$ in time poly(n). From now on we focus on proving Theorem 4.1.

We note that, using standard techniques, we can efficiently obtain a flow-paths decomposition of any semi-feasible solution to (LP-flow2): we can efficiently find, for every demand pair (s_i, t_i), a collection $\mathcal{P}_i$ of paths, connecting s_i to t_i, and for each path $P \in \mathcal{P}_i$, compute a value $f(P)$, such that:

- For each $i \in [k]$, $\sum_{P\in\mathcal{P}_i} f(P) = x_i$;
- For each $i \in [k]$, $|\mathcal{P}_i| \leq n$; and
- For each $v \in V(G)$, $\sum_{i\in[k]} \sum_{\substack{P\in\mathcal{P}_i:\\ v\in P}} f(P) \leq 1$.

It is sometimes more convenient to work with the above flow-paths decomposition version of a given semi-feasible solution to (LP-flow2).

We now assume that we are given some semi-feasible solution (x, f) to (LP-flow2), and define a new fractional solution based on it, where the flow between every demand pair is either 0 or w^*, for some value $w^* > 0$. First, for each demand pair (s_i, t_i) with $x_i \leq \frac{1}{2k}$, we set $x_i = 0$ and we set the corresponding flow values $f_i(e)$ for all edges $e \in E$ to 0. Since we can assume that $X^* \geq 1$ if the graph is connected, the total amount of flow between the demand pairs remains at least $X^*/2$. We then partition the remaining demand pairs into $q = \lceil \log 2k \rceil$ subsets, where for $1 \leq j \leq q$, set $\mathcal{M}_j$ contains all demand pairs (s_i, t_i) with $\frac{1}{2^j} < x_i \leq \frac{1}{2^{j-1}}$. There is some index $1 \leq j^* \leq q$, such that the total flow between the demand pairs in $\mathcal{M}_{j^*}$ is at least $\Omega(X^*/\log k)$. Let $w^* = \frac{1}{2^j}$. We further modify the LP-solution, as follows. First, for every demand pair $(s_i, t_i) \notin \mathcal{M}_{j^*}$, we set $x_i = 0$, and the corresponding flow values $f_i(e)$ for all edges $e \in E$ to 0. Next, for every demand pair $(s_i, t_i) \in \mathcal{M}_{j^*}$, we let $\beta_i = w^*/x_i$, so $\beta_i \leq 1$. We set $x_i = w^*$, and the new flow values $f_i(e)$ are obtained by scaling the original values by factor β_i. This gives a new solution to (LP-flow2), that we denote by (x', f'). The total amount of flow sent in this solution is $\Omega(X^*/\log k)$, and it is easy to verify that constraints (2)–(4) and (6) are satisfied. For every demand pair $(s_i, t_i) \in \mathcal{M}_{j^*}$, $x'_i = w^*$, and for all other demand pairs (s_i, t_i), $x'_i = 0$. It is easy to see that for every demand pair (s_i, t_i), $x'_i \leq x_i$. Therefore, if we find a constraint of type (5) that is violated by the new solution, then it is also violated by the old solution. Our goal now is to either find an integral solution routing $\Omega\left(\frac{X^*}{N^{9/19}\cdot \text{poly}\log n}\right)$ demand pairs, or to find a constraint of type (5) violated by the new LP-solution. In particular, if we find a subset $\mathcal{M}' \subseteq \mathcal{M}_{j^*}$ of demand pairs, with $\mathsf{OPT}(G, \mathcal{M}') \leq w^*|\mathcal{M}'|/2$, then set $\mathcal{M}'$ defines a violated constraint of type (5) for (LP-flow2). Since from now on we only focus on demand pairs in $\mathcal{M}_{j^*}$, for simplicity we denote $\mathcal{M} = \mathcal{M}_{j^*}$.

4.2 Well-Linked Decomposition

Like many other approximation algorithms for routing problems, we decompose our input instance into a collection of sub-instances that have some useful well-linkedness properties. Since the routing is on node-disjoint paths, we need to use a slightly less standard notion of node-well-linkedness, defined below. Throughout this paper, we use a parameter $\alpha_{\text{WL}} = \frac{w^*}{512\cdot\alpha_{\text{AKR}}\cdot\log k}$.

Definition 4.1 *Given a graph H and a set $\mathcal{T}'$ of its vertices, we say that $\mathcal{T}'$ is* α_{WL}-well-linked *in H iff for every pair $\mathcal{T}_1, \mathcal{T}_2$ of disjoint equal-sized subsets of $\mathcal{T}$, there is a set $\mathcal{P}$ of at least $\alpha_{\text{WL}} \cdot |\mathcal{T}_1|$ node-disjoint paths in H, connecting vertices of $\mathcal{T}_1$ to vertices of $\mathcal{T}_2$.*

Definition 4.2 *Given a sub-graph $H \subseteq G$ and a subset $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs with $\mathcal{T}(\mathcal{M}') \subseteq V(H)$, we say that $(H, \mathcal{M}')$ is a* well-linked instance*, iff $\mathcal{T}(\mathcal{M}')$ is α_{WL}-well-linked in H.*

The following theorem uses standard techniques, and its proof is deferred to Appendix.

Theorem 4.2 *There is an efficient algorithm to compute a collection $G_1, \ldots, G_r$ of disjoint sub-graphs of G, and for each $1 \leq j \leq r$, a set $\mathcal{M}^j \subseteq \mathcal{M}$ of demand pairs with $\mathcal{T}(\mathcal{M}^j) \subseteq V(G_j)$, such that:*

- *For all $1 \leq j \leq r$, $(G_j, \mathcal{M}^j)$ is a well-linked instance;*
- *For all $1 \leq j \neq j' \leq r$, there is no edge in G with one endpoint in G_j and the other in $G_{j'}$;*
- $\sum_{j=1}^r |\mathcal{M}^j| \geq 63|\mathcal{M}|/64$*; and*
- $\left|V(G) \setminus \left(\bigcup_{j=1}^r V(G_j)\right)\right| \leq \frac{w^* \cdot |\mathcal{M}|}{64}$.

For each $1 \leq j \leq r$, let $W_j = w^*|\mathcal{M}^j|$ be the contribution of the demand pairs in $\mathcal{M}^j$ to the current flow solution and let $W = \sum_{j=1}^r W_j = \Omega(X^*/\log k)$. Let $n_j = |V(G_j)|$, and let N_j be the number of the non-terminal vertices in G_j. The main tool in proving Theorem 4.1 is the following theorem.

Theorem 4.3 *There is an efficient algorithm, that computes, for every $1 \leq j \leq r$, one of the following:*

1. *Either a collection $\mathcal{P}^j$ of node-disjoint paths, routing $\Omega\left(W_j^{1/19}/\operatorname{poly}\log n\right)$ demand pairs of $\mathcal{M}^j$ in G_j; or*
2. *A collection $\tilde{\mathcal{M}}^j \subseteq \mathcal{M}^j$ of demand pairs, with $|\tilde{\mathcal{M}}^j| \geq |\mathcal{M}^j|/2$, such that $\mathsf{OPT}(G_j, \tilde{\mathcal{M}}^j) \leq w^*|\tilde{\mathcal{M}}^j|/8$.*

Before we prove Theorem 4.3, we show that Theorem 4.1 follows from it. We apply Theorem 4.3 to every instance $(G_j, \mathcal{M}^j)$, for $1 \leq j \leq r$. We say that instance $(G_j, \mathcal{M}^j)$ is a type-1 instance, if the first outcome happens for it, and we say that it is a type-2 instance otherwise. Let $I_1 = \{j \mid (G_j, \mathcal{M}^j) \text{ is a type-1 instance}\}$, and similarly, $I_2 = \{j \mid (G_j, \mathcal{M}^j) \text{ is a type-2 instance}\}$. We consider two cases, where the first case happens when $\sum_{j \in I_1} W_j \geq W/2$.

Case 1: $\sum_{j \in I_1} W_j \geq W/2$. We show that in this case, our algorithm returns a routing of $\Omega\left(\frac{X^*}{N^{9/19} \cdot \operatorname{poly}\log n}\right)$ demand pairs. We need the following lemma, whose proof is deferred to the Appendix. The proof uses standard techniques: namely, we show that the treewidth of each graph G_j is at least $\Omega(W_j/\log k)$, and so G_j must contain a large grid minor.

Lemma 4.4 *For each $1 \leq j \leq r$, $N_j \geq \Omega(W_j^2/\log^2 k)$.*

The number of the demand pairs we route in each type-1 instance $(G_j, \mathcal{M}^j)$ is then at least:

$$\Omega\left(\frac{W_j^{1/19}}{\operatorname{poly}\log n}\right)=\Omega\left(\frac{W_j}{W_j^{18/19}\cdot\operatorname{poly}\log n}\right)$$
$$=\Omega\left(\frac{W_j}{\left(\sqrt{N_j}\log k\right)^{18/19}\cdot\operatorname{poly}\log n}\right)$$
$$=\Omega\left(\frac{W_j}{N^{9/19}\cdot\operatorname{poly}\log n}\right).$$

Overall, since $\sum_{j\in I_1}W_j\geq W/2$, the number of the demand pairs routed is $\Omega\left(\frac{W}{N^{9/19}\cdot\operatorname{poly}\log n}\right)=\Omega\left(\frac{X^*}{N^{9/19}\cdot\operatorname{poly}\log n}\right)$.

Case 2: $\sum_{j\in I_2}W_j\geq W/2$**.** Let $\mathcal{M}'=\bigcup_{j\in I_2}\tilde{\mathcal{M}}^j$. Then $|\mathcal{M}'|=\sum_{j\in I_2}|\tilde{\mathcal{M}}^j|\geq\sum_{j\in I_2}\frac{|\mathcal{M}^j|}{2}\geq\frac{1}{4}\sum_{j=1}^r|\mathcal{M}^j|\geq\frac{|\mathcal{M}|}{8}$. We claim that the following inequality, that is violated by the current LP-solution, is a valid constraint of (LP-flow2):

$$\sum_{(s_i,t_i)\in\mathcal{M}'}x'_i\leq w^*|\mathcal{M}'|/2. \tag{7}$$

In order to do so, it is enough to prove that $\mathsf{OPT}(G,\mathcal{M}')<w^*|\mathcal{M}'|/2$. Assume otherwise, and let $\mathcal{P}^*$ be the optimal solution for instance $(G,\mathcal{M}')$, so $|\mathcal{P}^*|\geq w^*|\mathcal{M}'|/2$. We say that a path $P\in\mathcal{P}^*$ is bad if it contains a vertex of $V(G)\setminus\left(\bigcup_{j=1}^r V(G_j)\right)$. The number of such bad paths is bounded by the number of such vertices - namely, at most $\frac{w^*\cdot|\mathcal{M}|}{64}\leq\frac{w^*|\mathcal{M}'|}{8}\leq\frac{|\mathcal{P}^*|}{2}$. Therefore, at least $w^*|\mathcal{M}'|/4$ paths in $\mathcal{P}^*$ are good. Each such path must be contained in one of the graphs G_j corresponding to a type-2 instance. For each $j\in I_2$, let $\hat{\mathcal{M}}^j\subseteq\tilde{\mathcal{M}}^j$ be the set of the demand pairs routed by good paths of $\mathcal{P}^*$. Then, on the one hand, $\sum_{j\in I_2}|\hat{\mathcal{M}}^j|\geq w^*|\mathcal{M}'|/4=w^*\sum_{j\in I_2}|\tilde{\mathcal{M}}^j|/4$, while, on the other hand, since all demand pairs in $\hat{\mathcal{M}}^j$ can be routed simultaneously in G_j, for all $j\in I_2$, $|\hat{\mathcal{M}}^j|\leq w^*|\tilde{\mathcal{M}}^j|/8$, a contradiction. We conclude that $\mathsf{OPT}(G,\mathcal{M}')<w^*|\mathcal{M}'|/2$, and (7) is a valid constraint of (LP-flow2).

From now on, we focus on proving Theorem 4.3. Since from now on we only consider one instance $(G_j,\mathcal{M}^j)$, for simplicity, we abuse the notation and denote G_j by G, and $\mathcal{M}^j$ by $\mathcal{M}$. As before, we denote $\mathcal{T}=\mathcal{T}(\mathcal{M})$. We denote by $W=w^*\cdot|\mathcal{M}|$ the total amount of flow sent between the demand pairs in the new set $\mathcal{M}$ in the LP solution (note that this is not necessarily a valid LP-solution for the new instance, as some of the flow-paths may use vertices lying outside of G^j). We use n to denote the number of vertices in G. Value k - the number of the demand pairs in the original instance - remains unchanged. Our goal is to either find a collection of node-disjoint paths routing $\Omega(W^{1/19}/\operatorname{poly}(\log(nk)))$ demand pairs of $\mathcal{M}$ in G, or to find a collection $\tilde{\mathcal{M}}\subseteq\mathcal{M}$ of at least $|\mathcal{M}|/2$ demand pairs, such that $\mathsf{OPT}(G,\tilde{\mathcal{M}})\leq w^*|\tilde{\mathcal{M}}|/8$. We will rely on the fact that all terminals are α_{WL}-well-linked in G, for $\alpha_{\text{WL}}=\Theta(w^*/\log k)$. We assume that G is connected, since otherwise all terminals must be contained in a single connected component of G and we can discard all other connected components.

We assume that we are given an embedding of the graph G into the sphere. For every pair $v,v'\in V(G)$ of vertices, we let $d_{\text{GNC}}(v,v')$ be the length of the shortest G-normal curve connecting v to v' in this embedding, minus 1. It is easy to verify that d_{GNC} is a metric: that is, $d_{\text{GNC}}(v,v)=0$, $d_{\text{GNC}}(v,v')=d_{\text{GNC}}(v',v)$, and the triangle inequality holds for d_{GNC}. The value $d_{\text{GNC}}(v,v')$ can be computed efficiently, by solving an appropriate shortest path problem instance in the graph dual to G. Given a vertex v and a subset U of vertices of G, we denote by $d_{\text{GNC}}(v,U) = \min_{u\in U}\{d_{\text{GNC}}(v,u)\}$. Similarly, given two subsets U,U' of vertices of G, we denote $d_{\text{GNC}}(U,U') = \min_{u\in U,u'\in U'}\{d_{\text{GNC}}(u,u')\}$. Finally, given a G-normal curve C, and a vertex v in G, we let $d_{\text{GNC}}(v,C) = \min_{u\in V(C)}\{d_{\text{GNC}}(v,u)\}$.

Over the course of the algorithm, we will sometimes select some face of the drawing of G as the outer face and consider the resulting drawing of G in the plane. The function d_{GNC} remains unchanged, and it is only defined with respect to the fixed embedding of G into the sphere.

5 Enclosures, Shells, and Terminal Subsets

In this section we develop some of the technical machinery that we use in our algorithm, and describe the first steps of the algorithm. We start with enclosures around the terminals.

5.1 Constructing Enclosures

Throughout the algorithm, we use a parameter $\Delta = \lceil W^{2/19}\rceil$. We assume that $W > \Omega(\Delta)$, since otherwise W is bounded by a constant, and we can return the routing of a single demand pair. The goal of this step is to construct enclosures around the terminals, that are defined below. Recall that G is embedded on the sphere.

Definition 5.1 *An* enclosure *for terminal $t\in\mathcal{T}$ is a simple disc D_t containing the terminal t, whose boundary is denoted by C_t, that has the following properties. (Recall that $V(D_t)$ is the set of all vertices of G contained in D_t.)*

- *C_t is a simple closed G-normal curve with $\ell(C_t) = \Delta$;*
- *$|\mathcal{T}\cap V(D_t)| \leq 4\Delta/\alpha_{\text{WL}}$; and*
- *$V(D_t)$ induces a connected graph in G.*

The goal of this section is to prove the following theorem.

Theorem 5.1 *There is an efficient algorithm, that constructs an enclosure D_t for every terminal $t\in\mathcal{T}$, such that for all $t,t'\in\mathcal{T}$:*

- *If $D_t\subseteq D_{t'}$, then $D_t = D_{t'}$; and*
- *If $D_t\cap D_{t'} = \emptyset$, then there are Δ node-disjoint paths between $V(C_t)$ and $V(C_{t'})$ in G.*

Notice that since $\ell(C_t) = \Delta$, every vertex of $V(C_t)$ is an endpoint of a path connecting $V(C_t)$ to $V(C_{t'})$. In order to prove the theorem, we need the following two simple claims.

Claim 5.2 *Let D be any disc on the sphere, whose boundary C is a simple G-normal curve, such that $1\leq |V(D)| < |V(G)|-1$, and $G[V(D)]$ is connected (we allow D to consist of a single point, which must coincide with a vertex of G). Then we can efficiently find a disc D' with $V(D)\subsetneq V(D')$ and $|V(D')| = |V(D)|+1$, such that $G[V(D')]$ is a connected graph. Moreover, if C' is the boundary of D', then C' is a simple G-normal curve with $\ell(C') = \ell(C)+1$.*

Proof: If D consists of a single point corresponding to a vertex $v \in V(G)$, then let u be any neighbor of v in G. It is easy to construct a disc D' whose boundary only contains the vertices v and u, and has all the required properties, with $V(D') = \{v, u\}$. We now assume that $|V(D)| > 1$.

Let $u \in V(C)$ be any vertex that has a neighbor in $V(G) \setminus V(D)$: since G is connected and $|V(D)| < |V(G)|$, such a vertex exists. Let u' be a vertex lying next to u on C. Then there must be a vertex $v \in V(G) \setminus V(D)$, such that: (i) edge $e = (u, v)$ belongs to G; and (ii) there is a simple G-normal curve γ' connecting u' to v, that intersects G only at its endpoints, and intersects D only at u'. Let σ, σ' be the two segments of C whose endpoints are u and u'.

Notice that due to the edge $e = (u, v)$, there is also a G-normal curve γ connecting u to v, that intersects G only at its endpoints, and intersects D only at u. Let C_1 be the concatenation of σ, γ and γ', and let C_2 be the concatenation of σ', γ and γ'.

Let $x \in V(G) \setminus (V(D) \cup \{v\})$ be any vertex (such a vertex exists since $|V(D)| < |V(G)| - 1$), and let F be any face in the drawing of $G \cup C \cup \gamma \cup \gamma'$ incident on x. We can view the face F as the outer face of our drawing, to obtain a drawing of $G \cup C \cup \gamma \cup \gamma'$ in the plane. Using this view, curve C_1 defines a disc D_1 and curve C_2 defines a disc D_2 in the plane. Exactly one of these discs contains the disc D - assume w.l.o.g. that it is D_1. We then set $D' = D_1$ and $C' = C_1$. It is now immediate to verify that D' has all required properties. □

Claim 5.3 *Let H be any connected planar graph drawn on a sphere, and let s and t be two distinct vertices of H. Assume that the maximum number of internally node-disjoint paths between s and t in H is κ. Then we can efficiently find a simple closed H-normal curve C of length κ on the sphere, separating s from t, with $s, t \notin V(C)$. Moreover, if U and U' denote the sets of vertices lying strictly on each side of C, then $H[U \cup V(C)]$ and $H[U' \cup V(C)]$ are both connected.*

Proof: By Menger's theorem, we can efficiently find a set $X \subseteq V(H) \setminus \{s, t\}$ of κ vertices such that s and t are separated in the graph $H \setminus X$.

Let $\mathcal{P} = \{P_1, \ldots, P_\kappa\}$ be any set of κ internally node-disjoint paths connecting s to t in H, and let H' be the sub-graph of H obtained by taking the union of the paths in $\mathcal{P}$. The drawing of H' on the sphere consists of κ faces, where the boundary of each face is the union of two distinct paths in $\mathcal{P}$. We assume that the faces are $F_1, \ldots, F_\kappa$, and we assume without loss of generality that the boundary of each face F_i is $P_i \cup P_{i+1}$ (and the boundary of F_κ is $P_\kappa \cup P_1$). Notice that for each $1 \leq i \leq \kappa$, X contains exactly one internal vertex of P_i, that we denote by x_i. Let P_i^s, P_i^t be the two paths in $P_i \setminus \{x_i\}$, where P_i^s is the path containing s, so P_i^t contains t.

Fix some $1 \leq i \leq \kappa$, and let $H_i \subseteq H$ be the graph induced by all vertices lying inside F_i or on its boundary. Let $H_i' = H_i \setminus \{x_i, x_{i+1}\}$. Then there is no path in H_i' connecting a vertex of $P_i^s \cup P_{i+1}^s$ to a vertex of $P_i^t \cup P_{i+1}^t$: such a path would contradict the fact that s is disconnected from t in $H \setminus X$. Therefore, there is a curve γ_i, connecting x_i to x_{i+1} inside F_i, that intersects H only at its endpoints. Curve γ_i partitions F_i into two subfaces: F_i' and F_i'', with s lying on the boundary of F_i' and t lying on the boundary of F_i''. Since H is a connected graph, and γ_i only intersects H at its endpoints, both subgraphs induced by the vertices lying in F_i' and its boundary, and in F_i'' and its boundary, are connected.

We build the curve C by concatenating all curves γ_i for $1 \leq i \leq \kappa$. It is easy to verify that C has all required properties. □

Proof of Theorem 5.1. We show an efficient algorithm to construct the enclosures with the desired properties. Throughout the algorithm, we maintain a set $\{D_t\}_{t \in \mathcal{T}}$ of enclosures, such that for every pair $t, t' \in \mathcal{T}$ of terminals, if $D_t \subseteq D_{t'}$, then $D_t = D_{t'}$.

The initial set of enclosures is obtained as follows. For each terminal $t \in \mathcal{T}$, let D'_t be the disc containing a single point - the image of the vertex t. We apply Claim 5.2 $\Delta - 1$ times to D'_t, to obtain a disc D_t whose boundary is a simple G-normal curve of length Δ, and $D'_t \subseteq D_t$, while $|V(D_t)| = \Delta$. By Claim 5.2, $V(D_t)$ induces a connected sub-graph in G. Since $|V(D_t)| = \Delta$, D_t is a valid enclosure for t. While there is a pair t, t' of terminals with $D_t \subsetneq D_{t'}$, we set $D_t = D_{t'}$. This finishes the definition of the initial set of enclosures. We then perform a number of iterations. In every iteration, we consider all pairs t, t' of terminals with $D_t \cap D_{t'} = \emptyset$, and check whether there are Δ node-disjoint paths connecting the vertices of $V(C_t)$ to the vertices of $V(C_{t'})$ in G. If so, then we say that (t, t') is a good pair. If all such pairs are good, then we terminate the algorithm, and output the current set of enclosures. Otherwise, let (t, t') be a bad pair. Let H be the graph constructed from G, as follows: we delete all vertices of $V(D_t \setminus C_t)$, and add a source vertex a to the interior of D_t. We then connect a to every vertex of $V(C_t)$ with an edge. Similarly, we delete all vertices of $V(D_{t'} \setminus C_{t'})$, and add a destination vertex b to the interior of $D_{t'}$. We then connect b to every vertex of $V(C_{t'})$.

Let $\kappa \leq \Delta - 1$ be the maximum number of internally node-disjoint paths connecting a to b in H. We apply Claim 5.3 to find an H-normal closed curve C of length κ, separating a from b. Then C defines a G-normal curve of length κ, such that, if U and U' denote the sets of vertices of G lying strictly on each side of C, then $V(D_t) \subseteq U \cup V(C)$; $V(D_{t'}) \subseteq U' \cup V(C)$; and both $G[U \cup V(C)]$ and $G[U' \cup V(C)]$ are connected.

Notice that either $|U \cap \mathcal{T}| \leq |\mathcal{T}|/2$ or $|U' \cap \mathcal{T}| \leq |\mathcal{T}|/2$ holds - we assume w.l.o.g. that it is the former. Let D be the disc whose boundary is C, with $D_t \subseteq D$. Since $\ell(C) < \Delta$, from the well-linkedness of the terminals $|V(D) \cap \mathcal{T}| \leq \Delta/\alpha_{\text{WL}}$. We next apply Claim 5.2 to D repeatedly to obtain a disc D', whose boundary C' is a simple G-normal curve of length Δ, so that $D \subseteq D'$; $|V(D')| \leq |V(D)| + \Delta$, and $G[D']$ is a connected graph. It is easy to verify that D' is a valid enclosure for terminal t. We replace D_t with D'. If there is any terminal t'' with $D_{t''} \subsetneq D'$, then we replace $D_{t''}$ with D' as well. This finishes the description of an iteration. Notice that $\sum_{t \in \mathcal{T}} |V(D_t)|$ increases by at least 1 in every iteration, and so the number of iterations is bounded by $|V(G)|$. □

Distances between terminals. For every pair t, t' of terminals, we define the distance $d(t, t')$ between t and t' to be the length of the shortest G-normal open curve, with one endpoint in $V(C_t)$ and another in $V(C_{t'})$. (Notice that if $D_t \cap D_{t'} \neq \emptyset$, then $d(t, t') = 1$). We repeatedly use the following simple observation (a weak triangle inequality):

Observation 5.4 *For all $t, t', t'' \in \mathcal{T}$, $d(t, t'') \leq d(t, t') + d(t', t'') + \Delta/2$.*

Proof: Let γ be a G-normal curve of length $d(t, t')$ connecting a vertex of C_t to a vertex of $C_{t'}$, and let γ' be defined similarly for $d(t', t'')$. Let $u, u' \in V(C_{t'})$ be the vertices that serve as endpoints of γ and γ', respectively, and let σ be the shorter of the two segments of $C_{t'}$ between u and u', so the length of σ is at most $\Delta/2 + 2$. By combining γ, σ and γ', we obtain a G-normal curve of length at most $d(t, t') + d(t', t'') + \Delta/2$, connecting a vertex of C_t to a vertex of $C_{t''}$. □

5.2 Constructing Shells

Suppose we are given some terminal $t \in \mathcal{T}$ and an integer $r \geq 1$. In this section we show how to construct a shell of depth r around t, and explore its properties. Shells play a central role in our algorithm. In order to construct the shell, we need to fix a plane drawing of the graph G, by choosing one of the faces F_t of the drawing of G on the sphere as the outer face. The choice of the face F_t will affect the construction of the shell, but once the face F_t is fixed, the shell construction is fixed as well.

We require that for every vertex v on the boundary of F_t, $d_{\text{GNC}}(v, C_t) \geq r+1$, and that C_t separates all vertices on the boundary of F_t from t. We note that when we construct shells for different terminals t, t', we may choose different faces $F_t, F_{t'}$, and thus obtain different embeddings of G into the plane. We now define a shell.

Definition 5.2 *Suppose we are given a terminal $t \in \mathcal{T}$, a face F_t in the drawing of G on the sphere, and an integer $r \geq 1$, such that for every vertex v on the boundary of F_t, $d_{\text{GNC}}(v, C_t) \geq r+1$, and C_t separates t from the boundary of F_t.*

A shell $\mathcal{Z}^r(t)$ of depth r around t with respect to F_t is a collection $\mathcal{Z}^r(t) = (Z_1(t), Z_2(t), \ldots, Z_r(t))$ of r tight concentric cycles around C_t. In other words, all cycles $Z_h(t)$ are simple and disjoint from each other, and the following properties hold. For each $1 \leq h \leq r$, let $D(Z_h(t))$ be the disc whose boundary is $Z_h(t)$ in the planar drawing of G with F_t as the outer face. Then:

J1. $D_t \subsetneq D(Z_1(t)) \subsetneq D(Z_2(t)) \subsetneq \cdots \subsetneq D(Z_r(t))$; and

J2. for every $1 \leq h \leq r$, if H is the graph obtained from G by contracting all vertices lying in $D(Z_{h-1}(t))$ into a super-node a, then $Z_h(t) = \text{min-cycle}(H, a)$ (when $h = 1$, we contract D_t into a super-node a).

Notice that from this definition we immediately obtain the following additional properties:

J3. For every $1 \leq h \leq r$, for every vertex $v \in V(Z_h(t))$, there is a G-normal curve of length 2 connecting v to some vertex of $V(Z_{h-1}(t))$ (or to a vertex of $V(C_t)$ if $h = 1$).

J4. For every $1 \leq h \leq r$, for every vertex $v \in V(Z_h(t))$, there is a G-normal curve $\gamma(v)$ of length $h+1$ connecting v to some vertex of $V(C_t)$, so that $\gamma(v) \subseteq D(Z_h(t))$, and it is internally disjoint from $Z_h(t)$ and C_t.

Let $\tilde{U}$ be the set of all vertices $v \in V(G) \setminus V(D_t)$, such that $d_{\text{GNC}}(v, C_t) < r+1$. Clearly, $G \setminus \tilde{U} \neq \emptyset$, since the vertices on the boundary of F_t do not lie in $\tilde{U}$. We let Y_t be the connected component of $G \setminus \tilde{U}$ containing the vertices on the boundary of F_t. The following additional property follows from the definition of the shell:

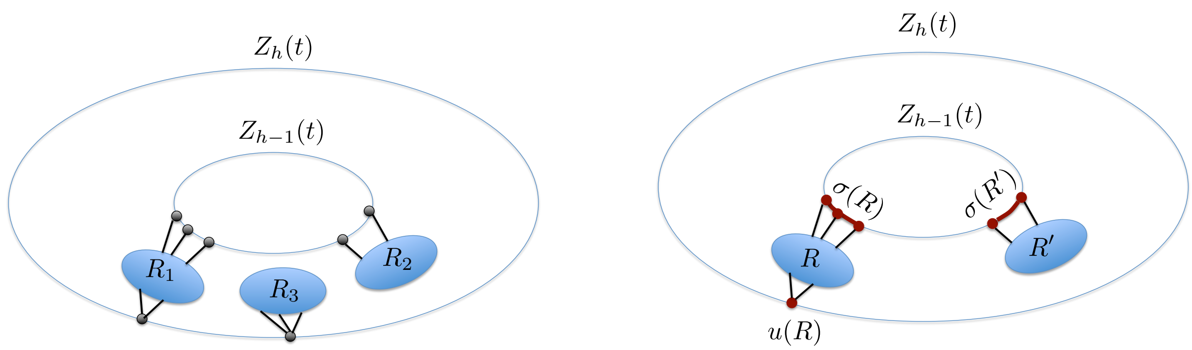
J5. All vertices of Y_t lie outside $D(Z_r(t))$.

Indeed, from Property (J4), for each $1 \leq h \leq r$, $Z_h(t) \cap Y_t = \emptyset$. Since Y_t is connected, $Z_r(t)$ must separate $V(D_t)$ from Y_t.

For convenience, we will always denote $Z_0(t) = C_t$ and $D(Z_0(t)) = D_t$. Note that $Z_0(t)$ is not a cycle in G – it is a simple closed G-normal curve, and so it is not part of the shell. We build the cycles $Z_1(t), \ldots, Z_r(t)$ one-by-one, and we maintain the invariant that for each $1 \leq h \leq r$, $(Z_1(t), \ldots, Z_h(t))$ is a shell of depth h around t with respect to F_t.

Assume that we have defined $Z_0(t), \ldots, Z_{h-1}(t)$ for some $1 \leq h < r$, such that the above invariant holds. In order to define $Z_h(t)$, consider the drawing of the graph G in the plane with F_t as the outer face, and delete all vertices lying in $D(Z_{h-1}(t))$ from it. Consider the face F where the deleted vertices used to be. Let Γ be the inner boundary of F. Then Γ must contain a single simple cycle Z, such that $D(Z_{h-1}(t)) \subseteq D(Z)$ (if no such cycle exists, then from Invariant (J4), for some vertex v on the boundary of F_t, there is a G-normal curve of length at most $r+1$ connecting v to C_t, which contradicts the choice of F_t). We let $Z_h(t) = Z$. It is easy to see that the invariant continues to hold. This finishes the construction of the shell. We now study its properties.

For each $1 \le h \le r$, we let U_h be the set of all vertices of G lying in $D^\circ(Z_h(t)) \setminus D(Z_{h-1}(t))$. Let $\mathcal{R}_h$ be the set of all connected components of $G[U_h]$. Each connected component $R \in \mathcal{R}_h$ may have at most one neighbor in $V(Z_h(t))$ from the definition of the shell, and it may have a number of neighbors in $V(Z_{h-1}(t))$. We say that $R \in \mathcal{R}_h$ is a *type-1 component* if it has one neighbor in $V(Z_h(t))$, and at least one neighbor in $V(Z_{h-1}(t))$. We denote by $u(R)$ the unique neighbor of R in $V(Z_h(t))$, and we denote by $L(R)$ the set of the neighbors of R that belong to $V(Z_{h-1}(t))$. We say that R is a *type-2 component*, if it has at least one neighbor in $V(Z_{h-1}(t))$ and no neighbors in $V(Z_h(t))$. In this case, we let $L(R)$ be the set of the neighbors of R lying in $V(Z_{h-1}(t))$, and $u(R)$ is undefined. Otherwise, we say that it is a *type-3 component*. In this case, it has exactly one neighbor in $V(Z_h(t))$, that we denote by $u(R)$, and no neighbors in $V(Z_{h-1}(t))$, so we set $L(R) = \emptyset$ (see Figure 1(a)). We sometimes refer to the vertices in set $L(R)$ as *the legs of R*.



(a) The three types of components in $\mathcal{R}_h$: R_1 is of type 1, R_2 is of type 2, and R_3 is of type 3.

(b) Definitions of vertices $u(R)$ and segments $\sigma(R)$

Figure 1: Structure of the shells

Consider now any type-1 or type-2 component $R \in \mathcal{R}_h$. For each such component R, we define a segment $\sigma(R)$ of $Z_{h-1}(R)$ containing all vertices of $L(R)$ as follows. If $|L(R)| = 1$, then let $\sigma(R)$ consist of the unique vertex of $L(R)$. Otherwise, we let $\sigma(R)$ be the smallest (inclusion-wise) segment of $Z_{h-1}(t)$ containing all vertices of $L(R)$, such that, if we let C be the union of $\sigma(R)$ and the outer boundary of the drawing of $G[V(R) \cup L(R)]$, then C separates R from t (see Figure 1(b)). Notice that $\{\sigma(R) \mid R \in \mathcal{R}_h \text{ is of type 1 or 2}\}$ is a nested set of segments of $Z_{h-1}(t)$. For consistency, for each type-3 component R, we define $\sigma(R) = \emptyset$. We denote $\mathcal{R} = \bigcup_{h=1}^{r} \mathcal{R}_h$. We will repeatedly use the following theorem.

Theorem 5.5 *There is an efficient algorithm, that computes, for every $1 \le h \le r$, for every component $R \in \mathcal{R}_h$, a disc $\eta(R) \subseteq D(Z_h(t))$, whose boundary, denoted by $\gamma(R)$, is a simple closed G-normal curve of length at most $2h + 1 + \Delta/2$, such that:*

1. *For all $R \in \mathcal{R}$, $G[V(R) \cup L(R)] \subseteq \eta(R)$, and, if $u(R)$ is defined, then $u(R) \in \eta(R)$;*
2. *For all $R \in \mathcal{R}$, $\gamma(R)$ is disjoint from all vertices in $\bigcup_{R' \in \mathcal{R}} V(R')$. In particular, for all $R' \in \mathcal{R}$, either $R' \subseteq \eta(R)$, or R' lies completely outside $\eta(R)$; and*
3. *For all $1 \le h \le r$, for all $R, R' \in \mathcal{R}_h$, either $\eta(R) \subseteq \eta(R')$, or $\eta(R') \subseteq \eta(R)$, or $\eta^\circ(R) \cap \eta^\circ(R') = \emptyset$. Moreover, if $R' \subseteq \eta(R)$, then $\sigma(R') \subseteq \sigma(R)$.*

Proof: We use the planar drawing of G where F_t is the outer face. Fix some $1 \le h \le r$. Let $\mathcal{R}_h^1, \mathcal{R}_h^2$, and $\mathcal{R}_h^3$ be the sets of type-1, type-2, and type-3 components of $\mathcal{R}_h$, respectively. For every type-3 component $R \in \mathcal{R}_h^3$, we let $\gamma(R)$ be a simple closed G-normal curve containing a single vertex of G - vertex $u(R)$, such that the disc $\eta(R)$, whose boundary is $\gamma(R)$, contains R, and it is disjoint from all other components of $\mathcal{R}$.

Consider now any type-1 component $R \in \mathcal{R}_h$, and let $a(R), a'(R)$ be the endpoints of $\sigma(R)$. Let $H \subseteq G$ be obtained from $G[V(R) \cup L(R)]$, by adding all edges connecting $u(R)$ to $V(R)$ to it. We draw two G-normal curves, $\gamma_1(R)$ connecting $u(R)$ to $a(R)$, and $\gamma_1'(R)$, connecting $u(R)$ to $a'(R)$ on either side of R, such that the curves $\gamma_1(R), \gamma_1'(R)$ do not contain any other vertices of G, and they follow the boundary of the drawing of H from the outside. Let $\gamma'(R)$ be the union of $\gamma_1(R)$ and $\gamma_1'(R)$, and let $\eta'(R)$ be the disc whose boundary is the union of $\gamma'(R)$ and $\sigma(R)$.

Similarly, given any type-2 component $R \in \mathcal{R}_h$, we denote by $a(R), a'(R)$ be the endpoints of $\sigma(R)$. Let $H = G[V(R) \cup L(R)]$, and let $\gamma'(R)$ be a simple G-normal curve connecting $a(R)$ to $a'(R)$, such that $\gamma'(R)$ does not contain any other vertices of G, and it follows the boundary of the drawing of H from the outside. Let $\eta'(R)$ be the disc whose boundary is the union of $\gamma'(R)$ and $\sigma(R)$.

Clearly, we can draw the curves $\left\{\gamma_1(R), \gamma_1'(R) \mid R \in \mathcal{R}_h^1\right\} \cup \left\{\gamma'(R) \mid R \in \mathcal{R}_h^2\right\}$, so that for all $R, R' \in \mathcal{R}_h^1 \cup \mathcal{R}_h^2$, either $\eta'(R) \subseteq \eta'(R')$, or $\eta'(R') \subseteq \eta'(R)$, or the interiors of $\eta'(R)$ and $\eta'(R')$ are disjoint. Moreover, if $\eta'(R) \subseteq \eta'(R')$, then $\sigma(R) \subseteq \sigma(R')$. Notice that the curves $\gamma'(R)$ are disjoint from all vertices in $\bigcup_{R'' \in \mathcal{R}} V(R'')$.

Let $A = \left\{a(R), a'(R) \mid R \in \mathcal{R}_h^1 \cup \mathcal{R}_h^2\right\}$. Recall that from Property (J4), for every vertex $a \in A$, there is a G-normal curve $\gamma(a)$ of length at most h, connecting a to some vertex $v \in V(C_t)$, such that $\gamma(a) \subseteq D(Z_{h-1}(t))$. In particular, $\gamma(a)$ must be disjoint from all vertices in $\bigcup_{R \in \mathcal{R}} V(R)$, as it must contain one vertex from each cycle $Z_1(t), \ldots, Z_{h-1}(t)$, and a vertex of $Z_0(t)$. Let $\Gamma = \{\gamma(a) \mid a \in A\}$. We can assume without loss of generality that the curves in Γ are non-crossing, and moreover, whenever two such curves meet, they continue together. In other words, if $\gamma(a) \cap \gamma(a') \neq \emptyset$, then this intersection is a simple G-normal curve that contains a vertex of C_t.

We say that a component $R \in \mathcal{R}_h^1 \cup \mathcal{R}_h^2$ is good if $\gamma(a(R))$ and $\gamma(a'(R))$ intersect, and we say that it is bad otherwise. From our assumption, if R is good, then $\gamma(a(R)) \cap \gamma(a'(R))$ is a curve, connecting some vertex v to some vertex of C_t. We then let $\gamma(R)$ be the union of $\gamma'(R)$, the segment of $\gamma(a(R))$ from $a(R)$ to v, and the segment of $\gamma(a'(R))$ from $a'(R)$ to v, and we let $\eta(R)$ be the disc whose boundary is $\gamma(R)$. Notice that for every component $R' \in \mathcal{R}_h^1 \cup \mathcal{R}_h^2$, if $\eta'(R') \subseteq \eta'(R)$, then R' is also a good component, and $\eta(R') \subseteq \eta(R)$.

For every bad component $R \in \mathcal{R}_h^1 \cup \mathcal{R}_h^2$, we let $b(R), b'(R) \in V(C_t)$ be the endpoints of $\gamma(a(R))$ and $\gamma(a'(R))$, respectively, and we let $\sigma'(R)$ be the segment of C_t, whose endpoints are $b(R)$ and $b'(R)$, such that the disc whose boundary is $\gamma'(R) \cup \sigma'(R) \cup \gamma(a(R)) \cup \gamma(a'(R))$ does not contain D_t.

Notice that the segments $\sigma'(R)$ for all bad components R form a nested set of intervals on C_t. This is since their corresponding segments $\sigma(R)$ are nested, and the curves in Γ are non-crossing. We say that a bad component R is large, iff $\sigma'(R)$ contains more than $\Delta/2+1$ vertices. Let $\mathcal{R}'$ be the set of all large bad components of $\mathcal{R}_h^1 \cup \mathcal{R}_h^2$. Then we can find an ordering $(R_1, R_2, \ldots, R_\ell)$ of the components in $\mathcal{R}'$, so that $\sigma'(R_1) \subseteq \sigma'(R_2) \subseteq \cdots \subseteq \sigma'(R_\ell)$.

For every bad component $R \notin \mathcal{R}'$, we let $\gamma(R)$ be the union of $\gamma'(R), \sigma'(R), \gamma(a(R))$, and $\gamma(a'(R))$, and we let $\eta(R)$ be the disc whose boundary is $\gamma(R)$. For every bad component $R \in \mathcal{R}'$, we let $\sigma''(R)$ be the segment of C_t with endpoints $b(R)$ and $b'(R)$, that is different from $\sigma'(R)$, so the length of $\sigma''(R)$ is at most $\Delta/2+1$. We let $\gamma(R)$ be the union of $\gamma'(R), \sigma''(R), \gamma(a(R))$, and $\gamma(a'(R))$, and we let $\eta(R)$ be the disc whose boundary is $\gamma(R)$. Notice that in either case, the length of $\gamma(R)$ is bounded by $2h+\Delta/2+1$. It is immediate to verify that the resulting discs $\eta(R)$ for all $R \in \bigcup_{h'=1}^{r} \mathcal{R}_{h'}$ have all required properties. Indeed, notice that for all type-1 and type-2 components $R \in \mathcal{R}_h$, $\eta'(R) \subseteq \eta(R)$. The first property then follows from the definition of $\eta'(R)$. For the third property, $\eta(R) \subseteq \eta(R')$ for $R, R' \in \mathcal{R}_h$ only if $\eta'(R) \subseteq \eta'(R')$, and from the construction of the discs $\eta'(R), \eta'(R')$, this can only happen if $\sigma(R) \subseteq \sigma(R')$. If neither of the discs $\eta'(R), \eta'(R')$ is contained in the other, then they are internally disjoint, and our construction of the discs $\eta(R), \eta(R')$ ensures that these two discs are also internally disjoint. Finally, consider two components $R, R' \in \mathcal{R}$. Our construction of the disc $\eta(R)$ ensures that $\gamma(R)$ is disjoint from $V(R')$, and so either $R' \subseteq \eta(R)$, or $R' \cap \eta(R) = \emptyset$. This establishes the remaining property.

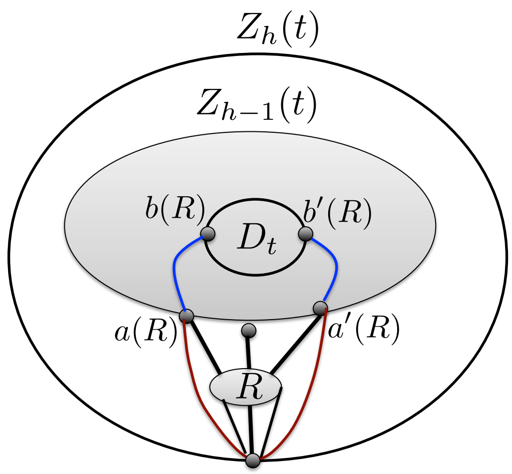


Figure 2: Building $\eta(R)$. Curve $\gamma'(R)$ is shown in red, and $\gamma(a(R)), \gamma(a'(R))$ are shown in blue.

□

We also need the following observation.

Observation 5.6 *For all $1 \leq h \leq r$, if $\tilde{U}_h$ is the set of vertices of G lying in $D(Z_h(t))$, then $G[\tilde{U}_h]$ is connected.*

Proof: The proof is by induction on h. Recall that $V(D_t)$ induces a connected sub-graph in G, from the definition of the enclosures. Assume now that $G[\tilde{U}_h]$ is connected for some $0 \leq h < r$. From the definition of $Z_{h+1}(t)$, and from the fact that G is a connected graph, it is immediate to verify that $G[\tilde{U}_{h+1}]$ is also connected. □

Finally, we need the following observation about the interactions of different shells.

Observation 5.7 *Let t, t' be any pair of terminals, and let $r, r' > 0$ be any integers, such that $d(t,t') > r + r' + 1$. Let $\mathcal{Z}^r(t) = (Z_1(t), \ldots, Z_r(t))$ be a shell of depth r around t with respect to some face F_t, and let $\mathcal{Z}^{r'}(t') = (Z_1(t') \ldots, Z_{r'}(t'))$ be a shell of depth r' around t' with respect to some face $F_{t'}$. Then for all $1 \leq h \leq r$ and $1 \leq h' \leq r'$, $Z_h(t) \cap Z_{h'}(t') = \emptyset$.*

Proof: Assume otherwise. Then there is some vertex $v \in Z_h(t) \cap Z_{h'}(t')$. From Property (J4), there is a G-normal curve $\gamma(v)$ of length at most $h + 1 \leq r + 1$ connecting v to some vertex of C_t, and similarly there is a G-normal curve $\gamma'(v)$ of length at most $h' + 1 \leq r' + 1$, connecting v to some vertex of $C_{t'}$. Combining $\gamma(v)$ with $\gamma'(v)$, we obtain a G-normal curve of length at most $r + r' + 1 < d(t,t')$, connecting a vertex of C_t to a vertex of $C_{t'}$, a contradiction. □

5.3 Terminal Subsets

Let $\Delta_0 = 20\left(\left\lceil \log_{10/9} n \right\rceil + 1\right)\Delta = \Theta(\Delta \log n)$. Our next step is to define a family of disjoint subsets of terminals, so that the terminals within each subset are close to each other, while the terminals belonging to different subsets are far enough from each other. We will ensure that almost all terminals of $\mathcal{T}$ belong to one of the resulting subsets. Following is the main theorem of this section.

Theorem 5.8 *There is an efficient algorithm to compute a collection $\mathcal{X} = \{X_1, \ldots, X_q\}$ of disjoint subsets of terminals of $\mathcal{T}$, such that:*

- *for each $1 \le i \le q$, for every pair $t, t' \in X_i$ of terminals, $d(t,t') \le \Delta_0$;*
- *for all $1 \le i \ne j \le q$, for every pair $t \in X_i$, $t' \in X_j$ of terminals $d(t,t') \ge 5\Delta$; and*
- *$\sum_{i=1}^{q} |X_i| \ge 0.9|\mathcal{T}|$.*

Proof: We start with $\mathcal{X} = \emptyset$ and $\tilde{\mathcal{T}} = \mathcal{T}$, and perform a number of iterations, each of which adds one subset of terminals to $\mathcal{X}$, and removes some terminals from $\tilde{\mathcal{T}}$. The iterations are executed while $\tilde{\mathcal{T}} \neq \emptyset$, and each iteration is executed as follows.

Let t^* be any terminal in $\tilde{\mathcal{T}}$. For all $1 \le i \le \left\lceil \log_{10/9} n \right\rceil + 1$, let Y_i contain all terminals $t \in \tilde{\mathcal{T}}$ with $d(t^*, t) \le 8\Delta i$. We use the following simple observation.

Observation 5.9 *There is some $2 \le i \le \left\lceil \log_{10/9} n \right\rceil + 1$, such that $|Y_i \setminus Y_{i-1}| \le |Y_{i-1}|/9$.*

Proof: If the claim is false, then for every $2 \le i \le \left\lceil \log_{10/9} n \right\rceil + 1$, $|Y_i| \ge 10|Y_{i-1}|/9$, and so $Y_{\lceil \log_{10/9} n \rceil + 1} > n$, which is impossible. □

Fix some $2 \le i \le \left\lceil \log_{10/9} n \right\rceil + 1$, such that $|Y_i \setminus Y_{i-1}| \le |Y_{i-1}|/9$. Notice that for every terminal $t \in Y_{i-1}$, $d(t^*, t) \le 8\Delta\left(\left\lceil \log_{10/9} n \right\rceil + 1\right)$, and so from Observation 5.4, for any pair $t, t' \in Y_{i-1}$ of terminals, $d(t,t') \le 16\Delta\left(\left\lceil \log_{10/9} n \right\rceil + 1\right) + \Delta/2 \le \Delta_0$. Moreover, if we consider any pair $t \in Y_{i-1}$, $t' \in \tilde{\mathcal{T}} \setminus Y_i$ of terminals, then $d(t,t') \ge 5\Delta$ must hold, since otherwise, from Observation 5.4, $d(t^*, t') \le d(t^*, t) + d(t,t') + \Delta/2 \le 8\Delta(i-1) + 5\Delta + \Delta/2 < 8\Delta i$, contradicting the fact that $t' \notin Y_i$. We remove all terminals of Y_i from $\tilde{\mathcal{T}}$, add the set $X = Y_{i-1}$ to $\mathcal{X}$, and continue to the next iteration.

Notice that in every iteration we discard all terminals in $Y_i \setminus Y_{i-1}$, and add a set containing $|Y_{i-1}|$ terminals to $\mathcal{X}$, where $|Y_i \setminus Y_{i-1}| \le |Y_{i-1}|/9$. Therefore, at the end of the algorithm, $\sum_{X \in \mathcal{X}} |X| \ge 0.9|\mathcal{T}|$. The remaining properties of the partition are now immediate to verify. □

We use a parameter $\tau = W^{18/19}$. We say that a set $X \in \mathcal{X}$ of terminals is *heavy* if $w^*|X| \ge \tau$, and we say that it is *light* otherwise. We say that a demand pair (s,t) is heavy iff both s and t belong to heavy subsets of terminals in $\mathcal{X}$. We say that it is light if at least one of the two terminals belongs to a light subset, and the other terminal belongs to some subset in $\mathcal{X}$. Note that a demand pair (s,t) may be neither heavy nor light, for example, if s or t lie in $\mathcal{T} \setminus \bigcup_{X \in \mathcal{X}} X$. Let $\mathcal{M}_0$ be the set of all demand pairs that are neither heavy nor light. Then $|\mathcal{M}_0| \le 0.2|\mathcal{M}|$. We say that Case 1 happens if there are at least $0.1|\mathcal{M}|$ light demand pairs, and we say that Case 2 happens otherwise. Notice that in Case 2, at least $0.7|\mathcal{M}|$ of the demand pairs are heavy. In the next two sections we handle Case 1 and Case 2 separately.

6 Case 1: Light Demand Pairs

Let $\mathcal{M}^L \subseteq \mathcal{M}$ be the set of all light demand pairs. Recall that $|\mathcal{M}^L| \geq 0.1|\mathcal{M}|$. We assume w.l.o.g. that for every pair $(s,t) \in \mathcal{M}^L$, t belongs to a light set in $\mathcal{X}$. We let $S^L, T^L \subseteq \mathcal{T}$ be the sets of the source and the destination vertices of the demand pairs in $\mathcal{M}^L$, respectively. Let $\mathcal{L} \subseteq \mathcal{X}$ be the set of all light terminal subsets. Recall that we have assumed that every terminal participates in exactly one demand pair. If $(s,t) \in \mathcal{M}$, then we say that s is the mate of t, and t is the mate of s. The goal of this section is to prove the following theorem.

Theorem 6.1 *Let* $p^* = \Theta\left(\frac{\alpha_{\mathrm{WL}}|\mathcal{M}^L|}{\tau \log n}\right)$. *There is an efficient algorithm, that computes a routing of at least* $\min\left\{\Omega(p^*), \Omega\left(\frac{\Delta}{p^* \log n}\right)\right\}$ *demand pairs via node-disjoint paths in* G.

We first show that Theorem 6.1 concludes the proof of Theorem 4.3 for Case 1. Indeed, since $|\mathcal{M}^L| \geq 0.1|\mathcal{M}|$, we get that $p^* = \Theta\left(\frac{\alpha_{\mathrm{WL}}|\mathcal{M}|}{\tau \log n}\right) = \Theta\left(\frac{w^*|\mathcal{M}|}{W^{18/19}\log n \log k}\right) = \Theta\left(\frac{W^{1/19}}{\log n \log k}\right)$. Notice also that $\Omega\left(\frac{\Delta}{p^* \log n}\right) = \Omega\left(\frac{W^{2/19}\log k}{W^{1/19}}\right) = \Omega\left(W^{1/19}\log k\right)$. Therefore, the algorithm routes $\Omega\left(\frac{W^{1/19}}{\log n \log k}\right)$ demand pairs via node-disjoint paths. The rest of this section is devoted to proving Theorem 6.1.

Our first step is to compute a large subset $\mathcal{M}_0 \subseteq \mathcal{M}^L$ of light demand pairs, so that, if we denote by S_0 and T_0 the sets of the source and the destination vertices of the demand pairs in $\mathcal{M}_0$, then there is a set $\mathcal{Q}$ of $|\mathcal{M}_0|$ node-disjoint paths connecting the vertices of S_0 to a subset of vertices of T^L, that we denote by T'. Additionally, we ensure that every terminal set $X \in \mathcal{L}$, $|X \cap T'| \leq 1$, and $|X \cap T_0| \leq 1$. We note that the sets S_0 and T' do not necessarily form demand pairs. We will eventually route a subset of the pairs of $\mathcal{M}_0$.

Theorem 6.2 *There is an efficient algorithm to compute a subset* $\mathcal{M}_0 \subseteq \mathcal{M}^L$ *of* $\kappa_0 = \Theta\left(\frac{\alpha_{\mathrm{WL}}|\mathcal{M}^L|}{\tau}\right)$ *demand pairs, and a subset* $T' \subseteq T^L$ *of* κ_0 *terminals, such that, if we denote by* S_0 *and* T_0 *the sets of the source and the destination vertices of the demand pairs in* $\mathcal{M}_0$*, then:*

- *There is a set* $\mathcal{Q}$ *of* κ_0 *node-disjoint paths connecting the vertices of* S_0 *to the vertices of* T'*; and*
- *For each set* $X \in \mathcal{L}$, $|X \cap T'| \leq 1$, *and* $|X \cap T_0| \leq 1$.

Proof: Recall that S^L and T^L are the sets of the source and the destination vertices, respectively, of the demand pairs in $\mathcal{M}^L$.

We build the following directed flow network $\mathcal{N}$. Start with graph G, and bi-direct all its edges. For every light set $X \in \mathcal{L}$, add two vertices: s_X, connecting to every vertex $s \in S^L$, whose mate $t \in X$, and t_X, to which every vertex $t \in T^L \cap X$ is connected. Finally, we add a global source vertex s_0, that connects with directed edges to every vertex s_X for $X \in \mathcal{L}$, and a global destination vertex t_0, to which every vertex t_X with $X \in \mathcal{L}$ connects. The capacities of s_0 and t_0 are infinite, the capacities of all vertices in $\{s_X, t_X \mid X \in \mathcal{L}\}$ are τ, and all other vertex capacities are unit. Let $|S^L| = |T^L| = \tilde{k}$. We claim that there is an s_0–t_0 flow of value at least $\alpha_{\mathrm{WL}} \cdot \tilde{k}$ in $\mathcal{N}$. Assume otherwise. Then there is a set Y of vertices, whose total capacity is less than $\alpha_{\mathrm{WL}} \cdot \tilde{k}$, separating s_0 from t_0 in $\mathcal{N}$. Let A denote the subset of vertices of $\{s_X \mid X \in \mathcal{L}\}$ that lie in Y, and let $B = Y \cap \{t_X \mid X \in \mathcal{L}\}$. Assume that $|A| = a$, $|B| = b$, and assume for simplicity that $a \geq b$ (the other case is symmetric). We next build a set $\tilde{S} \subseteq S^L$ of source vertices as follows: for every set $X \in \mathcal{L}$ with $s_X \in A$, we add all vertices $s \in S^L$ whose mate belongs to X, to set $\tilde{S}$ (so $\tilde{S}$ contains all vertices $s \in S^L$, such that some edge originating at a vertex of A enters s). Let $\tilde{S}' = S^L \setminus \tilde{S}$. Since each cluster $X \in \mathcal{X}$ contains at most τ/w^* terminals, $|\tilde{S}'| \geq \tilde{k} - \frac{a\tau}{w^*}$.

Similarly, we let $\tilde{T} \subseteq T^L$ contain all terminals t, such that, if $t \in X$ for cluster $X \in \mathcal{L}$, then $t_X \in B$. Let $\tilde{T}' = T^L \setminus \tilde{T}$. Then $|\tilde{T}'| \geq \tilde{k} - \frac{b\tau}{w^*} \geq \tilde{k} - \frac{a\tau}{w^*}$, since we assumed that $a \geq b$. We discard terminals from $\tilde{S}'$ and $\tilde{T}'$ until $|\tilde{S}'| = |\tilde{T}'| = \left\lceil \tilde{k} - \frac{a\tau}{w^*} \right\rceil$ holds. (We note that $\tilde{k} - \frac{a\tau}{w^*} > 0$, since the total capacity of all vertices in A is at most $a\tau < \alpha_{\mathrm{WL}}\tilde{k} < w^*\tilde{k}$, as $\alpha_{\mathrm{WL}} = \frac{w^*}{512 \cdot \alpha_{\mathrm{AKR}} \cdot \log k}$.)

Let $Y' = Y \setminus (A \cup B)$. Then Y' is a subset of vertices of G, and moreover, $G \setminus Y'$ does not contain any path connecting a vertex of $\tilde{S}'$ to a vertex of $\tilde{T}'$. Indeed, if $G \setminus Y'$ contains a path P connecting some vertex $s \in \tilde{S}'$ to some vertex $t' \in \tilde{T}'$, then it is easy to see that path P can be extended to an s_0–t_0 path in $\mathcal{N} \setminus Y$. Notice that:

$$|Y'| < \alpha_{\mathrm{WL}} \cdot \tilde{k} - a\tau - b\tau \leq \alpha_{\mathrm{WL}} \cdot \tilde{k} - a\tau.$$

But from the well-linkedness of terminals, there must be a set of at least $\alpha_{\mathrm{WL}}|\tilde{S}'| \geq \alpha_{\mathrm{WL}}(\tilde{k} - a\tau/w^*)$ paths connecting the vertices of $\tilde{S}'$ to the vertices of $\tilde{T}'$. Recall that $\alpha_{\mathrm{WL}} = \frac{w^*}{512 \cdot \alpha_{\mathrm{AKR}} \cdot \log k}$, and so:

$$\alpha_{\mathrm{WL}} \left(\tilde{k} - \frac{a\tau}{w^*} \right) = \alpha_{\mathrm{WL}}\tilde{k} - \frac{a\tau}{512\alpha_{\mathrm{AKR}} \log k} > \alpha_{\mathrm{WL}}\tilde{k} - a\tau > |Y'|,$$

a contradiction. We conclude that there is an s_0–t_0 flow F of value at least $\alpha_{\mathrm{WL}}\tilde{k}$ in $\mathcal{N}$.

Let $\mathcal{N}'$ be a directed flow network defined exactly like $\mathcal{N}$, except that now we set the capacity of every vertex in $\{s_X, t_X \mid X \in \mathcal{L}\}$ to 1. By scaling the flow F down by factor τ, we obtain a valid s_0–t_0 flow in $\mathcal{N}'$ of value at least $\alpha_{\mathrm{WL}}\tilde{k}/\tau$. From the integrality of flow, there is an integral s_0–t_0 flow in $\mathcal{N}'$ of value $\left\lceil \alpha_{\mathrm{WL}}\tilde{k}/\tau \right\rceil$. This flow defines a collection $\mathcal{Q}$ of $\kappa_0 = \left\lceil \alpha_{\mathrm{WL}}\tilde{k}/\tau \right\rceil = \Omega(\alpha_{\mathrm{WL}}|\mathcal{M}^L|/\tau)$ node-disjoint paths in graph G, connecting some vertices of S^L to some vertices of T^L. We let $S_0 \subseteq S^L$ and $T' \subseteq T^L$ be the sets of vertices where the paths of $\mathcal{Q}'$ originate and terminate, respectively, and we let T_0 contain all mates of the source vertices in S_0. We then set $\mathcal{M}_0$ to be the set of the demand pairs in which the vertices of S_0 participate. It is easy to verify that for each set $X \in \mathcal{L}$, $|T' \cap X|, |T_0 \cap X| \leq 1$ from the definition of the network $\mathcal{N}'$. □

We assume that $|\mathcal{M}_0| \geq 1000$, as otherwise we can route a single demand pair, and obtain a solution routing $\Omega\left(\frac{\alpha_{\mathrm{WL}}|\mathcal{M}^L|}{\tau}\right)$ demand pairs.

Recall that every set $X \in \mathcal{X}$ contains at most one terminal from T_0. Since $|S_0| = |T_0|$, there is some set $X_0 \in T_0$, that contains exactly one terminal $t_0 \in T_0$, and at most one additional terminal from S_0. We will view t_0 as our "center" terminal, and we discard from $\mathcal{M}_0$ the demand pairs in which t_0 and the terminal in $S_0 \cap X_0$ (if it exists) participate.

The main tool in our algorithm for Case 1 is a crossbar, that we define below. Let $\Delta_1 = \lfloor \Delta/6 \rfloor$ and $\Delta_2 = \lfloor \Delta_1/3 \rfloor$. Given a shell $\mathcal{Z}(t) = (Z_1(t), \ldots, Z_{\Delta_1}(t))$ of depth Δ_1 around some terminal t, we will always denote by $D^*(t) = D(Z_{\Delta_1}(t))$, and by $\tilde{D}(t) = D(Z_{\Delta_2}(t))$. We will view the cycles $Z_1(t), \ldots, Z_{\Delta_2}(t)$ as the "inner" part of the shell $\mathcal{Z}(t)$. The crossbar is defined with respect to some large enough subset $\mathcal{M}^* \subseteq \mathcal{M}_0$ of demand pairs (see Figure 3).

Definition 6.1 *Suppose we are given a subset $\mathcal{M}^* \subseteq \mathcal{M}_0$ of demand pairs and an integer $p \geq 1$. Let S^* and T^* be the sets of all source and all destination vertices participating in the demand pairs of $\mathcal{M}^*$, respectively. A p-crossbar for $\mathcal{M}^*$ consists of:*

- *For each terminal $t \in T^* \cup \{t_0\}$, a shell $\mathcal{Z}(t)$ of depth Δ_1 around t, such that for all $t, t' \in T^* \cup \{t_0\}$, $D^*(t) \cap D^*(t') = \emptyset$, and for all $s \in S^*$ and $t' \in T^* \cup \{t_0\}$, $s \notin D^*(t')$; and*

- *For each $v \in S^* \cup T^*$, a collection $\mathcal{P}(v)$ of paths, such that:*
  - *For each $s \in S^*$, $\mathcal{P}(s)$ contains exactly one path, connecting s to a vertex of C_{t_0};*
  - *For each $t \in T^*$, $\mathcal{P}(t)$ contains exactly p paths, where each path connects a vertex of C_t to a vertex of C_{t_0}; and*
  - *All paths in $\mathcal{P} = \bigcup_{v \in S^* \cup T^*} \mathcal{P}(v)$ are node-disjoint from each other.*

In order to route a large subset of the demand pairs in $\mathcal{M}^*$, we need a crossbar with slightly stronger properties, that we call a *good crossbar*, and define below.

Definition 6.2 *Given a set $\mathcal{M}^* \subseteq \mathcal{M}_0$ of demand pairs, where S^*, T^* are the sets of the source and the destination vertices of the demand pairs in $\mathcal{M}^*$ respectively, and an integer $p \geq 1$, a p-crossbar $\left(\{\mathcal{Z}(t)\}_{t \in T^* \cup \{t_0\}}, \{\mathcal{P}(v)\}_{v \in S^* \cup T^*}\right)$ is a* good crossbar*, if the following additional properties hold:*

C1. For all $t \in T^*$ and all $v \in (S^* \cup T^*) \setminus \{t\}$, all paths in $\mathcal{P}(v)$ are disjoint from $\tilde{D}(t)$.

C2. For all $t \in T^*$, all paths in $\mathcal{P}(t)$ are monotone with respect to $(Z_1(t), \ldots, Z_{\Delta_2}(t))$. Also, for all $v \in S^* \cup T^*$, all paths in $\mathcal{P}(v)$ are monotone with respect to $(Z_1(t_0), \ldots, Z_{\Delta_2}(t_0))$.

C3. We can partition $Z_{\Delta_2}(t_0)$ into a collection $\Sigma = \{\sigma(v) \mid v \in S^* \cup T^*\}$ of $|S^*| + |T^*|$ disjoint segments, such that for all $v, v' \in S^* \cup T^*$ with $v \neq v'$, $\sigma(v) \cap \mathcal{P}(v') = \emptyset$.

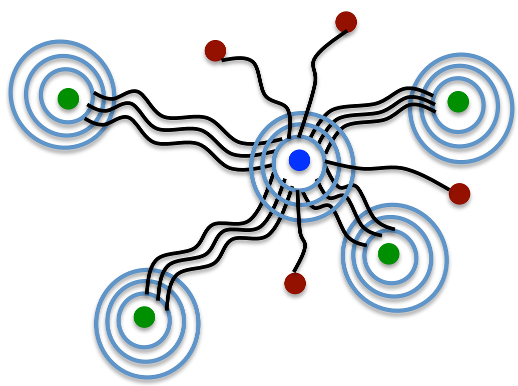


Figure 3: A crossbar. The center vertex t_0 is shown in blue, the vertices of S^* in red, and the vertices of T^* in green.

The following theorem shows that, given a p-crossbar in G, where p is large enough, we can route many demand pairs in $\mathcal{M}^*$.

Theorem 6.3 *Suppose we are given a subset $\mathcal{M}^* \subseteq \mathcal{M}_0$ of κ demand pairs, where S^* and T^* are the sets of all source and all destination vertices of the demand pairs in $\mathcal{M}^*$, respectively. Assume further that we are given a good p-crossbar $\left(\{\mathcal{Z}(t)\}_{t \in T^* \cup \{t_0\}}, \{\mathcal{P}(v)\}_{v \in S^* \cup T^*}\right)$ for $\mathcal{M}^*$, and let $q = \min\{\Delta_2, \lfloor (p-1)/2 \rfloor, \lceil \kappa/2 \rceil\}$. Then there is an efficient algorithm that routes at least q demand pairs in $\mathcal{M}^*$ via node-disjoint paths in G.*

Proof: We can assume without loss of generality that for every terminal $t \in T^*$, for every path $P \in \mathcal{P}(t)$, and for every $1 \leq j \leq \Delta_2$, $P \cap Z_j(t)$ consists of a single vertex. In order to see this, recall that for each such terminal t, path P and value j, $P \cap Z_j(t)$ is a path, from Property (C2). We contract each such path $P \cap Z_j(t)$ into a single vertex. We still maintain a good p-crossbar for $\mathcal{M}^*$ in the resulting graph, and any routing of a subset of the demand pairs in $\mathcal{M}^*$ in the new graph via node-disjoint paths immediately gives a similar routing of the same subset of the demand pairs in the original graph. Using a similar reasoning, we assume without loss of generality that for every path $P \in \bigcup_{v \in S^* \cup T^*} \mathcal{P}(v)$, for every $1 \leq j \leq \Delta_2$, $P \cap Z_j(t_0)$ is a single vertex.

Fix an arbitrary source vertex $s \in S^*$, and consider the unique path $P(s) \in \mathcal{P}(s)$. For all $1 \leq j \leq \Delta_2$, let v_j be the unique vertex in $Z_j(t_0) \cap P(s)$, and let e_j be the edge of $Z_j(t_0)$ incident to v_j, as we traverse $Z_j(t_0)$ starting from v_j in the clock-wise direction. Let R_j be the path obtained by deleting e_j from $Z_j(t_0)$. We view this path as directed in the counter-clock-wise direction along $Z_j(t_0)$, thinking of this as the left-to-right direction. Once we process every cycle $Z_j(t_0)$ for $1 \leq j \leq \Delta_2$ in this fashion, we obtain a collection $R_1, \ldots, R_{\Delta_2}$ of paths. Our routing will in fact only use the paths $R_1, \ldots, R_q$.

Let $\mathcal{P} = \bigcup_{v \in S^* \cup T^*} \mathcal{P}(v)$. For each $1 \leq j \leq \Delta_2$, and for each path $P \in \mathcal{P}$, let $u(P, j)$ be the unique vertex in $P \cap R_j$. The vertices $u(P, \Delta_2)$ define a natural left-to-right ordering $\mathcal{O}$ of the paths in $\mathcal{P}$: for $P, P' \in \mathcal{P}$, we denote $P \prec P'$ iff $u(P, \Delta_2)$ lies to the left of $u(P', \Delta_2)$ on R_{Δ_2}. Notice that, since the paths of $\mathcal{P}$ are monotone with respect to $Z_1(t_0), \ldots, Z_{\Delta_2}(t_0)$, for every pair $P, P' \in \mathcal{P}$ of paths with $P \prec P'$, for every $1 \leq j \leq \Delta_2$, $u(P, j)$ lies to the left of $u(P', j)$ on R_j. From Property (C3) of the crossbar, for each terminal $v \in S^* \cup T^*$, all paths in $\mathcal{P}(v)$ appear consecutively in the ordering $\mathcal{O}$. Therefore, we obtain a natural left-to-right ordering $\mathcal{O}'$ of the terminals: we say that terminal v lies to the left of terminal v' iff for all $P \in \mathcal{P}(v)$, $P' \in \mathcal{P}(v')$, $P \prec P'$.

We say that a demand pair $(s, t) \in \mathcal{M}^*$ is a left-to-right pair, if s appears before t in $\mathcal{O}'$, and we say that it is a right-to-left pair otherwise. At least $\lceil |\mathcal{M}^*|/2 \rceil$ of the pairs belong to one of these two types, and we assume w.l.o.g. that at least $\lceil |\mathcal{M}^*|/2 \rceil$ of the pairs are of the left-to-right type (otherwise we reverse the direction of the paths $\{R_j\}_{j=1}^{\Delta_2}$, and the orderings $\mathcal{O}, \mathcal{O}'$). We discard from $\mathcal{M}^*$ all right-to-left demand pairs, and we update the sets S^* and T^* accordingly. We discard additional demand pairs from $\mathcal{M}^*$ as needed, until $|\mathcal{M}^*| = q$ holds.

Consider the set S^* of the source vertices. We assume that $S^* = (s_1, \ldots, s_q)$, with the sources indexed in the order of their appearance in $\mathcal{O}'$. We define a set $\mathcal{P}^* = \left\{P_1^*, \ldots, P_q^*\right\}$ of node-disjoint paths, where for each $1 \leq i \leq q$, path P_i^* connects s_i to t_i. In order to do so, we first construct an initial set of paths, that we later modify.

The initial set $\mathcal{P}^*$ of paths is constructed as follows. Fix some $1 \leq i \leq q$. The initial path P_i^* is the concatenation of the segment of the unique path $P(s_i) \in \mathcal{P}(s_i)$ from s_i to $u(P, i)$, and the segment of R_i from $u(P, i)$ to the last (right-most) vertex of R_i (see Figure 4(a)).

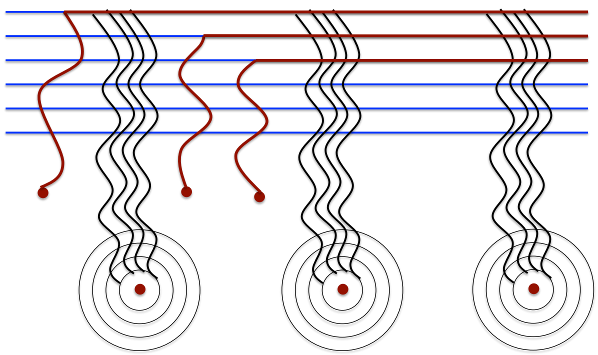
Clearly, all paths in $\mathcal{P}^*$ are node-disjoint, and each path P_i^* originates at s_i, but it does not terminate at t_i. We now modify the paths in $\mathcal{P}^*$ in order to fix this. We process the terminals $t_i \in T^*$ in their right-to-left order (the reverse order of their appearance in $\mathcal{O}'$). When we process terminal t_i, we will ensure that path P_i^* terminates at t_i, while all paths in $\mathcal{P}^*$ remain disjoint. Intuitively, in order to do so, we send the paths $P_{i+1}^*, \ldots, P_q^*$ "around" the terminal t_i, using the paths of $\mathcal{P}(t_i)$ and the q innermost cycles of the shell $\mathcal{Z}(t_i)$.

Formally, throughout the algorithm, we maintain the following invariants:

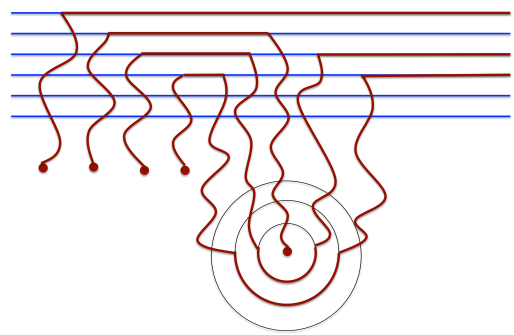
- The paths in $\mathcal{P}^* = \left\{P_1^*, \ldots, P_q^*\right\}$ always remain disjoint.
- Once terminal t_i is processed, in all subsequent iterations, path P_i^* connects s_i to t_i.
- Let t' be the last terminal processed, and for each $1 \leq i \leq q$, let v^i be the leftmost vertex of R_i that lies on any path in $\mathcal{P}(t')$. Let $P(s_i)$ be the unique path in $\mathcal{P}(s_i)$. Then if s_i lies to the left

of t' in $\mathcal{O}'$, the current path P_i^* contains the segment of R_i between $u(P(s_i), i)$ and v^i (possibly excluding v^i).

We now describe an iteration when the terminal t_i is processed. Let $\{Q_1, \ldots, Q_{2q+1}\}$ be any set of $2q+1$ distinct paths in $\mathcal{P}(t_i)$, indexed according to their ordering in $\mathcal{O}$ (recall that from our definition of q, $|\mathcal{P}(t_i)| = p \geq 2q+1$). We do not modify the paths $P_1^*, \ldots, P_{i-1}^*$ at this step. Path P_i^* is modified as follows. From our invariants, vertex $u(Q_{q+1}, i)$ belongs to P_i^*. We discard the last segment of P_i^* that starts at $u(Q_{q+1}, i)$, and then concatenate the remaining path with the segment of Q_{q+1} from its starting point $a \in C_{t_i}$ to $u(Q_{q+1}, i)$. In this way, we obtain a path originating at s_i and terminating at a. Since the sub-graph of G induced by $V(D(t_i))$ is connected, we can extend this path inside $D(t_i)$, so it now terminates at t_i.



(a) The initial paths, together with the inner shells of the terminals in T^*. Recall that for all $t \in T^*$ and $v \in S^* \cup T^*$ with $t \neq v$, all paths in $\mathcal{P}(v)$ are disjoint from the inner shell of t.



(b) Modifying the routing

Figure 4: Constructing the routing

For all $i' > i$, if $s_{i'}$ lies to the left of t_i, then path $P_{i'}^*$ is modified to "go around" t_i, using the paths $Q_{i'}, Q_{2q+2-i'}$ (that is, the i'th paths from the right and from the left), and the cycle $Z_{i'}(t_i)$, as follows. Let Q' be the segment of path $Q_{i'}$, starting from the unique vertex of $Q_{i'} \cap Z_{i'}(t_i)$ (that we denote by x), to vertex $u(Q_{i'}, i')$. Let Q'' be the segment of path $Q_{2q+2-i'}$, starting from the unique vertex of $Q_{2q+2-i'} \cap Z_{i'}(t_i)$ (that we denote by y), to vertex $u(Q_{2q+2-i'}, i')$. Consider now the two segments σ, σ' of $Z_{i'}(t_i)$, whose endpoints are x and y. Exactly one of these segments (say σ) is disjoint from the new path P_i^*. We remove from $P_{i'}^*$ the segment of $R_{i'}$ between $u(Q_{i'}, i')$ and $u(Q_{2q+2-i'}, i')$, and replace it with the concatenation of Q', σ and Q'' (see Figure 4(b)). It is easy to see that all invariants continue to hold. □

In order to complete the proof of Theorem 6.1, we prove the following theorem that allows us to construct a large enough crossbar.

Theorem 6.4 *There is an efficient algorithm that either finds a routing of $\Omega(\kappa_0)$ demand pairs of $\mathcal{M}_0$ via node-disjoint paths, or computes a subset $\mathcal{M}^* \subseteq \mathcal{M}_0$ of $\Omega(\kappa_0/\log n)$ demand pairs, and a good p-crossbar for $\mathcal{M}^*$, with $p = \Omega(\Delta/\kappa_0)$.*

Recall that $\kappa_0 = \Theta\left(\frac{\alpha_{\mathrm{WL}}|\mathcal{M}^L|}{\tau}\right)$. Letting $p^* = \frac{\kappa_0}{\log n} = \Omega\left(\frac{\alpha_{\mathrm{WL}}|\mathcal{M}^L|}{\tau \log n}\right)$, from Theorem 6.3, we can efficiently find a routing of at least $\min\left\{\Omega(p^*), \Omega\left(\frac{\Delta}{p^* \log n}\right)\right\}$ demand pairs, concluding the proof of Theorem 6.1. The rest of this section is dedicated to proving Theorem 6.4. We do so in two steps. In the first step, we either find a routing of $\Omega(\kappa_0)$ demand pairs in $\mathcal{M}_0$, or compute a large subset $\mathcal{M}_3 \subseteq \mathcal{M}_0$ of demand pairs, and a p-crossbar with respect to $\mathcal{M}_3$ (that may not be good). If the outcome of the first step is the former, then we terminate the algorithm and return the resulting routing. Otherwise, in the following step, we modify the set $\mathcal{M}_3$ and the crossbar in order to obtain the final set $\mathcal{M}^*$, and a good crossbar with respect to $\mathcal{M}^*$.

6.1 Constructing a Basic Crossbar

In this section we prove the following theorem.

Theorem 6.5 *There is an efficient algorithm that either computes a routing of $\Omega(\kappa_0)$ demand pairs in $\mathcal{M}_0$ via node-disjoint paths, or finds a subset $\mathcal{M}_3 \subseteq \mathcal{M}_0$ of $\kappa_3 = \Omega(\kappa_0)$ demand pairs, and a p_1-crossbar for $\mathcal{M}_3$, for $p_1 = \Omega(\Delta/\kappa_0)$.*

We construct the crossbar in two steps. Let S_0, T_0 be the sets of all source and all destination vertices corresponding to the demand pairs in $\mathcal{M}_0$, respectively. In the first step, we construct a shell $\mathcal{Z}(t)$ of depth Δ_1 around every terminal $t \in T_0 \cup \{t_0\}$, so that their corresponding discs $D^*(t)$ are disjoint. We then select a subset $\mathcal{M}_1 \subseteq \mathcal{M}_0$ of demand pairs, so that, if we denote by S_1 and T_1 the sets of all source and all destination vertices of the pairs in $\mathcal{M}_1$, respectively, then for all $s \in S_1$ and $t' \in T_1 \cup \{t_0\}$, $s \notin D^*(t')$. In the second step, we select a subset $\mathcal{M}_3 \subseteq \mathcal{M}_1$ of demand pairs, and complete the construction of the crossbar by computing the sets $\mathcal{P}(v)$ of paths for all $v \in \mathcal{T}(\mathcal{M}_3)$.

Step 1: the Shells. Consider any terminal $t \in T_0 \cup \{t_0\}$, and let $t' \neq t$ be any other terminal in $T_0 \cup \{t_0\}$. We let F_t be the face of the drawing of G on the sphere containing the image of t'. Notice that from Theorem 6.2, t and t' lie in different sets of $\mathcal{X}$, and so, from Theorem 5.8, $d(t,t') \geq 5\Delta$. In particular, for every vertex v on the boundary of F_t, $d_{\mathrm{GNC}}(v,t) > \Delta_1 + 1$. We can now construct a shell $\mathcal{Z}(t)$ of depth Δ_1 around t, with respect to F_t.

Additionally, for every terminal $t \in T'$, we construct a shell $\mathcal{Z}(t)$ of depth Δ_1 around t similarly, letting F_t be the face incident on any terminal $t' \in T' \setminus \{t\}$. This concludes the construction of the shells. We need the following lemma.

Lemma 6.6 *For all $t_1, t_2 \in T_0 \cup \{t_0\}$, if $t_1 \neq t_2$, then $D^*(t_1) \cap D^*(t_2) = \emptyset$. Moreover, for all $s \in S_0$, $s \notin D^*(t_0)$.*

Proof: Let $t_1, t_2 \in T_0 \cup \{t_0\}$ be any two distinct terminals, and assume for contradiction that $D^*(t_1) \cap D^*(t_2) \neq \emptyset$. We need the following observation.

Observation 6.7 *Either $D_{t_2} \subseteq D^*(t_1)$, or $D_{t_1} \subseteq D^*(t_2)$ must hold.*

Proof: If $D^*(t_2) \subseteq D^*(t_1)$, or $D^*(t_1) \subseteq D^*(t_2)$, then we are done, so assume that neither of these holds. From Observation 5.7, $Z_{\Delta_1}(t_1) \cap Z_{\Delta_1}(t_2) = \emptyset$. Since $D^*(t_1) \cap D^*(t_2) \neq \emptyset$, the only other possibility is that the union of the discs $D^*(t_1)$, $D^*(t_2)$ is the entire sphere, and the boundary of each disc is contained in the other disc. In particular, $Z_{\Delta_1}(t_2) \subseteq D^*(t_1)$. We claim that all vertices of C_{t_2} must lie in disc $D^*(t_1)$. Indeed, assume for contradiction that some vertex $v \in V(C_{t_2})$ does not lie in $D^*(t_1)$. Since all vertices of $Z_{\Delta_1}(t_2)$ lie in $D^*(t_1)$, there is a G-normal curve γ of length at most $\Delta/2 + \Delta_1 + 2$, connecting v to some vertex of $Z_{\Delta_1}(t_2)$, and this curve must contain some vertex $v' \in Z_{\Delta_1}(t_1)$. But then there is a G-normal curve of length at most $\Delta_1 + 1$ connecting v' to a vertex of C_{t_1}, so $d(t_1,t_2) \leq 2\Delta_1 + \Delta/2 + 2 < 5\Delta$, a contradiction. Therefore, all vertices of C_{t_2} appear in $D^*(t_1)$.

Let t'_1 be the terminal we have used to define the face F_{t_1}, so t'_1 lies outside $D^*(t_1)$. From Property (J5) of the shells, it is easy to see that $D_{t'_1}$ is disjoint from $D^*(t_1)$. If $D_{t_2} \not\subseteq D^*(t_1)$, then t'_1 must lie inside D_{t_2}. Moreover, since $D_{t'_1} \cap D^*(t_1) = \emptyset$, we get that $D_{t'_1} \subsetneq D_{t_2}$, contradicting the construction of enclosures. □

We assume w.l.o.g. that $D_{t_2} \subseteq D^*(t_1)$. Since for all $0 \leq j, j' \leq \Delta_1$, $Z_j(t_1) \cap Z_{j'}(t_2) = \emptyset$, there must be some index $1 \leq j \leq \Delta_1$, such that $D_{t_2} \subseteq D^\circ(Z_j(t_1)) \setminus D(Z_{j-1}(t_1))$. Let U_j be the set of all vertices of G lying in $D^\circ(Z_j(t_1)) \setminus D(Z_{j-1}(t_1))$, and let $\mathcal{R}_j$ be the set of all connected components in $G[U_j]$. Since $V(D_{t_2})$ induces a connected graph in G, there is some connected component $R \in \mathcal{R}_j$, such that $V(D_{t_2}) \subseteq V(R)$. From Theorem 5.5, there is a G-normal curve $\gamma(R)$ of length at most $2j + \Delta/2 + 1 \leq 2\Delta_1 + \Delta/2 + 1 < \Delta$, such that the disc $D(\gamma(R))$ contains R, and $D(\gamma(R)) \subseteq D^*(t_1)$.

Let t'_1 be the terminal we have used to define the face F_{t_1}, so t'_1 lies outside $D^*(t_1)$ and $D_{t'_1}$ is disjoint from $D^*(t_1)$. Curve $\gamma(R)$ then separates $D_{t'_1}$ from D_{t_2}, and clearly $D_{t'_1} \cap D_{t_2} = \emptyset$. But from our construction of enclosures, there is a set of Δ node-disjoint paths connecting $V(C_{t_2})$ to $V(C_{t'_1})$, all of which must intersect $\gamma(R)$, contradicting the fact that $\gamma(R)$ contains fewer than Δ vertices.

Assume now for contradiction that some vertex $s \in S_0$ lies in $D^*(t_0)$. Recall that s must lie in some set of $\mathcal{X} \setminus \{X_0\}$ (where X_0 is the set containing t_0), and so $d(s, t_0) \geq 5\Delta$. Therefore, for all $1 \leq j \leq \Delta_1$, $V(C_s) \cap V(Z_j(t_0)) = \emptyset$ must hold. Moreover, $D^*(t_0) \not\subseteq D_s$, since otherwise, from the definition of enclosures, $D_{t_0} = D_s$ and $d(t_0, s) = 1$, a contradiction. Since $D^*(t_0) \cap D_s \neq \emptyset$, we get that $D_s \subseteq D^*(t_0)$. The rest of the proof follows the same reasoning as before. □

We say that a demand pair $(s, t) \in \mathcal{M}_0$ is a type-1 pair if $s \in D^*(t)$, and we say that it is a type-2 demand pair otherwise. Notice that we can route all type-1 demand pairs via node-disjoint paths, where each pair (s, t) is routed in $G[V(D^*(t))]$ (since from Observation 5.6, this graph is connected). Therefore, if at least half the demand pairs in $\mathcal{M}_0$ are type-1 pairs, then we obtain a routing of $\Omega(\kappa_0)$ demand pairs and terminate the algorithm. We assume from now on that at least $\kappa_0/2$ demand pairs in $\mathcal{M}_0$ are type-2 demand pairs. We next build a directed conflict graph H, that contains a vertex $v(s, t)$ for each type-2 demand pair $(s, t) \in \mathcal{M}_0$. There is a directed edge from $v(s, t)$ to $v(s', t')$ iff $s \in D^*(t')$. Since the discs $\{D^*(t)\}_{t \in T_0}$ are mutually disjoint, the out-degree of every vertex of H is at most 1, and the total average degree of every vertex in H is at most 2. Since $|V(H)| \geq \kappa_0/2$, using standard techniques, we can find an independent set I of $\kappa_1 = \Omega(\kappa_0)$ vertices in H. We let $\mathcal{M}_1 = \{(s, t) \in \mathcal{M}_0 \mid v(s, t) \in I\}$. Let S_1 and T_1 be the sets of source and destination vertices, respectively, of the demand pairs in $\mathcal{M}_1$. Then for all $t, t' \in T_1$, $D^*(t) \cap D^*(t') = \emptyset$, and for all $s \in S_1, t' \in T_1$, $s \notin D^*(t')$.

Step 2: the Paths. Recall that $\kappa_1 = \Omega(\kappa_0) = \Theta\left(\frac{\alpha_{\mathrm{WL}}|\mathcal{M}|}{\tau}\right) = \Theta\left(\frac{W^{1/19}}{\log k}\right)$, while $\Delta = \Theta(W^{2/19})$. We will assume throughout that $\kappa_1 < \Delta/12$, as otherwise κ_0 is bounded by some constant, and routing a single demand pair is sufficient. The following lemma will be used in order to compute the set $\mathcal{M}_3 \subseteq \mathcal{M}_1$ of the demand pairs.

Lemma 6.8 *There is a set $\mathcal{P}^S$ of $\left\lfloor \frac{\kappa_1}{2} \right\rfloor - 1$ node-disjoint paths, connecting vertices of S_1 to vertices of C_{t_0}.*

Proof: Assume otherwise. Then there is a set Y of at most $\left\lfloor \frac{\kappa_1}{2} \right\rfloor - 2$ vertices, such that $G \setminus Y$ contains no path from a vertex of $S_1 \setminus Y$ to a vertex of $V(C_{t_0}) \setminus Y$.

Let $\mathcal{Q}' \subseteq \mathcal{Q}$ be a subset of the paths, computed in Theorem 6.2, that connect every terminal $s \in S_1$ to some terminal of T', so $|\mathcal{Q}'| = \kappa_1$. Let $\mathcal{Q}'' \subseteq \mathcal{Q}'$ be the set of paths that contain no vertices of Y, so $|\mathcal{Q}''| \geq \kappa_1/2 + 2$. Recall that the terminals of T' all belong to distinct sets of $\mathcal{X}$, and so the discs

$\{D_{t'}\}_{t' \in T'}$ are mutually disjoint. Moreover, there is at most one terminal t'' in $T' \cap X_0$, where $X_0 \in \mathcal{X}$ is the set containing t_0. Then there must be at least one path $Q \in \mathcal{Q}''$, such that, if we denote by $s \in S_1$, $t' \in T'$ its two endpoints, then $t' \notin X_0$, and $D_{t'}$ contains no vertex of Y. We now show that there is a path connecting s to a vertex of C_{t_0} in $G \setminus Y$, leading to a contradiction.

From the construction of enclosures, there is a set $\mathcal{P}(t', t_0)$ of Δ node-disjoint paths, connecting the vertices of C_{t_0} to the vertices of $C_{t'}$. Since we have assumed that $\kappa_1 < \Delta/12$, at least one such path, say P, is disjoint from Y. Let v' be the endpoint of P lying on $C_{t'}$. Assume first that $s \notin D_{t'}$. Then Q must contain a vertex of $C_{t'}$, that we denote by v. Since $G[V(D_{t'})]$ is connected, there is a path $P' \subseteq G[V(D_{t'})]$ connecting v to v', and this path is disjoint from Y. The union of Q, P and P' then contains a path connecting s to a vertex of $V(C_{t_0})$, which is disjoint from Y, a contradiction.

Assume now that $s \in D_{t'}$. Then there is a path $P' \subseteq G[V(D_{t'})]$, connecting s to v'. The union of P and P' then contains a path connecting s to a vertex of $V(C_{t_0})$, which is disjoint from Y, a contradiction. □

Let $S_2 \subseteq S_1$ be the set of the source vertices where the paths of $\mathcal{P}^S$ originate, let $T_2 \subseteq T_1$ be the set of their mates, and let $\mathcal{M}_2 \subseteq \mathcal{M}_1$ be the set of the demand pairs in which the vertices of S_2 participate. Notice that from our choice of t_0, there is at most one demand pair $(s, t) \in \mathcal{M}_1$, where t belongs to the subset $X_0 \in \mathcal{X}$ containing t_0. If such a pair belongs to $\mathcal{M}_1$, then we discard it from $\mathcal{M}_1$, and update S_1 and T_1 accordingly. Let $\kappa_2 = |\mathcal{M}_2|$, so $\kappa_2 \geq \kappa_1/2 - 3 \geq \Omega(\kappa_0)$. We now use the following theorem to compute the subset $\mathcal{M}_3$ of the demand pairs and construct the corresponding paths for the crossbar.

Theorem 6.9 *There is an efficient algorithm to compute a subset $\mathcal{M}_3 \subseteq \mathcal{M}_2$ of $\lfloor \kappa_2/2 \rfloor$ demand pairs, and a p-crossbar for $\mathcal{M}_3$, where $p = \Omega(\Delta/\kappa_2)$.*

Proof: Consider any terminal $t \in T_2$, and let $X_0 \in \mathcal{X}$ be the set containing t_0. Since $t \notin X_0$, $d(t, t_0) \geq 5\Delta$, and in particular, $D_t \cap D_{t_0} = \emptyset$. Therefore, there is a set $\tilde{\mathcal{P}}(t)$ of Δ node-disjoint paths, connecting the vertices of $V(C_t)$ to the vertices of $V(C_{t_0})$.

We construct a directed flow network $\mathcal{N}$, by starting from G, and bi-directing each of its edges. We then introduce several new vertices: a global source vertex s^* of infinite capacity; an additional vertex vertex $\tilde{s}$ of capacity $\lfloor \kappa_2/2 \rfloor$, and, for each terminal $t \in T_2$ a vertex s_t of capacity $\lfloor \Delta/(2\kappa_2) \rfloor$. We connect s^* to $\tilde{s}$, and to every vertex in $\{s_t\}_{t \in T_2}$. Vertex $\tilde{s}$ in turn connects to every vertex $s \in S_2$, and for each $t \in T_2$, vertex s_t connects to every vertex in $V(C_t)$. Finally, we introduce a global destination vertex t^* of infinite capacity, and connect every vertex in $V(C_{t_0})$ to it. All vertices whose capacities have not been set so far have capacity 1.

Let B be the total capacity of all vertices in $\{\tilde{s}\} \cup \{s_t\}_{t \in T_2}$. It is easy to see that there is a flow of value at least B in the resulting network: we send $\frac{1}{2}$ flow unit along each path in set $\mathcal{P}^S$ (discarding one path if $|S_2|$ is odd), and $\frac{1}{2\kappa_2}$ flow units on each path in $\bigcup_{t \in T_2} \tilde{\mathcal{P}}(t)$ (if the flow through some vertex s_t is too high due to the rounding of the capacities down, we simply lower the flow on one of the corresponding paths in $\tilde{\mathcal{P}}(t)$). Since the paths in $\mathcal{P}^S$ are node-disjoint, the total flow on these paths causes congestion at most $\frac{1}{2}$ on the vertices whose capacity is 1. Since for each $t \in T_2$, the paths in $\tilde{\mathcal{P}}(t)$ are node-disjoint, the total congestion caused by the flow on paths in $\bigcup_{t \in T_2} \tilde{\mathcal{P}}(t)$ on vertices whose capacity is 1 is at most $1/2$. Therefore, we obtain a valid s^*-t^* flow of value B. From the integrality of flow, there is an integral flow of the same value.

This flow gives a collection $\mathcal{P}^1$ of $\lfloor \kappa_2/2 \rfloor$ paths connecting some vertices of S_2 to some vertices of $V(C_{t_0})$, and, for each $t \in T_2$, a collection $\mathcal{P}'(t)$ of $\left\lfloor \frac{\Delta}{2\kappa_2} \right\rfloor$ paths connecting the vertices of $V(C_t)$ to the vertices of $V(C_{t_0})$, such that all paths in $\mathcal{P}^1 \cup \{\mathcal{P}'(t) \mid t \in T_2\}$ are mutually node-disjoint. We let $S_3 \subseteq S_2$ be the set of the source vertices where the paths of $\mathcal{P}^1$ originate. For each $s \in S_3$, let $P(s) \in \mathcal{P}^1$ be the unique path originating at s, and let $\mathcal{P}(s) = \{P(s)\}$. Let $\mathcal{M}_3 \subseteq \mathcal{M}_2$ contain all demand pairs whose source belongs to S_3, and let $T_3 \subseteq T_2$ be the set of the corresponding destination vertices. Then $|\mathcal{M}_3| = \lfloor \kappa_2/2 \rfloor$, and all paths in set $\mathcal{P}' = \bigcup_{v \in S_3 \cup T_3} \mathcal{P}(v)$ are mutually disjoint. For each terminal $t \in T_3$, set $\mathcal{P}(t)$ contains $\Omega(\Delta/\kappa_2)$ paths, as required. □

This concludes the proof of Theorem 6.5. Since $p_1 = \Omega(\Delta/\kappa_0)$ and $\kappa_3 \le \kappa_0$, we will assume without loss of generality that $p_1 \cdot \kappa_3 < \Delta_2/24$ (otherwise we can lower the value of p_1 until the inequality holds and discard the appropriate number of paths from sets $\mathcal{P}(t)$ for $t \in T_3 \cup \{t_0\}$).

6.2 Constructing a Good Crossbar

In this section, we complete the proof of Theorem 6.4, by modifying the basic crossbar constructed in the previous section, in order to turn it into a good crossbar. Let $\left(\bigcup_{t \in T_3 \cup \{t_0\}} \mathcal{Z}(t), \bigcup_{v \in S_3 \cup T_3} \mathcal{P}(v)\right)$ be the crossbar constructed in the previous section.

For every terminal $t \in T_3$, let $A_t \subseteq V(C_t), B_t \subseteq V(C_{t_0})$ be the sets of p_1 vertices each, where the paths of $\mathcal{P}(t)$ originate and terminate, respectively. For a source vertex $s \in S_3$, we let A_s contain a single vertex s, and B_s contain the vertex of $V(C_{t_0})$ where the unique path in $\mathcal{P}(s)$ terminates. Let $A = \bigcup_{v \in S_3 \cup T_3} A_v$ and $B = \bigcup_{v \in S_3 \cup T_3} B_v$. Then $|A| = |B| = |\mathcal{P}| = |\mathcal{M}_3|(p_1 + 1) < \Delta_2/6$ from our assumption.

Definition 6.3 *Given two equal-sized disjoint subsets U_1, U_2 of vertices of G, a U_1–U_2 linkage is a set of $|U_1|$ node-disjoint paths, connecting the vertices of U_1 to the vertices of U_2.*

The following observation is immediate from the definition of a crossbar.

Observation 6.10 *Let $\mathcal{P}'$ be any A–B linkage in G. Then $\left(\bigcup_{t \in T_3 \cup \{t_0\}} \mathcal{Z}(t), \mathcal{P}'\right)$ is a p_1-crossbar for $\mathcal{M}_3$.*

Our algorithm consists of three steps. In the first step, we re-route the paths in $\mathcal{P}$, so that they become disjoint from the relevant inner shells, ensuring Property (C1). In the following step, the paths in $\mathcal{P}$ are further re-routed to ensure their monotonicity with respect to the relevant inner shells, obtaining Property (C2). The set $\mathcal{M}_3$ of the demand pairs remains unchanged in these two steps. In the last step, we carefully select a final subset $\mathcal{M}^* \subseteq \mathcal{M}_3$ of the demand pairs to ensure Property (C3). We discard some paths from set $\mathcal{P}$, but the paths themselves are not modified at this step.

Step 1: Disjointness of Paths from Inner Shells

The goal of this step is to modify the paths in set $\mathcal{P}$, in order to ensure Property (C1). In fact, we will ensure a slightly stronger property that we will use in subsequent steps.

Suppose we are given any A–B linkage $\mathcal{P}'$. For every terminal $t \in T_3$, we let $\mathcal{P}'(t) \subseteq \mathcal{P}'$ be the set of paths originating from the vertices of A_t, and for every source vertex $s \in S_3$, we let $\mathcal{P}'(s)$ contain the unique path of $\mathcal{P}'$ originating at s. We will always view the paths in $\mathcal{P}'$ as directed from A to B. Consider now any path $P \in \mathcal{P}'$. Clearly, path P has to cross $Z_{\Delta_1}(t_0)$. Let v_P be the last vertex on P that lies on $Z_{\Delta_1}(t_0)$, and let e_P be the edge immediately preceding v_P on P. Let v'_P be the other endpoint of edge e_P. Then $P \setminus \{e_P\}$ consists of two disjoint paths, that we denote by P_1 and P_2, respectively, where P_1 starts at a vertex of $V(C_t)$ and terminates at v'_P, and P_2 starts at v_P and terminates at a vertex of $V(C_{t_0})$.

We then denote $A' = \{v'_P \mid P \in \mathcal{P}'\}$, $B' = \{v_P \mid P \in \mathcal{P}'\}$, and $\tilde{E} = \{e_P \mid P \in \mathcal{P}'\}$. We also denote $\mathcal{P}'_1 = \{P_1 \mid P \in \mathcal{P}'\}$ and $\mathcal{P}'_2 = \{P_2 \mid P \in \mathcal{P}'\}$. Notice that $\mathcal{P}'_1$ is an A–A' linkage, and $\mathcal{P}'_2$ is a B'–B linkage. Observe that for every terminal $t \in T_3 \cup \{t_0\}$, all vertices of $A' \cup B'$ must lie outside $D(Z_{\Delta_1-1}(t))$. Also, from our definitions, every path in $\mathcal{P}'_2$ is contained in $D^*(t_0)$.

Remark 6.1 *We note that the definitions of the sets A', B' of vertices and the set $\tilde{E}$ of edges depend on the A–B linkage $\mathcal{P}'$. When we modify the linkage, these sets may change as well.*

We are now ready to define good A–B linkages.

Definition 6.4 *We say that an A–B linkage $\mathcal{P}'$ is a* good linkage*, iff:*

- *All paths in $\mathcal{P}'_1$ are disjoint from $\tilde{D}(t_0)$; and*
- *For every $t \in T_3$, if some path $P \in \mathcal{P}'_1 \cup \mathcal{P}'_2$ intersects $\tilde{D}(t)$, then $P \in \mathcal{P}'_1$, and it originates at a vertex of A_t.*

Notice that if $\mathcal{P}'$ is a good A–B linkage, then $\left(\bigcup_{t \in T_3 \cup \{t_0\}} \mathcal{Z}(t), \mathcal{P}'\right)$ is a p_1-crossbar for $\mathcal{M}_3$ that has Property (C1). The goal of this step is to prove the following theorem.

Theorem 6.11 *There is an efficient algorithm that computes a good A–B linkage $\mathcal{P}'$.*

The remainder of this step focuses on the proof of Theorem 6.11. In order to prove the theorem, we start with the A–B linkage $\mathcal{P}$, given by the basic crossbar that we have computed in the previous section, and iteratively modify it. Let E_1 be the set of all edges that belong to cycles $\bigcup_{t \in T_3 \cup \{t_0\}} \bigcup_{j=\Delta_2+1}^{\Delta_1} Z_j(t)$. Given any set $\mathcal{Q}$ of node-disjoint paths in G, let $c(\mathcal{Q})$ be the total number of all edges lying on the paths in $\mathcal{Q}$ that do not belong to E_1, so $c(\mathcal{Q}) = \sum_{P \in \mathcal{Q}} |E(P) \setminus E_1|$. The following lemma is key to proving Theorem 6.11.

Lemma 6.12 *Let $\mathcal{P}$ be any A–B linkage, and assume that it is not a good linkage. Then there is an efficient algorithm to compute an A–B linkage $\mathcal{P}'$ with $c(\mathcal{P}') < c(\mathcal{P})$.*

In order to complete the proof of Theorem 6.11, we start with the original A–B linkage $\mathcal{P}$. While the current linkage $\mathcal{P}$ is not a good one, we apply Lemma 6.12 to it, to obtain another A–B linkage $\mathcal{P}'$ with $c(\mathcal{P}') < c(\mathcal{P})$. The algorithm terminates when we obtain a good A–B linkage. Since the values $c(\mathcal{P})$ are integers bounded by $|E(G)|$, and they decrease by at least 1 in every iteration, the algorithm is guaranteed to terminate after $|E(G)|$ iterations. It now remains to prove Lemma 6.12.

Proof of Lemma 6.12. Let $\mathcal{P}$ be any A–B linkage, and assume that it is not a good linkage. Let $\mathcal{Q} = \mathcal{P}_1 \cup \mathcal{P}_2$, so $\mathcal{Q}$ is an $(A \cup B')$–$(A' \cup B)$ linkage in $G \setminus \tilde{E}$, and $|\mathcal{Q}| = 2|\mathcal{M}_3|(p_1+1) < 6\kappa_3 p_1 < \Delta_2/3$ from our assumption. We will use the following simple observation.

Observation 6.13 *Let $\mathcal{Q}'$ be any $(A \cup B')$–$(A' \cup B)$ linkage in $G \setminus \tilde{E}$, and let G' be the graph obtained by the union of the paths in $\mathcal{Q}'$ and the edges of $\tilde{E}$. Then G' contains an A–B linkage $\mathcal{P}'$. Moreover, if $c(\mathcal{Q}') < c(\mathcal{Q})$, then $c(\mathcal{P}') < c(\mathcal{P})$.*

Proof: We construct the following auxiliary directed graph $\tilde{G}$. The vertices of $\tilde{G}$ are $A \cup A' \cup B \cup B'$. The edges are defined as follows. First, for every path $Q \in \mathcal{Q}'$, that originates at a vertex $v \in A \cup B'$ and terminates at a vertex $u \in A' \cup B$, we add a directed edge from v to u. Notice that since $\mathcal{Q}'$ is an $(A \cup B')$–$(A' \cup B)$ linkage, this gives a directed matching from the vertices of $A \cup B'$ to the vertices of

$A' \cup B$. Finally, for every edge $e_P = (v'_P, v_P) \in \tilde{E}$, we add a directed edge from v'_P to v_P to $\tilde{G}$. Then in the resulting graph $\tilde{G}$, every vertex of A has in-degree 0 and out-degree 1; every vertex of B has out-degree 0 and in-degree 1, and every vertex of $A' \cup B'$ has in-degree and out-degree 1. Therefore, graph $\tilde{G}$ is a collection of directed paths and cycles, and it must contain an A–B linkage $\tilde{\mathcal{P}}$. By replacing, for every path $P \in \tilde{\mathcal{P}}$, the edges representing the paths of $\mathcal{Q}'$ on P with the corresponding paths, we obtain an A–B linkage $\mathcal{P}'$. Since the edges of $\tilde{E}$ participate in the paths in $\mathcal{P}$, and the paths of $\mathcal{P}'$ are contained in graph G', it is immediate to verify that $c(\mathcal{P}') < c(\mathcal{P})$. □

From Observation 6.13, it is now enough to construct an $(A \cup B')$–$(A' \cup B)$ linkage $\mathcal{Q}'$ in $G \setminus \tilde{E}$, with $c(\mathcal{Q}') < c(\mathcal{Q})$. The following technical claim is central to achieving this.

Claim 6.14 *Let H be any planar graph embedded in the plane, such that H is a union of a set $\mathcal{Z} = \{Z_1, \ldots, Z_h\}$ of disjoint cycles with $D(Z_1) \subsetneq D(Z_2) \subsetneq \ldots \subsetneq D(Z_h)$, and a set $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$ of $r < h$ node-disjoint paths. Assume that for $i \in \{1, 2\}$, the paths of $\mathcal{Q}_i$ originate at a set U_i of vertices, and terminate at a set U'_i of vertices. Assume further that the vertices of U_1 lie in $D^\circ(Z_1)$, while the vertices of $U'_1 \cup U_2 \cup U'_2$ lie outside $D(Z_h)$. Let H' be the graph obtained from H by deleting, for every path $Q \in \mathcal{Q}_2$, every edge and vertex of Q contained in $D^\circ(Z_1)$. Then there is a set $\mathcal{Q}'$ of r node-disjoint paths, connecting the vertices of $U_1 \cup U_2$ to the vertices of $U'_1 \cup U'_2$ in H'.*

Proof: Let $U = U_1 \cup U_2$ and $U' = U'_1 \cup U'_2$. Assume otherwise. From Menger's theorem, there is a set Y of at most $r - 1$ vertices, such that in $H' \setminus Y$, no path connects a vertex of $U \setminus Y$ to a vertex of $U' \setminus Y$. Let $|\mathcal{Q}_1| = r_1$ and $|\mathcal{Q}_2| = r_2$.

Observe that all paths of $\mathcal{Q}_1$ are contained in H', so for each path $P \in \mathcal{Q}_1$, there must be a distinct vertex $y_P \in Y$ lying on P. Let $Y_1 \subseteq Y$ be the set of all such vertices, so $|Y_1| = r_1$.

Let $\mathcal{Q}^* \subseteq \mathcal{Q}_2$ be the set of all paths that are disjoint from $D^\circ(Z_1)$, and denote $|\mathcal{Q}^*| = r^*$. Then every path $P \in \mathcal{Q}^*$ is contained in H', and as before, Y must contain a distinct vertex y_P lying on P. Let Y_2 be the set of all such vertices $\{y_P \mid P \in \mathcal{Q}^*\}$. Since the paths in $\mathcal{Q}$ are disjoint, $Y_1 \cap Y_2 = \emptyset$, and $|Y_1| + |Y_2| = r_1 + r^*$. Let $Y_3 = Y \setminus (Y_1 \cup Y_2)$, and let $\mathcal{Q}'' = \mathcal{Q}_2 \setminus \mathcal{Q}^*$. Then $|Y_3| \le |\mathcal{Q}''| - 1$, and due to the disjointness of the paths in $\mathcal{Q}$, every path in $\mathcal{Q}''$ is disjoint from $Y_1 \cup Y_2$. Therefore, there is some path $Q \in \mathcal{Q}''$, such that $Q \cap Y = \emptyset$. Recall that $Q \cap D^\circ(Z_1) \neq \emptyset$, and so Q must intersect Z_1. Let v', v'' be the first and the last vertices of Q that lie on Z_1. Recall that the endpoints of Q lie outside $D(Z_h)$. Let Q' be the segment of Q between its first endpoint and v', and let Q'' be the segment of Q between v'' and its last endpoint. Then both Q' and Q'' are contained in H', and each of these paths intersects every cycle in $\mathcal{Z}$. The number of these cycles is $h > r$ from our assumption. Therefore, there is some $1 \le j \le h$, such that Z_j is disjoint from Y. By combining Z_j with Q' and Q'', we obtain a path connecting a vertex of $U \setminus Y$ to a vertex of $U' \setminus Y$ in $H' \setminus Y$, a contradiction. □

We are now ready to complete the proof of Lemma 6.12. Let $\mathcal{P}$ be the given A–B linkage, and assume that it is not a good linkage.

Assume first that some path $P \in \mathcal{P}_1$ has a non-empty intersection with $\tilde{D}(t_0)$. We let $\mathcal{Q}_1 = \mathcal{P}_2$, $\mathcal{Q}_2 = \mathcal{P}_1$, and $\mathcal{Z} = (Z_{\Delta_2+1}(t_0), \ldots, Z_{\Delta_1-2}(t_0))$. Observe that $|\mathcal{Z}| = \Delta_1 - \Delta_2 - 2 \ge \Delta_2 > 24 p_1 \kappa_3 > |\mathcal{Q}|$, the paths in $\mathcal{Q}_1$ originate at the vertices of C_{t_0}, that lie in $D^\circ(Z_{\Delta_2+1})$, and terminate at the vertices of B', that lie outside $D(Z_{\Delta_1-2}(t_0))$ (after we reverse them), while the paths of $\mathcal{Q}_2$ originate and terminate outside $D(Z_{\Delta_1-2}(t_0))$ (we also reverse them). From our assumption, at least one path in $\mathcal{Q}_2$ intersects $D^\circ(Z_{\Delta_2+1})$. Let H be the graph obtained by the union of the cycles in $\mathcal{Z}$ and the paths in $\mathcal{Q}$. Since the edges of $\tilde{E}$ cannot lie on the cycles of $\mathcal{Z}$, $H \subseteq G \setminus \tilde{E}$. We can now apply Claim 6.14 to obtain a new $(A \cup B')$–$(A' \cup B)$ linkage $\mathcal{Q}'$ in graph $G \setminus \tilde{E}$. Moreover, since we delete all edges lying on the paths of $\mathcal{Q}_2$ that belong to $D^\circ(Z_{\Delta_2+1}(t_0))$, it is easy to verify that $c(\mathcal{Q}') < c(\mathcal{Q})$.

Assume now that there is some path $P \in \mathcal{Q}$, and some terminal $t \in T_3$, such that P intersects $\tilde{D}(t)$, but it does not originate at a vertex of A_t. Let $\mathcal{Q}_1 \subseteq \mathcal{P}_1$ be the set of all paths originating from the vertices of A_t, and let $\mathcal{Q}_2 = \mathcal{Q} \setminus \mathcal{Q}_1$. It is easy to verify that we can apply Claim 6.14 as before, to obtain a new $(A \cup B')$–$(A' \cup B)$ linkage $\mathcal{Q}'$ in graph $G \setminus \tilde{E}$, with $c(\mathcal{Q}') < c(\mathcal{Q})$.

Using Observation 6.13, we can now obtain an A–B linkage $\mathcal{P}'$ with $c(\mathcal{P}') < c(\mathcal{P})$. □

Step 2: Monotonicity with Respect to Inner Shells

In this step we further modify the paths in set $\mathcal{P}$, in order to ensure Property (C2). As before, given a good A–B linkage $\mathcal{P}'$, for every terminal $t \in T_3$ we denote by $\mathcal{P}'(t) \subseteq \mathcal{P}'$ the set of paths originating at the vertices of A_t, and for each $s \in S_3$, we let $\mathcal{P}'(s) \subseteq \mathcal{P}'$ contain the unique path originating at s. We define the sets A', B' of vertices, the set $\tilde{E}$ of edges, and the sets $\mathcal{P}'_1, \mathcal{P}'_2$ of paths with respect to $\mathcal{P}'$ exactly as before.

Definition 6.5 *We say that a good A–B linkage $\mathcal{P}'$ is perfect if all paths in $\mathcal{P}'$ are monotone with respect to $Z_1(t_0), \ldots, Z_{\Delta_2}(t_0)$, and for all $t \in T_3$, all paths in $\mathcal{P}'(t)$ are monotone with respect to $Z_1(t), \ldots, Z_{\Delta_2}(t)$.*

In this step we prove the following theorem, that immediately gives a p_1-crossbar for $\mathcal{M}_3$ with Properties (C1) and (C2).

Theorem 6.15 *There is an efficient algorithm, that, given a good A–B linkage $\mathcal{P}$, computes a perfect A–B linkage $\mathcal{P}'$.*

Proof: Our first step is to re-route the paths in $\mathcal{P}_2$, so that they become monotone with respect to $Z_1(t_0), \ldots, Z_{\Delta_2}(t_0)$. Recall that from our definition, the paths of $\mathcal{P}_2$ originate from the set $B' \subseteq V(Z_{\Delta_1}(t_0))$ of vertices, terminate at the set $B \subseteq V(C_{t_0})$ of vertices, and they are internally disjoint from $V(Z_{\Delta_1}(t_0)) \cup V(C_{t_0})$. Let $\mathcal{Z} = \{Z_1(t_0), \ldots, Z_{\Delta_2}(t_0)\}$, $C = C_{t_0}$, and $Y = Z_{\Delta_1}(t_0)$. Recall that from the construction of the shells, $\mathcal{Z}$ is a set of Δ_2 tight concentric cycles around C. Let H be the graph obtained by the union of the cycles in $\mathcal{Z}$ and the set $\mathcal{P}_2$ of paths. We use Theorem 2.5 to obtain a new B'–B linkage $\mathcal{P}'_2$ in H, such that the paths in $\mathcal{P}'_2$ are monotone with respect to $Z_1(t_0), \ldots, Z_{\Delta_2}(t_0)$. Moreover, since the paths in $\mathcal{P}_1$ are disjoint from the graph H, the paths in $\mathcal{P}'_2 \cup \mathcal{P}_1$ remain disjoint. Let $\tilde{\mathcal{P}}$ be the new A–B linkage, obtained by concatenating the paths in $\mathcal{P}_1$, the edges of $\tilde{E}$, and the paths in $\mathcal{P}'_2$. Observe that $\tilde{\mathcal{P}}$ remains a good linkage.

For each terminal $t \in T_3$, let $\tilde{\mathcal{P}}(t) \subseteq \tilde{\mathcal{P}}$ be the set of paths originating from the vertices of A_t. For every terminal $t \in T_3$, we now re-route the paths in $\tilde{\mathcal{P}}(t)$, as follows. We let $\mathcal{Z} = \{Z_1(t), \ldots, Z_{\Delta_2}(t)\}$ and $C = C_t$. As before, $\mathcal{Z}$ is a set of Δ_2 tight concentric cycles around C. We let Y be the sub-graph of G induced by $V(D_{t_0})$. Let H' be the graph obtained by the union of the cycles in $\mathcal{Z}$ and the set $\tilde{\mathcal{P}}(t)$ of paths. Let H be the union of H' and $G[Y]$. Then the paths in $\tilde{\mathcal{P}}(t)$ originate at the vertices of A_t lying on C_t, terminate at the vertices of Y, and are internally disjoint from $C \cup Y$. Moreover, since $\tilde{\mathcal{P}}$ is a good linkage, all paths in $\tilde{\mathcal{P}} \setminus \tilde{\mathcal{P}}(t)$ are disjoint from graph H'. Using Theorem 2.5, we can find a new set $\mathcal{P}'(t)$ of p_1 disjoint paths in graph H', connecting vertices of $V(C_t)$ to vertices of $V(C_{t_0})$. The paths in $\mathcal{P}'(t)$ are guaranteed to be disjoint from all paths in $\tilde{\mathcal{P}} \setminus \mathcal{P}'(t)$, and they remain monotone with respect to $Z_1(t_0), \ldots, Z_{\Delta_2}(t_0)$, since the paths in $\tilde{\mathcal{P}}(t)$ had this property, and $D^*(t) \cap D^*(t_0) = \emptyset$. They also remain disjoint from all discs $\tilde{D}(t')$ for all $t' \in T_3$ distinct from t. We replace the paths of $\tilde{\mathcal{P}}(t)$ with the paths of $\mathcal{P}'(t)$ in $\tilde{\mathcal{P}}$, obtaining a new good A–B linkage, and continue to the next iteration. The final set $\mathcal{P}'$ of paths is obtained from $\tilde{\mathcal{P}}$ once all terminals of T_3 are processed. □

Step 3: Ensuring Property (C3)

So far we have constructed a p_1-crossbar $\left(\bigcup_{t\in T_3\cup\{t_0\}}\mathcal{Z}(t),\mathcal{P}\right)$, that has properties (C1) and (C2).

In this step we will discard some demand pairs from $\mathcal{M}_3$, and some paths from sets $\mathcal{P}(t)$ for terminals $t\in T_3$, so that the resulting set $\mathcal{M}^*$ of the demand pairs, together with the shells around their destination vertices, and the resulting set $\mathcal{P}^*$ of paths give a good crossbar. We do not alter the paths themselves, so Properties (C1) and (C2) will continue to hold.

Consider some path $P\in\mathcal{P}$, and recall that P is monotone with respect to $Z_1(t_0),\ldots,Z_{\Delta_2}(t_0)$. Let P' be the sub-path of P, starting from its first vertex (that lies in A), and terminating at the first vertex of P that lies on $Z_{\Delta_2}(t_0)$. Let $\mathcal{P}'=\{P'\mid P\in\mathcal{P}\}$. As before, for every terminal $t\in T_3$, we denote by $\mathcal{P}'(t)\subseteq\mathcal{P}'$ the set of paths originating at the vertices of A_t, and for each $s\in S_3$, we denote by $\mathcal{P}'(s)\subseteq\mathcal{P}'$ the set containing the unique path originating from s. Let e^* be any edge of $Z_{\Delta_2}(t_0)$, and let R^* be the path $Z_{\Delta_2}(t_0)\setminus e^*$. We prove the following theorem.

Theorem 6.16 *There is an efficient algorithm to compute a subset $\mathcal{M}_4\subseteq\mathcal{M}_3$ of $\Omega(\kappa_3/\log n)$ demand pairs, and to select, for each terminal $t\in\mathcal{T}(\mathcal{M}_4)\cap T_3$ a subset $\mathcal{P}''(t)\subseteq\mathcal{P}'(t)$ of $\Omega(p_1)$ paths, such that there is a partition $\Sigma=\{\sigma(v)\mid v\in\mathcal{T}(\mathcal{M}_4)\}$ of R^* into disjoint segments, where for each $t\in\mathcal{T}(\mathcal{M}_4)\cap T_3$, the paths in $\mathcal{P}''(t)$ are disjoint from $R^*\setminus\sigma(t)$, while for each $s\in\mathcal{T}(\mathcal{M}_4)\cap S_3$, the unique path in $\mathcal{P}'(s)$ is disjoint from $R^*\setminus\sigma(s)$.*

Let S_4 and T_4 denote the sets of all source and all destination vertices of the demand pairs in $\mathcal{M}_4$. Let $\mathcal{P}''=\left(\bigcup_{t\in T_4}\mathcal{P}''(t)\right)\cup\left(\bigcup_{s\in S_4}\mathcal{P}'(s)\right)$.

In other words, Theorem 6.16 ensures that for each terminal $t\in T_4$, the endpoints of the paths in $\mathcal{P}''(t)$ appear on R^* consecutively, with respect to the endpoints of all paths in $\mathcal{P}''$. It is now immediate to complete the construction of the good crossbar for $\mathcal{M}_4$. For every path $P\in\mathcal{P}(t)$ for all $t\in T_4$, we include P in set $\mathcal{P}^*$ only if the corresponding path $P'\in\mathcal{P}'$ belongs to $\mathcal{P}''$. Since the paths in $\mathcal{P}$ are monotone with respect to $Z_1(t_0),\ldots,Z_{\Delta_2}(t_0)$, it is easy to see that Property (C3) is satisfied in the resulting crossbar, and we obtain a good p^*-crossbar for $\mathcal{M}_4$, with $p^*=\Omega(p_1)=\Omega(\Delta/\kappa_0)$, and $|\mathcal{M}_4|=\Omega(\kappa_0/\log n)$, as required. We now focus on the proof of Theorem 6.16.

Proof of Theorem 6.16. Let G' be the graph obtained from G, after removing all vertices and edges lying in $D^\circ_{\Delta_2}(t_0)$. Observe that all paths in $\mathcal{P}'$ are still contained in G'. We view the face of G' where t_0 used to reside as the outer face. Therefore, for each terminal $t\in T_3$, the paths of $\mathcal{P}'(t)$ now connect the vertices of $V(C_t)$ to the vertices lying on the boundary of the outer face.

Consider any destination vertex $t\in T_3$, and let $\mathcal{Q}\subseteq\mathcal{P}'(t)$ be any subset of its corresponding paths, with $|\mathcal{Q}|>2$. Let σ be the shortest sub-path of R^*, containing all endpoints of the paths in $\mathcal{Q}$. Taking the union of σ, $\mathcal{Q}$ and $Z_{\Delta_2}(t)$, we obtain a new auxiliary graph $H(t,\mathcal{Q})$. Let $\gamma(t,\mathcal{Q})$ be the closed curve serving as the outer boundary of this graph, and let $D(t,\mathcal{Q})$ be the disc whose boundary is $\gamma(t,\mathcal{Q})$.

Given two destination vertices $t,t'\in T_3$ with $t\neq t'$, and any two subsets $\mathcal{Q}(t)\subseteq\mathcal{P}'(t)$ and $\mathcal{Q}(t')\subseteq\mathcal{P}'(t')$ of paths, notice that the discs $D(t,\mathcal{Q}(t))$ and $D(t',\mathcal{Q}(t'))$ are either completely disjoint from each other, or one is contained in the other. We say that there is a conflict between $(t,\mathcal{Q}(t))$ and $(t',\mathcal{Q}(t'))$ in the latter case. We also say that there is a conflict between $(t,\mathcal{Q}(t))$ and a source vertex $s\in S_3$, if $s\in D(t,\mathcal{Q}(t))$.

Let $\tilde{\mathcal{M}}_1=\mathcal{M}_3$. Over the course of this step, we will define a series of subsets of demand pairs, $\tilde{\mathcal{M}}_3\subseteq\tilde{\mathcal{M}}_2\subseteq\tilde{\mathcal{M}}_1$, where $|\tilde{\mathcal{M}}_3|=\Omega(|\tilde{\mathcal{M}}_1|/\log n)$. For each $1\le i\le 3$, we will denote by $\tilde{S}_i$ and $\tilde{T}_i$ the sets of the source and the destination vertices of the pairs in $\tilde{\mathcal{M}}_i$, respectively. Our final set of the demand pairs will be $\mathcal{M}^*=\tilde{\mathcal{M}}_3$. For every terminal $t\in\mathcal{T}(\mathcal{M}^*)$, we will define a series of subsets of paths $\mathcal{Q}_3(t) \subseteq \mathcal{Q}_2(t) \subseteq \mathcal{Q}_1(t)$, where we let $\mathcal{Q}_1(t) = \mathcal{P}'(t)$, and we will ensure that $|\mathcal{Q}_3(t)| = \Omega(|\mathcal{Q}_1(t)|) = \Omega(p_1)$. Moreover, we will ensure that for all $t, t' \in \tilde{T}_3$, there is no conflict between $(t, \mathcal{Q}_3(t))$ and $(t', \mathcal{Q}_3(t'))$, and for all $s \in \tilde{S}_3$ and $t \in \tilde{T}_3$, there is no conflict between $(t, \mathcal{Q}_3(t))$ and s.

Our first step is to eliminate all conflicts between the destination vertices in $\tilde{T}_1$. We build a graph F, whose vertex set is $\tilde{T}_1$, and there is a directed edge (t', t) iff (i) $t \neq t'$; (ii) $D(t', \mathcal{Q}_1(t')) \subseteq D(t, \mathcal{Q}_1(t))$; and (iii) there is no terminal $t'' \in \tilde{T}_1 \setminus \{t, t'\}$ with $D(t', \mathcal{Q}_1(t')) \subseteq D(t'', \mathcal{Q}_1(t'')) \subseteq D(t, \mathcal{Q}_1(t))$. It is easy to see that F is a directed forest.

We use Claim 2.2, to obtain a partition $\left\{R_1, \ldots, R_{\lceil \log n \rceil}\right\}$ of $V(F)$ into subsets, such that, for each $1 \leq j \leq \lceil \log n \rceil$, $F[R_j]$ is a collection $\mathcal{Y}_j$ of disjoint paths, and if vertices $v, v' \in R_j$ lie on two distinct paths in $\mathcal{Y}_j$, then neither is a descendant of the other in F.

Clearly, there is an index $1 \leq j \leq \lceil \log n \rceil$, with $|R_j| \geq |\tilde{T}_1| / \lceil \log n \rceil$. We let $\tilde{\mathcal{M}}_2 \subseteq \tilde{\mathcal{M}}_1$ be the set of the demand pairs whose destination vertices lie in R_j, and we define the sets $\tilde{S}_2$ and $\tilde{T}_2$ of source and destination vertices accordingly. For each $t \in \tilde{T}_2$, we now define a subset $\mathcal{Q}_2(t) \subseteq \mathcal{Q}_1(t)$ of paths, to ensure that there are no conflicts between the terminals in $\tilde{T}_2$.

Recall that $F[R_j] = \mathcal{Y}_j$ is a collection of paths, and for vertices $v, v' \in R_j$, if there is a directed path from v to v' in F, then v, v' lie on the same path in $\mathcal{Y}_j$. Therefore, if t, t' do not lie on the same path in $\mathcal{Y}_j$, there is no conflict between $(t, \mathcal{Q}_1(t))$ and $(t', \mathcal{Q}_1(t'))$. So we only need to resolve conflicts between terminals lying on the same path of $\mathcal{Y}_j$.

Let $P = (t_1, t_2, \ldots, t_r)$ be any such directed path. Then $D(t_1, \mathcal{Q}_1(t_1)) \subseteq D(t_2, \mathcal{Q}_1(t_2)) \subseteq \cdots \subseteq D(t_r, \mathcal{Q}_1(t_r))$. Disc $D(t_{r-1}, \mathcal{Q}_1(t_{r-1}))$ partitions the paths in $\mathcal{Q}_1(t_r)$ into two subsets, that go on each side of the disc. By discarding the paths in one of these two subsets (the one containing fewer paths), we can eliminate the conflict between t_r and the remaining terminals on path P. Therefore, there is a subset $\mathcal{Q}_2(t_r) \subseteq \mathcal{Q}_1(t_r)$, containing at least $|\mathcal{Q}_1(t_r)|/2$ paths, such that $(t_r, \mathcal{Q}_2(t_r))$ has no conflict with $(t_i, \mathcal{Q}_1(t_i))$ for any $1 \leq i < r$. We process all other terminals $t \in P$ similarly, obtaining a subset $\mathcal{Q}_2(t) \subseteq \mathcal{Q}_1(t)$ of paths, where $|\mathcal{Q}_2(t)| \geq |\mathcal{Q}_1(t)|/2$, and for all $t, t' \in P$ with $t \neq t'$, there is no conflict between $(t, \mathcal{Q}_2(t))$ and $(t', \mathcal{Q}_2(t'))$.

This completes the definition of the set $\tilde{\mathcal{M}}_2$ of demand pairs, and the corresponding sets $\mathcal{Q}_2(t)$ of paths for $t \in \tilde{T}_2$. Our next step is to eliminate conflicts between pairs $(t, \mathcal{Q}_2(t))$ for $t \in \tilde{T}_2$ and the sources $s \in \tilde{S}_2$. For every demand pair $(s, t) \in \tilde{\mathcal{M}}_2$, if there is a conflict between $(t, \mathcal{Q}_2(t))$ and s, then we can discard a subset of at most half the paths from $\mathcal{Q}_2(t)$ in order to eliminate this conflict. The resulting set of paths is denoted by $\mathcal{Q}_3(t)$. Therefore, we will assume from now on that for every demand pair $(s, t) \in \tilde{\mathcal{M}}_2$, there is no conflict between $(t, \mathcal{Q}_3(t))$ and s.

Finally, we build a conflict graph H, whose vertex set is $\left\{v_{s,t} \mid (s,t) \in \tilde{\mathcal{M}}_2\right\}$, and there is a directed edge $(v_{s,t}, v_{s',t'})$ iff there is a conflict between s and $(t', \mathcal{Q}_3(t'))$. Since the discs $\{D(t, \mathcal{Q}_3(t))\}_{t \in \tilde{T}_2}$ are all disjoint, every vertex $v_{s,t}$ has at most one out-going edge. Therefore, every vertex-induced subgraph of H has at least one vertex whose total degree (counting the incoming and the outgoing edges) is at most 2. Using standard techniques, we can find an independent set I of vertices in H, with $|I| = \Omega(|V(H)|)$. Our final set $\tilde{\mathcal{M}}_3$ of demand pairs contains all pairs (s, t) with $v_{s,t} \in I$. We define the sets $\tilde{S}_3$ and $\tilde{T}_3$ of the source and the destination vertices accordingly, and the sets $\mathcal{Q}_3(t)$ for the destination vertices $t \in \tilde{T}_3$ remain the same. From the above discussion, for each $t, t' \in \tilde{T}_3$ with $t \neq t'$, there is no conflict between $(t, \mathcal{Q}_3(t))$ and $(t', \mathcal{Q}_3(t'))$, and for all $s \in \tilde{S}_3$, $t' \in \tilde{T}_3$, there is no conflict between $(t', \mathcal{Q}_3(t'))$ and s. For every source vertex $s \in \tilde{S}_3$, the set $\mathcal{Q}_3(s)$ contains the same path as the original set $\mathcal{P}'(s)$. $\square$

7 Case 2: Heavy Demand Pairs.

In this case, we assume that at least $0.7|\mathcal{M}|$ demand pairs are heavy. Let $\mathcal{H} = \{X_1, \ldots, X_q\} \subseteq \mathcal{X}$ be the collection of all heavy subsets of terminals, so $q \leq 2W/\tau$, and let $\mathcal{M}^h$ be the set of all heavy demand pairs, so for all $(s,t) \in \mathcal{M}^h$, both s and t lie in the sets of $\mathcal{H}$.

We partition the set $\mathcal{M}^h$ of demand pairs into q^2 subsets, where for $1 \leq i,j \leq q$, set $\mathcal{M}_{i,j}$ contains all demand pairs (s,t) with $s \in X_i$ and $t \in X_j$ (notice that it is possible that $i = j$). We then find an approximate solution to each resulting problem separately. The main theorem of this section is the following.

Theorem 7.1 *There is an efficient algorithm, that for each $1 \leq i,j \leq q$, computes a subset $\mathcal{M}'_{i,j} \subseteq \mathcal{M}_{i,j}$ of at least $5|\mathcal{M}_{i,j}|/6$ demand pairs, and a collection $\mathcal{P}_{i,j}$ of node-disjoint paths routing a subset of the demand pairs in $\mathcal{M}'_{i,j}$ in G, with $|\mathcal{P}_{i,j}| \geq \min\left\{\frac{\mathsf{OPT}(G,\mathcal{M}'_{i,j})}{c_1\Delta_0^8\log^3 n}, \frac{\alpha_{\mathrm{WL}}\cdot|\mathcal{M}'_{i,j}|}{c_2\Delta_0^2}\right\}$, for some universal constants c_1 and c_2.*

Before we prove this theorem, we show that it concludes the proof of Theorem 4.3 for Case 2. Let set π contain all pairs (i,j) with $1 \leq i,j \leq q$, such that $|\mathcal{M}_{i,j}| \geq 0.1|\mathcal{M}|/q^2$, and let $\tilde{\mathcal{M}} = \bigcup_{(i,j)\in\pi}\mathcal{M}_{i,j}$. Since the total number of heavy demand pairs is at least $0.7|\mathcal{M}|$, it is easy to verify that $|\tilde{\mathcal{M}}| \geq 0.6|\mathcal{M}|$.

We apply Theorem 7.1, to compute, for each $(i,j) \in \pi$, the subset $\mathcal{M}'_{i,j} \subseteq \mathcal{M}_{i,j}$ of at least $5|\mathcal{M}_{i,j}|/6$ demand pairs and the corresponding set $\mathcal{P}_{i,j}$ of paths routing a subset of the demand pairs in $\mathcal{M}'_{i,j}$. Let $\tilde{\mathcal{M}}' \subseteq \tilde{\mathcal{M}}$ be the set of all demand pairs in $\bigcup_{(i,j)\in\pi}\mathcal{M}'_{i,j}$. Then:

$$|\tilde{\mathcal{M}}'| = \sum_{(i,j)\in\pi}|\mathcal{M}'_{i,j}| \geq \sum_{(i,j)\in\pi}5|\mathcal{M}_{i,j}|/6 = 5|\tilde{\mathcal{M}}|/6 \geq |\mathcal{M}|/2.$$

If, for any $(i,j) \in \pi$, we obtain a solution with $|\mathcal{P}_{i,j}| \geq \frac{W}{2^{13}\cdot\alpha_{\mathrm{AKR}}\cdot c_1c_2q^2\Delta_0^8\log^3 n\cdot\log k}$, then we return the set $\mathcal{P}_{i,j}$ as our final solution.

Substituting $\Delta_0 = O(\Delta\log n)$, $\Delta = \lceil W^{2/19}\rceil$, $q = O(W/\tau)$, and $\tau = W^{18/19}$, we get that:

$$\begin{aligned}|\mathcal{P}_{i,j}| &\geq \Omega\left(\frac{W\tau^2}{W^2\Delta^8\log^{11}n\log k}\right)\\ &\geq \Omega\left(\frac{W^{36/19}}{W\cdot W^{16/19}\log^{11}n\log k}\right)\\ &= \Omega\left(\frac{W^{1/19}}{\log^{11}n\log k}\right).\end{aligned}$$

Otherwise, for all $(i,j) \in \pi$, the resulting solution $|\mathcal{P}_{i,j}| < \frac{W}{2^{13}\cdot\alpha_{\mathrm{AKR}}\cdot c_1c_2q^2\Delta_0^8\log^3 n\cdot\log k}$. We then return the subset $\tilde{\mathcal{M}}'$ of demand pairs. As observed above, $|\tilde{\mathcal{M}}'| \geq |\mathcal{M}|/2$, so it is now enough to show that $\mathsf{OPT}(G,\tilde{\mathcal{M}}') \leq w^*|\tilde{\mathcal{M}}'|/8$.

Assume otherwise, and let $\mathcal{P}^*$ be a solution to instance $(G,\tilde{\mathcal{M}}')$, routing a subset $\mathcal{M}^* \subseteq \tilde{\mathcal{M}}'$ of at least $w^*|\tilde{\mathcal{M}}'|/8 \geq w^*|\mathcal{M}|/16$ demand pairs. Then there is a pair of indexes $(i,j) \in \pi$, such that $|\mathcal{M}^*\cap\mathcal{M}'_{i,j}| \geq \frac{w^*|\mathcal{M}|}{16q^2}$. Therefore, $\mathsf{OPT}(G,\mathcal{M}'_{i,j}) \geq \frac{w^*|\mathcal{M}|}{16q^2}$. From Theorem 7.1, we compute a set $\mathcal{P}_{i,j}$ of paths, routing a subset of demand pairs of $\mathcal{M}'_{i,j}$, with either $|\mathcal{P}_{i,j}| \geq \frac{\mathsf{OPT}(G,\mathcal{M}'_{i,j})}{c_1\Delta_0^8\log^3 n}$, or $|\mathcal{P}_{i,j}| \geq \frac{\alpha_{\mathrm{WL}}\cdot|\mathcal{M}'_{i,j}|}{c_2\Delta_0^2}$.

In the former case,

$$|\mathcal{P}_{i,j}| \geq \frac{\mathsf{OPT}(G, \mathcal{M}'_{i,j})}{c_1 \Delta_0^8 \log^3 n} \geq \frac{w^* |\mathcal{M}|}{16 c_1 q^2 \Delta_0^8 \log^3 n} = \frac{W}{16 c_1 q^2 \Delta_0^8 \log^3 n},$$

while in the latter case, observe that $|\mathcal{M}'_{i,j}| \geq 5|\mathcal{M}_{i,j}|/6 \geq |\mathcal{M}|/(12q^2)$ from the definition of π and $\tilde{M}$. Therefore,

$$|\mathcal{P}_{i,j}| \geq \frac{\alpha_{\mathrm{WL}} \cdot |\mathcal{M}'_{i,j}|}{c_2 \Delta_0^2} \geq \frac{w^* \cdot |\mathcal{M}|}{12 \cdot 512 \cdot \alpha_{\mathrm{AKR}} \cdot c_2 q^2 \Delta_0^2 \log k} > \frac{W}{2^{13} \cdot \alpha_{\mathrm{AKR}} \cdot c_1 c_2 q^2 \Delta_0^8 \log^2 n \log k},$$

a contradiction.

From now on we focus on proving Theorem 7.1. We fix a pair of indices $1 \leq i, j \leq q$. In order to simplify the notation, we denote $\mathcal{M}_{i,j}$ by $\mathcal{N}$, X_i by X and X_j by Y. Our goal is to compute a subset $\mathcal{N}' \subseteq \mathcal{N}$ of at least $5|\mathcal{N}|/6$ demand pairs, together with a set $\mathcal{P}$ of at least $\min\left\{\Omega\left(\frac{\mathsf{OPT}(G,\mathcal{N}')}{\Delta_0^8 \log^3 n}\right), \Omega\left(\frac{\alpha_{\mathrm{WL}}|\mathcal{N}'|}{\Delta_0^2}\right)\right\}$ disjoint paths, routing a subset of the demand pairs in $\mathcal{N}'$. Let $x \in X$ and $y \in Y$ be any pair of terminals. Recall that for every terminal $t \in \mathcal{T}(\mathcal{N}) \cap X$, $d(t,x) \leq \Delta_0$, and for every terminal $t \in \mathcal{T}(\mathcal{N}) \cap Y$, $d(t,y) \leq \Delta_0$. We consider two subcases. The first subcase happens when $d(x,y) > 5\Delta_0$, and otherwise the second subcase happens. Note that the second subcase includes the case where $X = Y$.

7.1 Subcase 2a: $d(x,y) > 5\Delta_0$

In this case, we set $\mathcal{N}' = \mathcal{N}$. We will compute a set $\mathcal{P}$ of at least $\Omega\left(\frac{\mathsf{OPT}(G,\mathcal{N})}{\Delta_0^6 \log n}\right)$ node-disjoint paths, routing a subset of the demand pairs in $\mathcal{N}$. We start by defining a simpler special case of the problem, and show that we can find a good approximation algorithm for this special case. The special case is somewhat similar to routing on a cylinder, and we solve it by reducing it to this setting. We then show that the general problem in Case 2a reduces to this special case.

A Special Case

Suppose we are given a connected planar graph $\hat{G}$ embedded on the sphere, and two disjoint simple cycles Z, Z' in $\hat{G}$. Suppose also that we are given a set $\hat{\mathcal{M}}$ of demand pairs, where all source vertices lie on Z and all destination vertices lie on Z' (we note that the same vertex may participate in a number of demand pairs). Let $D(Z), D(Z')$ be two discs with boundaries Z and Z', respectively, so that $D(Z) \cap D(Z') = \emptyset$. Assume additionally that we are given a closed $\hat{G}$-normal curve C of length at most Δ, that is contained in $D^\circ(Z)$, so that for every vertex $v \in Z$, there is a $\hat{G}$-normal curve $\gamma(v)$ of length at most $2\Delta_0$ connecting v to a vertex of C, and $\gamma(v)$ is internally disjoint from Z and C. Similarly, assume that we are given a closed $\hat{G}$-normal curve C' of length at most Δ, that is contained in $D^\circ(Z')$, so that for every vertex $v' \in Z'$ there is a $\hat{G}$-normal curve $\gamma(v')$ of length at most $2\Delta_0$, connecting v' to a vertex of C', and $\gamma(v')$ is internally disjoint from Z' and C' (see Figure 5). This finishes the definition of the special case. The following theorem shows that we can obtain a good approximation for it.

Theorem 7.2 *There is an efficient algorithm, that, given any instance $(\hat{G}, \hat{\mathcal{M}})$ of the* NDP *problem as above, computes a solution of value at least $\Omega\left(\frac{OPT(\hat{G},\hat{\mathcal{M}})}{\Delta_0^2 \log n}\right)$.*

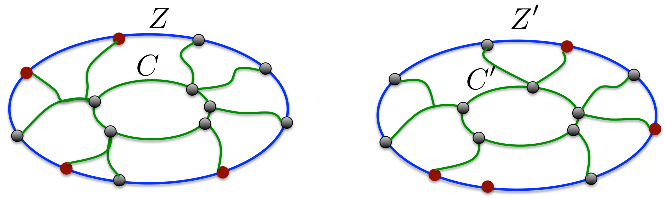


Figure 5: The special case, with the terminals shown in red.

Proof: The algorithm is very simple: we reduce the problem to routing on a cylinder, by creating two holes in the sphere. The first hole is $D(Z)$: we delete all edges and vertices that appear inside the disc, except for the edges and the vertices of Z. The second hole is $D(Z')$: we similarly delete all edges and vertices that lie inside the disc, except for those lying on its boundary. Let $\hat{G}'$ be the resulting graph. We then apply the $O(\log n)$-approximation algorithm for NDP-Cylinder to the resulting problem, to obtain a routing of at least $\Omega\left(\frac{\mathsf{OPT}(\hat{G}',\hat{\mathcal{M}})}{\log n}\right)$ demand pairs. In order to complete the analysis of the algorithm, it is enough to prove that $\mathsf{OPT}(\hat{G}',\hat{\mathcal{M}}) \geq \Omega\left(\frac{\mathsf{OPT}(\hat{G},\hat{\mathcal{M}})}{\Delta_0^2}\right)$.

Notice that we can assume without loss of generality that all curves in set $\{\gamma(v) \mid v \in V(Z)\}$ are mutually non-crossing, and moreover, whenever two curves meet, they continue together. In other words, for all $v, v' \in V(Z)$, $\gamma(v) \cap \gamma(v')$ is a contiguous curve that has a vertex of C as its endpoint. We make a similar assumption for curves in $\{\gamma(v) \mid v \in V(Z')\}$.

Consider the optimal solution to instance $(\hat{G},\hat{\mathcal{M}})$, and let $\mathcal{P}_0$ be the set of paths in this solution. We will gradually modify the set $\mathcal{P}_0$ of paths, to obtain path sets $\mathcal{P}_1, \mathcal{P}_2, \ldots$, until we obtain a feasible solution to instance $(\hat{G}',\hat{\mathcal{M}})$. For every $i \geq 0$, we will denote by $\hat{\mathcal{M}}_i$ the set of the demand pairs routed by $\mathcal{P}_i$, and by κ_i its cardinality. Recall that $\kappa_0 = \mathsf{OPT}(\hat{G},\hat{\mathcal{M}})$. We assume that $\kappa_0 \geq 512\Delta_0^2$, as otherwise a solution routing a single demand pair gives the desired approximation, and such a solution exists in $\hat{G}'$, as it must be connected.

We delete from $\mathcal{P}_0$ all paths that use the vertices of C or C'. Since both curves have length at most Δ, we delete at most 2Δ paths in this step. Let $\mathcal{P}_1$ be the resulting set of paths, and $\hat{\mathcal{M}}_1$ the set of the demand pairs routed by $\mathcal{P}_1$.

Our next step is to build a conflict graph H. Its set of vertices, $V(H) = \left\{v(s,t) \mid (s,t) \in \hat{\mathcal{M}}_1\right\}$. There is a directed edge from $v(s,t)$ to $v(s',t')$, iff the path $P(s,t) \in \mathcal{P}_1$ routing the pair (s,t) contains a vertex of $V(\gamma(s')) \cup V(\gamma(t'))$, and we say that there is a conflict between (s,t) and (s',t') in this case. Since we assume that the paths in $\mathcal{P}_1$ are node-disjoint, and since $|V(\gamma(s'))|, |V(\gamma(t'))| \leq 2\Delta_0$ for all $(s',t') \in \hat{\mathcal{M}}_1$, the in-degree of every vertex in H is at most $4\Delta_0$. Therefore, the average degree (including the incoming and the outgoing edges) of every induced sub-graph of H is at most $8\Delta_0$, and there is an independent set $I \subseteq V(H)$ of cardinality at least $\frac{|\mathcal{P}_1|}{8\Delta_0+1} \geq \frac{|\mathcal{P}_0|}{16\Delta_0}$ in H.

Let $\hat{\mathcal{M}}_2$ be the set of all demand pairs (s,t) with $v(s,t) \in I$, and let $\mathcal{P}_2 \subseteq \mathcal{P}_1$ be the set of paths routing the demand pairs in $\hat{\mathcal{M}}_2$. Recall that the paths in $\mathcal{P}_2$ are disjoint from $C \cup C'$. Moreover, if $P(s,t) \in \mathcal{P}_2$ is the path routing the pair $(s,t) \in \hat{\mathcal{M}}_2$, then for every demand pair $(s',t') \neq (s,t)$ in $\hat{\mathcal{M}}_2$, path $P(s,t)$ is disjoint from both $\gamma(s')$ and $\gamma(t')$. It is now easy to verify that the demand pairs in $\hat{\mathcal{M}}_2$ are non-crossing, that is, we can find an ordering $\hat{\mathcal{M}}_2 = \{(s_1,t_1), \ldots, (s_{\kappa_2}, t_{\kappa_2})\}$ of the demand pairs in $\hat{\mathcal{M}}_2$, so that $s_1, \ldots, s_{\kappa_2}$ appear in this counter-clock-wise order on Z, while $t_1, \ldots, t_{\kappa_2}$ appear in this clock-wise order on Z'.

Let $\kappa_3 = \left\lfloor \frac{\kappa_2}{8\Delta_0} \right\rfloor - 1$, and let $\hat{\mathcal{M}}_3 = \{(s_{8\Delta_0 r}, t_{8\Delta_0 r}) \mid 1 \le r \le \kappa_3\}$. In other words, we space the demand pairs in $\hat{\mathcal{M}}_2$ out, by adding one in $8\Delta_0$ such pairs to $\hat{\mathcal{M}}_3$. Let $\mathcal{P}_3 \subseteq \mathcal{P}_2$ be the set of paths routing the demand pairs in $\mathcal{M}_3$, so $|\mathcal{P}_3| \ge \frac{|\mathcal{P}_2|}{32\Delta_0} \ge \frac{|\mathcal{P}_0|}{512\Delta_0^2}$. Our final step is to show that all demand pairs in $\hat{\mathcal{M}}_3$ can be routed in graph $\hat{G}'$ via node-disjoint paths. In order to do so, for each such demand pair $(s_{8\Delta_0 r}, t_{8\Delta_0 r})$, we define a segment μ_r of Z containing $s_{8\Delta_0 r}$, and a segment μ'_r of Z' containing $t_{8\Delta_0 r}$, as follows. For convenience, denote $8\Delta_0 r$ by ℓ. The first segment, μ_r, is simply the segment of Z from $s_{\ell-4\Delta_0}$ to $s_{\ell+4\Delta_0-1}$, as we traverse Z in the counter-clock-wise order. The second segment, μ'_r, is the segment of Z from $t_{\ell-4\Delta_0}$ to $t_{\ell+4\Delta_0-1}$, as we traverse Z' in the clock-wise order. It is immediate to verify that all segments of Z in $\{\mu_r \mid 1 \le r \le \kappa_3\}$ are mutually disjoint, and the same holds for all segments of Z' in $\{\mu'_r \mid 1 \le r \le \kappa_3\}$. The crux of the analysis is the following lemma.

Lemma 7.3 *Let $(s_{8\Delta_0 r}, t_{8\Delta_0 r}) \in \mathcal{M}_3$ be any demand pair, and let $P \in \mathcal{P}_3$ be the path routing it. Then $P \cap Z \subseteq \mu_r$, and $P \cap Z' \subseteq \mu'_r$.*

Before we prove this lemma, we show that we can use it to obtain a routing of the demand pairs in $\hat{\mathcal{M}}_3$ in graph $\hat{G}'$ via node-disjoint paths. Let $(s_\ell, t_\ell) \in \hat{\mathcal{M}}_3$ be any demand pair (where $\ell = 8\Delta_0 r$ for some $1 \le r \le \kappa_3$), and let P_ℓ be the path routing (s_ℓ, t_ℓ) in $\mathcal{P}_3$, that we view as directed from s_ℓ towards t_ℓ. Clearly, P_ℓ intersects both μ_r and μ'_r. Let v_ℓ be the last vertex of P_ℓ lying on μ_r. Then there is some other vertex appearing on P_ℓ after v_ℓ that belongs to μ'_r. We let v'_ℓ be the first such vertex on P_ℓ, and we let P'_ℓ be the segment of P_ℓ between v_ℓ and v'_ℓ. Let P^*_ℓ be the path obtained as follows: we start with a segment of μ_r between s_ℓ and v_ℓ; we then follow P'_ℓ to v'_ℓ, and finally we use a segment of μ'_r between v'_ℓ and t_ℓ. From Lemma 7.3, it is immediate to verify that the resulting paths in set $\{P^*_\ell \mid \ell = 8\Delta_0 r; 1 \le r \le \kappa_3\}$ are completely disjoint, contained in $\hat{G}'$, and they route all demand pairs in $\hat{\mathcal{M}}_3$. It now remains to prove Lemma 7.3.

Proof: Fix some demand pair $(s_{8\Delta_0 r}, t_{8\Delta_0 r}) \in \mathcal{M}_3$, and let $P \in \mathcal{P}_3$ be the path routing it. We show that $P \cap Z \subseteq \mu_r$. The proof that $P \cap Z' \subseteq \mu'_r$ is symmetric. For convenience, we denote $8\Delta_0 r$ by ℓ from now on.

Assume otherwise, and let v be the first vertex on P that belongs to $Z \setminus \mu_r$. Let P' be the sub-path of P from its first vertex to v. Consider the planar embedding of $\hat{G}$, where we fix any face contained in $D(Z')$ as the outer face. In this planar embedding, denote by Y the union of $D(C), \gamma(s_\ell), P'$, and $\gamma(v)$, and let R be the outer boundary of Y (notice that P' may intersect $\gamma(s_\ell)$, and that $\gamma(s_\ell)$, $\gamma(v)$ are not necessarily disjoint).

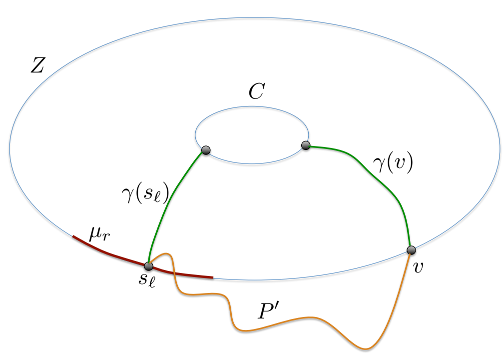


Figure 6: Constructing the curve R

Notice that, since there are no conflicts in $\mathcal{P}_2$, curve R does not cross any curve in $C' \cup \{\gamma(t_h) \mid 1 \le h \le \kappa_2\}$, and so all destination vertices of the demand pairs in $\hat{\mathcal{M}}_2$ lie on the outside of R. Let $S_1 =$

$\{s_{\ell-4\Delta_0}, \ldots, s_{\ell-1}\}$, and $S_2 = \{s_{\ell+1}, \ldots, s_{\ell+4\Delta_0-1}\}$. Denote $\Gamma_1 = \{\gamma(s_i) \mid s_i \in S_1\}$ and $\Gamma_2 = \{\gamma(s_i) \mid s_i \in S_2\}$. Since the curves in set $\{\gamma(u) \mid u \in V(Z)\}$ are non-crossing, if σ, σ' are the two segments of Z whose endpoints are s_ℓ and v, then all vertices of $S_1 \setminus \{v\}$ lie on one of the segments (say σ), while all vertices of $S_2 \setminus \{v\}$ lie on the other segment. Moreover, since path P' cannot cross any curve in $\Gamma_1 \cup \Gamma_2$, either all sources of S_1, or all sources of S_2 lie inside the curve R - let us assume that it is the former.

All sources of S_1 are separated by R from their destinations vertices, and yet all corresponding demand pairs are routed by $\mathcal{P}_2$. Therefore, at least $4\Delta_0 - 2$ paths in $\mathcal{P}_2$ must cross the curve R. Recall that none of these paths can cross C, P', or $\gamma(s_\ell)$. Therefore, all these paths must cross $\gamma(v)$. But since the length of $\gamma(v)$ is at most $2\Delta_0$, and the paths are node-disjoint, this is impossible. □

□

Completing the Proof

We now complete the proof of Theorem 7.1 for Case 2a, by reducing it to the special case defined above.

We assume that $\mathsf{OPT}(G, \mathcal{N}) > 2^{13}\Delta_0^4$, since otherwise we can route a single demand pair and obtain a valid solution. We denote by S and T the sets of all source and all destination vertices of the demand pairs in $\mathcal{N}$, respectively.

Our first step is to construct shells $\mathcal{Z}(x) = (Z_1(x), \ldots, Z_{2\Delta_0}(x))$ and $\mathcal{Z}(y) = (Z_1(y), \ldots, Z_{2\Delta_0}(y))$ of depth $2\Delta_0$ around x and y, respectively. We would like to ensure that disc $D(Z_{2\Delta_0}(x))$ contains all terminals of X and no terminals of Y, and similarly, disc $D(Z_{2\Delta_0}(y))$ contains all terminals of Y and no terminals of X. In order to ensure this, when constructing the shell $\mathcal{Z}(x)$, we let the face F_x (that is viewed as the outer face in the plane embedding of G when constructing the shell) be the face incident on the terminal y, and similarly we let F_y be the face incident on x (recall that $d(x, y) > 5\Delta_0$, so this choice of faces is consistent with the requirement that the shell depth is bounded by $\min\{d_{\mathrm{GNC}}(v, C_x)\} - 1$ over all vertices v lying on the boundary of F_x). Let $\mathcal{Z}(x) = (Z_1(x), \ldots, Z_{2\Delta_0}(x))$ and $\mathcal{Z}(y) = (Z_1(y), \ldots, Z_{2\Delta_0}(y))$ be the resulting shells.

Claim 7.4 *Disc $D(Z_{2\Delta_0}(x))$ contains all terminals of X and no terminals of Y, and similarly, disc $D(Z_{2\Delta_0}(y))$ contains all terminals of Y and no terminals of X. Moreover, $D(Z_{2\Delta_0}(x)) \cap D(Z_{2\Delta_0}(y)) = \emptyset$.*

Proof: Let $t \in X$ be any terminal, and assume for contradiction that $t \notin D(Z_{2\Delta_0}(x))$. Recall that $d(x, t) \leq \Delta_0$, and so there is a G-normal curve γ of length at most Δ_0, connecting some vertex $v \in C_t$ to some vertex $u \in C_x$. Then every vertex $v' \in C_t$ has a G-normal curve $\gamma(v')$ of length at most $\Delta_0 + \Delta$ connecting it to u. Therefore, all vertices of C_t lie in the disc $D(Z_{\Delta_0+\Delta}(x))$ (if some vertex $v' \in C_t$ does not lie in this disc, then there are $\Delta_0 + \Delta$ disjoint cycles separating v' from u, a contradiction). We conclude that all vertices of C_t lie in disc $D(Z_{2\Delta_0}(x))$, but t does not lie in that disc. This can only happen if the outer face $F_x \subseteq D_t$, meaning that y lies in D_t. But then, from our definition of enclosures, $d(t, y) = 1$ must hold. However, $t \in X$, $y \in Y$, and $X \neq Y$, so $d(t, y) \geq 5\Delta$ from the definition of the family $\mathcal{X}$ of sets of terminals, a contradiction. We conclude that all terminals of X lie in disc $D(Z_{2\Delta_0}(x))$. Using a similar reasoning, all terminals of Y lie in disc $D(Z_{2\Delta_0}(y))$.

In order to complete the proof of the theorem it is now enough to show that $D(Z_{2\Delta_0}(x)) \cap D(Z_{2\Delta_0}(y)) = \emptyset$. Observe that from Property (J5) of the shells, the vertices of $V(D_y)$ all lie outside the disc $D(Z_{2\Delta_0}(x))$, as all such vertices belong to the connected component Y_x. Therefore, $y \in D(Z_{2\Delta_0}(y)) \setminus D(Z_{2\Delta_0}(x))$, and similarly $x \in D(Z_{2\Delta_0}(x)) \setminus D(Z_{2\Delta_0}(y))$, so neither of the two discs is contained in the other. Assume for contradiction that the intersection of the two discs is non-empty. Observe first that the boundaries of the two discs cannot intersect. Indeed, assume otherwise, and let v be any vertex lying in $Z_{2\Delta_0}(x) \cap Z_{2\Delta_0}(y)$. Then there are G-normal curves γ and γ' of length at most $2\Delta_0$ each, connecting v to a vertex of C_x and a vertex of C_y, respectively, implying that $d(x, y) \leq 4\Delta_0$, a contradiction. Therefore, all vertices of $Z_{2\Delta_0}(y)$ must lie inside the disc $D(Z_{2\Delta_0}(x))$. However, the vertices of C_y lie outside $D(Z_{2\Delta_0}(x))$, and at least one such vertex v can be connected to some vertex of $Z_{2\Delta_0}(y)$ with a G-normal curve of length at most $2\Delta_0 + 1$. This curve must intersect $Z_{2\Delta_0}(x)$ at some vertex that we denote by u, and u can be in turn connected to a vertex of C_x by a G-normal curve of length at most $2\Delta_0 + 1$, implying that $d(x, y) < 5\Delta_0$, a contradiction. □

For consistency of notation, we will denote $Z_0(x) = C_x$ and $Z_0(y) = C_y$, even though both C_x and C_y are G-normal curves and not cycles.

Let $U = \bigcup_{h=0}^{2\Delta_0} V(Z_h(x))$, and let $U' = \bigcup_{h'=0}^{2\Delta_0} V(Z_{h'}(y))$. For each $1 \leq h \leq 2\Delta_0$, let U_h be the set of vertices lying in $D^\circ(Z_h(x)) \setminus D(Z_{h-1}(x))$, and let $\mathcal{R}_h$ be the set of all connected components of $G[U_h]$. For each $1 \leq h' \leq 2\Delta_0$, we define $U'_{h'}$ and $\mathcal{R}'_{h'}$ with respect to the shell $\mathcal{Z}(y)$ similarly. Let $\mathcal{R} = \bigcup_{h=1}^{2\Delta_0} \mathcal{R}_h$ and let $\mathcal{R}' = \bigcup_{h'=1}^{2\Delta_0} \mathcal{R}'_{h'}$.

Our next step is to define a mapping $\beta : S \to 2^U$ of all source vertices in S to subsets of vertices of U, and a mapping $\beta' : T \to 2^{U'}$ of all destination vertices in T to subsets of vertices of U'. Every vertex in $S \cup T$ will be mapped to a subset of at most three vertices. We will then replace each demand pair (s, t) with the set $\beta(s) \times \beta'(t)$ of demand pairs. Eventually, for every pair $0 \leq h, h' \leq 2\Delta_0$ of indices, we will define a subset $\tilde{\mathcal{M}}_{h,h'}$ of the new demand pairs, containing all pairs whose sources lie on $Z_h(x)$ and destinations lie on $Z_{h'}(y)$, to obtain an instance of the special case, that will then be solved using Theorem 7.2.

We start by defining the mapping of the sources. For every source $s \in S$, we will also define a curve $\Gamma(s)$ that we will use later in our analysis. First, for every source vertex $s \in S \cap U$, we set $\beta(s) = \{s\}$, and we let $\Gamma(s)$ contain the vertex s only. Next, fix any vertex $v^* \in C_x$. For every source vertex $s \in V(D_x) \setminus V(C_x)$, we set $\beta(s) = \{v^*\}$, and $\Gamma(s) = C_x$. Finally, consider some component $R \in \mathcal{R}_h$, for some $1 \leq h \leq 2\Delta_0$, and let $s \in S \cap V(R)$ be any source lying in R. If $|L(R)| \leq 2$, then we let $\beta(s) = L(R) \cup \{u(R)\}$ if $u(R)$ is defined, and $\beta(s) = L(R)$ otherwise. If $|L(R)| > 2$, then we let $\beta(s) = \{v\}$, where v is a leg of R, such that v is not an endpoint of $\sigma(R)$ (so it is not the first and not the last leg of R). We then let $\Gamma(s)$ be the outer boundary $\gamma(R)$ of the disc $\eta(R)$ given by Theorem 5.5. Recall that the length of $\Gamma(s)$ is bounded by $4\Delta_0 + \Delta/2 + 1 < 5\Delta_0$. Notice that for every source $s \in S$, we now have defined a G-normal curve $\Gamma(s)$ of length at most $5\Delta_0$. An important property of this curve is that the disc whose boundary is $\Gamma(s)$ cannot contain any demand pair in $\mathcal{N}$, as the disc itself is contained in $D(Z_{2\Delta_0}(x))$. We define the mapping $\beta' : T \to 2^{U'}$, and the corresponding curves $\Gamma(t)$ for the destination vertices $t \in T$ similarly.

Let $\tilde{\mathcal{M}} = \bigcup_{(s,t) \in \mathcal{N}} \beta(s) \times \beta'(t)$. In the following two theorems, we show that the problems $(G, \mathcal{N})$ and $(G, \tilde{\mathcal{M}})$ are equivalent to within relatively small factors.

Theorem 7.5 *There is an efficient algorithm, that, given any solution to instance $(G, \tilde{\mathcal{M}})$, that routes κ demand pairs, finds a solution to instance $(G, \mathcal{N})$, routing at least $\frac{\kappa}{21\Delta_0}$ demand pairs.*

Proof: Let $\mathcal{P}_0$ be any collection of disjoint paths in graph G, routing a subset $\tilde{\mathcal{M}}_0 \subseteq \tilde{\mathcal{M}}$ of κ demand pairs. We assume that $\kappa \geq 21\Delta_0$, as otherwise we can return a routing of a single demand pair in $\mathcal{N}$. For every demand pair $(\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}_0$, let (s, t) be any corresponding demand pair in $\mathcal{N}$, that is, $\tilde{s} \in \beta(s)$ and $\tilde{t} \in \beta'(t)$.

We build a conflict graph H, whose vertex set is $\left\{v(\tilde{s}, \tilde{t}) \mid (\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}_0\right\}$, and there is a directed edge from $v(\tilde{s}_1, \tilde{t}_1)$ to $v(\tilde{s}_2, \tilde{t}_2)$ iff the unique path $P(\tilde{s}_1, \tilde{t}_1) \in \mathcal{P}_0$ routing the pair $(\tilde{s}_1, \tilde{t}_1)$ intersects $\Gamma(s_2)$

or $\Gamma(t_2)$ (in which case we say that there is a conflict between $(\tilde{s}_1, \tilde{t}_1)$ and $(\tilde{s}_2, \tilde{t}_2)$). Since all paths in $\mathcal{P}_0$ are node-disjoint, and all curves $\Gamma(s), \Gamma(t)$ have lengths at most $5\Delta_0$, the in-degree of every vertex in H is at most $10\Delta_0$. Therefore, we can efficiently compute an independent set I of size at least $\frac{\kappa}{20\Delta_0+1} \geq \frac{\kappa}{21\Delta_0}$ in H.

Let $\tilde{\mathcal{M}}_1 = \left\{(\tilde{s}, \tilde{t}) \mid v(\tilde{s}, \tilde{t}) \in I\right\}$, and let $\mathcal{P}_1 \subseteq \mathcal{P}_0$ be the set of paths routing the demand pairs in $\tilde{\mathcal{M}}_1$. Let $\mathcal{M}' = \left\{(s,t) \mid (\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}_1\right\}$. It is now enough to show that all demand pairs in $\mathcal{M}'$ can be routed in G. Consider any demand pair $(\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}_1$, and let $P \in \mathcal{P}_1$ be the path routing $(\tilde{s}, \tilde{t})$. We will extend the path P, so it connects s to t. Notice that if $s \in U$, then $s = \tilde{s}$. Assume now that $s \notin U$. If $s \in V(D_x) \setminus V(C_x)$, then $\tilde{s} \in V(C_x)$. Since $G[V(D_x)]$ is a connected graph, we can extend path P inside the disc D_x, so it now originates at s. As $\Gamma(s) = C_x$, no other source of a demand pair in $\tilde{\mathcal{M}}_1$ may lie on C_x, and no other path in $\mathcal{P}_1$ contains a vertex of D_x. Finally, assume that $s \in V(R)$ for some component $R \in \mathcal{R}$. Since the disc whose boundary is $\Gamma(s)$ contains R, all vertices of $L(R)$, and $u(R)$ (if such is defined), no other path in $\mathcal{P}_1$ may contain a vertex of $L(R) \cup \{u(R)\}$. Moreover, since no demand pair in $\tilde{\mathcal{M}}$ is contained in the disc whose boundary is $\Gamma(s)$, no other path in $\mathcal{P}_1$ may intersect R. We extend the path P inside R, so it now originates at s. We perform the same transformation to path P to ensure that it terminates at t. It is easy to see that the resulting collection of paths is disjoint. □

Theorem 7.6 $\mathsf{OPT}(G, \tilde{\mathcal{M}}) \geq \frac{\mathsf{OPT}(G,\mathcal{N})}{21\Delta_0}$.

Proof: Let $\mathcal{P}_0$ be the set of paths in the optimal solution to instance $(G, \mathcal{N})$, and let $\mathcal{M}_0$ be the set of the demand pairs they route.

As before, we define a conflict graph H, whose vertex set is $\{v(s,t) \mid (s,t) \in \mathcal{M}_0\}$, and there is a directed edge from $v(s_1,t_1)$ to $v(s_2,t_2)$ iff the unique path $P(s_1,t_1) \in \mathcal{P}_0$ routing the pair (s_1,t_1) intersects $\Gamma(s_2)$ or $\Gamma(t_2)$ (in which case we say that there is a conflict between (s_1,t_1) and (s_2,t_2)). Since all paths in $\mathcal{P}_0$ are node-disjoint, and all curves $\Gamma(s), \Gamma(t)$ have lengths at most $5\Delta_0$, the in-degree of every vertex in H is at most $10\Delta_0$. Therefore, there is an independent set I of size at least $\frac{\mathsf{OPT}(G,\mathcal{N})}{20\Delta_0+1} \geq \frac{\mathsf{OPT}(G,\mathcal{N})}{21\Delta_0}$ in H.

Let $\mathcal{M}_1 = \{(s,t) \mid v(s,t) \in I\}$, and let $\mathcal{P}_1 \subseteq \mathcal{P}_0$ be the set of paths routing the demand pairs in $\mathcal{M}_1$. We show that we can route $|\mathcal{M}_1|$ demand pairs of $\tilde{\mathcal{M}}$ in G via node-disjoint paths. Let S_1 and T_1 be the sets of all source and all destination vertices of the pairs in $\mathcal{M}_1$, respectively.

Consider any source vertex $s \in S_1$. We say that s is a *good source vertex* if $s \in U$, or s belongs to some component $R \in \mathcal{R}$, such that $|L(R)| \leq 2$. Otherwise, s is a *bad source vertex*. Notice that if s is a good source vertex, then the path $P \in \mathcal{P}_1$ that originates at s must contain a vertex $s' \in \beta(s)$: if $s \in U$, then $\beta(s) = \{s\}$; otherwise, if $s \in R$ for some component $R \in \mathcal{R}$ with $|L(R)| \leq 2$, then $\beta(s) = L(R) \cup \{u(R)\}$ if $u(R)$ is defined, and $\beta(s) = L(R)$ otherwise. In either case, in order to enter R, path P has to visit a vertex of $\beta(s)$. Therefore, if s is a good source vertex, then some vertex $s' \in P$ belongs to $\beta(s)$. Similarly, we say that a destination vertex $t \in T_1$ is a *good destination vertex* if $t \in U'$ or t belongs to some component $R' \in \mathcal{R}'$ with $|L(R')| \leq 2$. Otherwise, it is a *bad destination vertex*. As before, if t is a good destination vertex, then the path $P \in \mathcal{P}_1$ terminating at t must contain some vertex $t' \in \beta(t)$.

We transform the paths in $\mathcal{P}_1$ in two steps, to ensure that they connect demand pairs in $\tilde{\mathcal{M}}$. In the first step, for every path $P \in \mathcal{P}_1$ originating at a good source $s \in S_1$, we truncate P at the first vertex $s' \in \beta(s)$, so it now originates at s'. Similarly, if P terminates at a good destination vertex $t \in T_1$, we truncate P at the last vertex $t' \in \beta(t)$, so it now terminates at t'. Let $\mathcal{P}'_1$ be the resulting set of paths. Notice that the paths in $\mathcal{P}'_1$ remain node-disjoint.

In order to complete our transformation, we need to take care of bad source and destination vertices. Let $s \in S_1$ be any bad destination vertex. If $s \in V(D_x) \setminus V(C_x)$, then let Q_s be any path connecting s to the unique vertex $s' \in \beta(s)$, so that $Q_s \subseteq D_x$. Such a path exists, since $G[V(D_x)]$ is connected. Otherwise, $s \in R$ for some component $R \in \mathcal{R}$ with $|L(R)| \geq 3$. Recall that in this case, $\beta(s)$ contains a unique vertex, that we denote by s', which is a leg of R, and it is not one of the endpoints of $\sigma(R)$. We then let $Q(s)$ be any path connecting s to s' in the sub-graph of G induced by $V(R) \cup \{s'\}$. We define paths $Q(t)$ for bad destination vertices $t \in T_1$ similarly. By concatenating the paths in $\{Q(s)\}$ for all bad source vertices $s \in S_1$, $\mathcal{P}_1'$, and $\{Q(t)\}$ for all bad destination vertices $t \in T_1$, we obtain the desired collection $\tilde{\mathcal{P}}$ of at least $\frac{\mathsf{OPT}(G,\mathcal{N})}{21\Delta_0}$ paths, routing demand pairs in $\tilde{\mathcal{M}}$. It now only remains to show that the paths in $\tilde{\mathcal{P}}$ are disjoint. Recall that the paths in $\mathcal{P}_1'$ were node-disjoint.

Consider some bad source vertex $s \in S_1$, and let $P \in \mathcal{P}_1'$ be the path originating at s. We show that $Q(s)$ is disjoint from all paths in $\mathcal{P}_1' \setminus \{P\}$, and it is disjoint from all other paths $Q(s_1)$ for $s_1 \in S_1 \setminus \{s\}$.

Assume first that $s \in V(D_x) \setminus V(C_x)$. Then $\Gamma(s) = C_x$, and so no other path in $\mathcal{P}_1$ (and hence in $\mathcal{P}_1'$) can contain a vertex of D_x. It follows that $Q(s)$ is disjoint from all paths in $\mathcal{P}_1' \setminus \{P\}$. It is also disjoint from all other paths $Q(s_1)$ for $s_1 \in S_1 \setminus \{s\}$, since in order for $Q(s_1)$ to intersect D_x, vertex s_1 must lie on C_x, and this is impossible.

Assume now that $s \in V(R)$ for some $R \in \mathcal{R}$. Recall that the disc whose boundary is $\Gamma(s)$ contains $R \cup L(R)$. Since no other path in $\mathcal{P}_1'$ may intersect $\Gamma(s)$, and no demand pair is contained in the disc whose boundary is $\Gamma(s)$, no other path in $\mathcal{P}_1'$ intersects $R \cup L(R)$, and so all such paths are disjoint from $Q(s)$. Consider now some other bad source vertex $s_1 \in S_1$. Note that s_1 cannot lie in R, since in this case the path of $\mathcal{P}_1'$ originating at s_1 would have crossed $\Gamma(s)$. Therefore, s_1 must lie in some other component $R' \in \mathcal{R}$. Then the only way for $Q(s)$ and $Q(s_1)$ to intersect is when $s' = s_1'$. In particular, R, R' should both belong to some set $\mathcal{R}_h$ for $1 \leq h \leq 2\Delta_0$. But since the segments $\{\sigma(R) \mid R \in \mathcal{R}_h\}$ are nested, due to the way we chose the mappings $\beta(s)$ and $\beta(s_1)$, this is impossible.

We can similarly prove that for each bad destination vertex $t \in T_1$, if $P' \in \mathcal{P}_1'$ is the path terminating at t, then $Q(t)$ is disjoint from all paths in $\mathcal{P}_1' \setminus \{P'\}$, and from all paths $Q(t_1)$, where $t_1 \in T_1 \setminus \{t\}$ is a bad destination vertex. Altogether, this proves that the paths in $\tilde{\mathcal{P}}$ are disjoint. $\square$

For each $0 \leq h, h' \leq 2\Delta_0$, let $\tilde{\mathcal{M}}_{h,h'} \subseteq \tilde{\mathcal{M}}$ be the set of all demand pairs $(\tilde{s}, \tilde{t})$ with $\tilde{s} \in Z_h(x)$ and $\tilde{t} \in Z_{h'}(y)$.

If $h = 0$ or $h' = 0$, then, since $|V(C_x)|, |V(C_y)| \leq \Delta$, $\mathsf{OPT}(G, \tilde{\mathcal{M}}_{h,h'}) \leq \Delta$. We route any demand pair in $\tilde{\mathcal{M}}_{h,h'}$ to obtain a factor-Δ approximation to the problem $(G, \tilde{\mathcal{M}}_{h,h'})$. If both $h, h' > 0$, then we apply Theorem 7.2 to obtain a collection $\mathcal{P}_{h,h'}$ of at least $\Omega\left(\frac{\mathsf{OPT}(G,\tilde{\mathcal{M}}_{h,h'})}{\Delta_0^2 \log n}\right)$ disjoint paths, routing demand pairs in $\tilde{\mathcal{M}}_{h,h'}$. We then take the best among all resulting solutions.

Notice that $\left\{\tilde{\mathcal{M}}_{h,h'} \mid 0 \leq h, h' \leq 2\Delta_0\right\}$ partition the set $\tilde{\mathcal{M}}$ of demand pairs, and so there is a pair $0 \leq h, h' \leq 2\Delta_0$ of indices with $\mathsf{OPT}(G, \tilde{\mathcal{M}}_{h,h'}) \geq \frac{\mathsf{OPT}(G,\tilde{\mathcal{M}})}{(2\Delta_0+1)^2} \geq \Omega\left(\frac{\mathsf{OPT}(G,\mathcal{N})}{\Delta_0^3}\right)$. Therefore, we obtain a routing of at least $\Omega\left(\frac{\mathsf{OPT}(G,\mathcal{N})}{\Delta_0^6 \log n}\right)$ demand pairs.

7.2 Subcase 2b: $d(x, y) \leq 5\Delta_0$

We again start by defining a special case of the problem, which is similar to the problem of routing on a disc. We show an approximation algorithm for this special case that reduces it to the problem of routing on a disc, and we later use this special case in order to handle the general problem in Case 2b.

A Special Case

Suppose we are given a connected planar graph $\hat{G}$ embedded on the sphere, a cycle Z in $\hat{G}$, and a set $\hat{\mathcal{M}}$ of demand pairs, such that each terminal of $\mathcal{T}(\hat{\mathcal{M}})$ lies on Z. Assume additionally that we are given a closed $\hat{G}$-normal curve C of length at most Δ, that is disjoint from Z. Let $D(Z)$ be the disc whose boundary is Z, which contains C. Assume further that for every vertex $v \in Z$, there is a $\hat{G}$-normal curve $\gamma(v)$ of length at most $16\Delta_0$ connecting v to a vertex of C, so that $\gamma(v)$ is contained in $D(Z)$ and it is internally disjoint from C. This finishes the definition of the special case.

Next, we reduce this special case to routing on a disc, by creating a hole in the sphere. The hole is $D^{\circ}(Z)$, so we delete all edges and vertices that appear inside $D(Z)$, except for the edges and the vertices of Z. Let $\hat{G}'$ be the resulting graph. We can now apply the $O(\log n)$-approximation algorithm for NDP-Disc to the resulting problem, to obtain a routing of at least $\Omega\left(\frac{\mathsf{OPT}(\hat{G}',\hat{\mathcal{M}})}{\log n}\right)$ demand pairs. In order to complete the analysis of the algorithm, we prove that $\mathsf{OPT}(\hat{G}', \hat{\mathcal{M}}) \geq \Omega\left(\frac{\mathsf{OPT}(\hat{G},\hat{\mathcal{M}})}{\Delta_0^2 \log n}\right)$.

Theorem 7.7 $\mathsf{OPT}(\hat{G}', \hat{\mathcal{M}}) \geq \Omega\left(\frac{\mathsf{OPT}(\hat{G},\hat{\mathcal{M}})}{\Delta_0^2 \log n}\right)$.

Proof: As before, we can assume without loss of generality that all curves in set $\{\gamma(v) \mid v \in V(Z)\}$ are mutually non-crossing, and for all $v, v' \in V(Z)$, $\gamma(v) \cap \gamma(v')$ is a contiguous curve that has a vertex of C as its endpoint.

Consider the optimal solution to instance $(\hat{G}, \hat{\mathcal{M}})$, and let $\mathcal{P}_0$ be the set of paths in this solution. As before, we will gradually modify the set $\mathcal{P}_0$ of paths, to obtain path sets $\mathcal{P}_1, \mathcal{P}_2, \ldots$, until we obtain a feasible solution to instance $(\hat{G}', \hat{\mathcal{M}})$. For every $i \geq 0$, we will denote by $\hat{\mathcal{M}}_i$ the set of the demand pairs routed by $\mathcal{P}_i$, and by κ_i its cardinality. Recall that $\kappa_0 = \mathsf{OPT}(\hat{G}, \hat{\mathcal{M}})$. We assume that $\kappa_0 \geq 2^{13}\Delta_0^2 \log n$, as otherwise a solution routing a single demand pair gives the desired approximation, and such a solution exists in $\hat{G}'$, as it must be connected.

We delete from $\mathcal{P}_0$ all paths that use the vertices of C. Since C contains at most Δ vertices, we delete at most Δ paths in this step. Let $\mathcal{P}_1$ be the resulting set of paths, and $\hat{\mathcal{M}}_1$ the set of the demand pairs routed by $\mathcal{P}_1$.

Our next step is to build a conflict graph H, almost exactly as before. The set of vertices is $V(H) = \left\{v(s,t) \mid (s,t) \in \hat{\mathcal{M}}_1\right\}$. There is a directed edge from $v(s,t)$ to $v(s',t')$, iff the path $P(s,t) \in \mathcal{P}_1$ routing the pair (s,t) contains a vertex of $V(\gamma(s')) \cup V(\gamma(t'))$, and we say that there is a conflict between (s,t) and (s',t') in this case. Since we assume that the paths in $\mathcal{P}_1$ are node-disjoint, and since $|V(\gamma(s'))|, |V(\gamma(t'))| \leq 16\Delta_0$ for all $(s',t') \in \hat{\mathcal{M}}_1$, the in-degree of every vertex in H is at most $32\Delta_0$. As before, there is an independent set $I \subseteq V(H)$ of cardinality at least $\frac{|\mathcal{P}_1|}{32\Delta_0+1} \geq \frac{|\mathcal{P}_0|}{64\Delta_0}$ in H.

Let $\hat{\mathcal{M}}_2$ be the set of all demand pairs (s,t) with $v(s,t) \in I$, and let $\mathcal{P}_2 \subseteq \mathcal{P}_1$ be the set of paths routing the demand pairs in $\hat{\mathcal{M}}_2$. Recall that the paths in $\mathcal{P}_2$ are disjoint from C. Moreover, if $P(s,t) \in \mathcal{P}_2$ is the path routing the pair $(s,t) \in \hat{\mathcal{M}}_2$, then for every demand pair $(s',t') \neq (s,t)$ in $\hat{\mathcal{M}}_2$, path $P(s,t)$ is disjoint from both $\gamma(s')$ and $\gamma(t')$. It is now easy to verify that the demand pairs in $\hat{\mathcal{M}}_2$ are non-crossing with respect to the cycle Z.

We now depart from the proof of Theorem 7.2. We use Lemma 2.3, in order to compute a partition $(\mathcal{N}_1, \ldots, \mathcal{N}_{4\lceil \log n \rceil})$ of the set $\hat{\mathcal{M}}_2$ of the demand pairs, so that for all $1 \leq a \leq 4\lceil \log n \rceil$, set $\mathcal{N}_a$ is r_a-split, for some $r_a \geq 0$. Then there must be an index $1 \leq a \leq 4\lceil \log n \rceil$, such that $|\mathcal{N}_a| \geq \frac{|\hat{\mathcal{M}}_2|}{4\lceil \log n \rceil} \geq \Omega\left(\frac{|\mathcal{P}_0|}{\Delta_0 \log n}\right)$. We let $\hat{\mathcal{M}}_3 = \mathcal{N}_a$, and $\mathcal{P}_3 \subseteq \mathcal{P}_2$ the set of paths routing the demand pairs in $\hat{\mathcal{M}}_3$. We then denote r_a by ρ, and the partition of $\hat{\mathcal{M}}_3$ associated with the definition of the ρ-split set of demand pairs by $\mathcal{M}^1, \ldots, \mathcal{M}^\rho$. We also denote by $\Sigma = (\sigma_1, \ldots, \sigma_{2\rho})$ the corresponding partition of Z into intervals. We assume without loss of generality that for all $1 \le z \le \rho$, all source vertices of the demand pairs in $\mathcal{M}^z$ lie on σ_{2z-1}, and all corresponding destination vertices lie on σ_{2z}.

Let I_1 contain all indices $1 \le z \le \rho$, with $|\mathcal{M}^z| \le 128\Delta_0$, and let I_2 contain all remaining indices. If $\sum_{z \in I_1} |\mathcal{M}^z| \ge |\hat{\mathcal{M}}_3|/2$, then we can obtain a routing of at least $\frac{|\hat{\mathcal{M}}_3|}{256\Delta_0} \ge \Omega\left(\frac{|\mathcal{P}_0|}{\Delta_0^2 \log n}\right)$ demand pairs, as follows: for each $z \in I_1$, we route any demand pair in $\mathcal{M}^z$ via the segment $\sigma_{2z-1} \cup \sigma_{2z}$ of Z. Therefore, we assume from now on that $\sum_{z \in I_2} |\mathcal{M}^z| \ge \frac{|\hat{\mathcal{M}}_3|}{2} \ge \Omega\left(\frac{|\mathcal{P}_0|}{\Delta_0 \log n}\right)$. We denote $\hat{\mathcal{M}}_4 = \bigcup_{z \in I_2} \mathcal{M}^z$, and we let $\mathcal{P}_4 \subseteq \mathcal{P}_3$ be the set of paths routing the demand pairs in $\hat{\mathcal{M}}_4$. For all $z \in I_2$, we denote $|\mathcal{M}^z|$ by κ_4^z.

The rest of the proof is very similar to the rest of the proof of Theorem 7.2, except that now we deal with each subset $\mathcal{M}^z$ for $z \in I_2$ of the demand pairs separately. Fix some $z \in I_2$, and denote $\mathcal{M}^z = \left\{(s_1^z, t_1^z), \ldots, (s_{\kappa_4^z}^z, t_{\kappa_4^z}^z)\right\}$. Since the demand pairs in $\mathcal{M}^z$ are non-crossing, and due to the definition of the ρ-split instance, we can assume without loss of generality that $s_1^z, \ldots, s_{\kappa_4^z}^z, t_{\kappa_4^z}^z, \ldots, t_1^z$ appear in this counter-clock-wise order on Z. Let $\kappa_5^z = \left\lfloor \frac{\kappa_4^z}{64\Delta_0} \right\rfloor - 1$. Since $\kappa_4^z \ge 128\Delta_0$, $\kappa_5^z \ge \frac{\kappa_4^z}{256\Delta_0}$. We then let $\hat{\mathcal{M}}^z$ contain all demand pairs $(s_{64\Delta_0 r}^z, t_{64\Delta_0 r}^z)$, for $1 \le r \le \kappa_5^z$, and we let $\hat{\mathcal{M}}_5 = \bigcup_{z \in I_2} \hat{\mathcal{M}}^z$. Notice that $\kappa_5 = |\hat{\mathcal{M}}_5| = \sum_{z \in I_2} \kappa_5^z \ge \sum_{z \in I_2} \frac{\kappa_4^z}{256\Delta_0} \ge \frac{\kappa_4}{256\Delta_0} \ge \Omega\left(\frac{\kappa_0}{\Delta_0^2 \log n}\right)$. We now show that all demand pairs in $\hat{\mathcal{M}}_5$ can be routed in graph $\hat{G}'$. This is done similarly to the proof of Theorem 7.2.

Fix some $z \in I_2$, and consider the set $\hat{\mathcal{M}}^z = \left\{(s_{64\Delta_0 r}^z, t_{64\Delta_0 r}^z) \mid 1 \le r \le \kappa_5^z\right\}$ of demand pairs. For each such demand pair $(s_{64\Delta_0 r}^z, t_{64\Delta_0 r}^z)$, we define two segments $\mu_r^z \subseteq \sigma_{2z-1}$ and $\tilde{\mu}_r^z \subseteq \sigma_{2z}$ of Z, as follows. For brevity of notation, denote $64\Delta_0 r$ by ℓ. The first segment, μ_r^z, is the segment of Z from $s_{\ell-32\Delta_0}^z$ to $s_{\ell+32\Delta_0-1}^z$, as we traverse Z in the counter-clock-wise order. The second segment, $\tilde{\mu}_r^z$, is the segment of Z from $t_{\ell-32\Delta_0}^z$ to $t_{\ell+32\Delta_0-1}^z$, as we traverse Z' in the clock-wise order. It is immediate to verify that all segments of Z in $\{\mu_r^z, \tilde{\mu}_r^z \mid z \in I_2, 1 \le r \le \kappa_5^z\}$ are mutually disjoint. We use the following analogue of Lemma 7.3.

Lemma 7.8 *For every $z \in I_2$, for every demand pair $(s_{64\Delta_0 r}^z, t_{64\Delta_0 r}^z) \in \hat{\mathcal{M}}^z$, if $P \in \mathcal{P}_5$ is the path routing this pair, then $P \cap Z \subseteq \mu_r^z \cup \tilde{\mu}_r^z$.*

We can now use this lemma to re-route the paths in $\mathcal{P}_5$, similarly to the proof of Theorem 7.1. Fix some $z \in I_2$, and let $(s_{64\Delta_0 r}^z, t_{64\Delta_0 r}^z) \in \hat{\mathcal{M}}^z$ be some demand pair, with $1 \le r \le \kappa_5^z$. For simplicity, we denote $\ell = 64\Delta_0 r$. Let P_ℓ be the path routing this demand pair in $\mathcal{P}_5$, that we view as directed from s_ℓ towards t_ℓ. Clearly, P_ℓ intersects both μ_r^z and $\tilde{\mu}_r^z$. Let v_ℓ be the last vertex of P_ℓ lying on μ_r^z. Then there is some other vertex appearing on P_ℓ after v_ℓ that belongs to $\tilde{\mu}_r^z$. We let v'_ℓ be the first such vertex on P_ℓ, and we let P'_ℓ be the segment of P_ℓ between v_ℓ and v'_ℓ. Let P^*_ℓ be the path obtained as follows: we start with a segment of μ_r^z between s_ℓ^z and v_ℓ; we then follow P'_ℓ to v'_ℓ, and finally we use a segment of $\tilde{\mu}_r^z$ between v'_ℓ and t_ℓ^z. From Lemma 7.8, it is immediate to verify that we obtain a set of node-disjoint paths routing all demand pairs in $\hat{\mathcal{M}}_5$ in graph $\hat{G}'$. It now remains to prove Lemma 7.8.

Proof: Fix some $z \in I_2$, and consider some demand pair $(s_{64\Delta_0 r}^z, t_{64\Delta_0 r}^z) \in \hat{\mathcal{M}}^z$. Let $P \in \mathcal{P}_5$ be the path routing this pair. We partition the cycle Z into two segments: σ containing σ_{2z-1} and σ' containing σ_{2z} arbitrarily, so σ now contains all source vertices, and σ' now contains all destination vertices of the pairs in $\mathcal{M}^z$. It is enough to show that $P \cap \sigma \subseteq \mu_r^z$ and $P \cap \sigma' \subseteq \tilde{\mu}_r^z$. We prove that $P \cap \sigma \subseteq \mu_r^z$. The proof that $P \cap \sigma' \subseteq \tilde{\mu}_r^z$ is symmetric.

Assume for contradiction that $P \cap \sigma \not\subseteq \mu_r^z$, and let v be the first vertex on P that belongs to $\sigma \setminus \mu_r^z$, where we view P as directed from s_ℓ^z to t_ℓ^z. Let P' be the sub-path of P from its first vertex to v.

Consider the planar embedding of $\hat{G}$, where we fix any face lying outside of $D(Z)$ as the outer face. In this planar embedding, denote by Y the union of $D(C), \gamma(s^z_\ell), P'$, and $\gamma(v)$, and let R be the outer boundary of Y (see Figure 7).

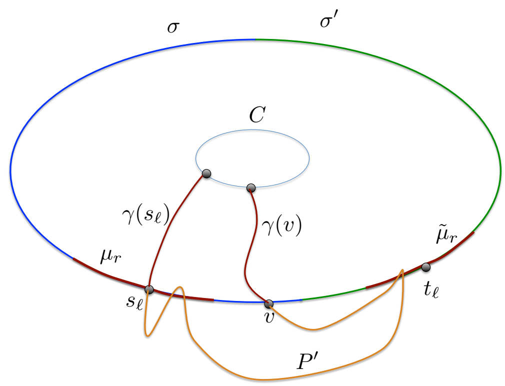


Figure 7: Constructing the curve R. The segments σ and σ' of Z are shown in blue and green, respectively. We omit the superscript z for brevity of notation.

Notice that, since there are no conflicts in $\mathcal{P}_4$, curve R does not cross any curve in the set $C \cup \left\{\gamma(t) \mid t \in \mathcal{T}(\hat{\mathcal{M}}_4) \setminus \{s^z_\ell, t^z_\ell\}\right\}$, and so, since both s^z_ℓ and v lie on σ, all destination vertices of the demand pairs in $\mathcal{M}^z$, except for possibly t^z_ℓ, lie on one side of R. Let $S_1 = \left\{s^z_{\ell-32\Delta_0}, \ldots, s^z_{\ell-1}\right\}$, and $S_2 = \left\{s^z_{\ell+1}, \ldots, s^z_{\ell+32\Delta_0-1}\right\}$. Denote $\Gamma_1 = \{\gamma(s^z_i) \mid s^z_i \in S_1\}$ and $\Gamma_2 = \{\gamma(s^z_i) \mid s^z_i \in S_2\}$. Since the curves in set $\{\gamma(u) \mid u \in V(Z)\}$ are non-crossing, if β, β' are the two segments of Z whose endpoints are s_ℓ and v, then all vertices of $S_1 \setminus \{v\}$ lie on one of the segments (say β), while all vertices of $S_2 \setminus \{v\}$ lie on the other segment. Moreover, since path P' cannot cross any curve in $\Gamma_1 \cup \Gamma_2$, the vertices of $S_1 \setminus \{v\}$ lie on one side of R, and the vertices of $S_2 \setminus \{v\}$ lie on the other side of R. Therefore, either all vertices of $S_1 \setminus \{v\}$ are separated by R from all destination vertices of the demand pairs in $\mathcal{M}^z$ (except for possibly t^z_ℓ), or the same holds for $S_2 \setminus \{v\}$ - we assume without loss of generality that it is the former.

All sources of $S_1 \setminus \{v\}$ are then separated by R from their destinations vertices, and yet all corresponding demand pairs are routed by $\mathcal{P}_2$. Therefore, at least $32\Delta_0 - 2$ paths in $\mathcal{P}_2$ must cross the curve R. Recall that none of these paths can cross C, P', or $\gamma(s^z_\ell)$. Therefore, all these paths must cross $\gamma(v)$. But since the length of $\gamma(v)$ is at most $16\Delta_0$, and the paths are node-disjoint, this is impossible. □ □

The rest of the proof follows the same strategy as the proof for Case 2a, but it is somewhat more involved. We break it into three steps. In the first step, we construct a single shell around the vertex x. In the second step, we map the terminals to the cycles of the shell. In the final step, we reduce the problem to the special case, by constructing a cycle Z and mapping a subset of the terminals to the vertices of Z.

We assume that $\mathsf{OPT}(G, \mathcal{N}) > 2^{13}\Delta_0^4$, since otherwise we can route a single demand pair and obtain a valid solution.

Step 1: the Shell

In this step we construct a shell $\mathcal{Z}(x) = (Z_1(x), \ldots, Z_h(x))$ of depth $h \leq 8\Delta_0$ around x. We first give a high-level explanation of why the construction of the shell in this case is more challenging than that in Case 2a, and motivate our construction. Recall that given such a shell $\mathcal{Z}(x)$, we have defined, for all $1 \leq h' \leq h$, a set $U_{h'}$ of vertices contained in $D^\circ(Z_{h'}(x)) \setminus D(Z_{h'-1}(x))$, and a set $\mathcal{R}_{h'}$ of all connected components of $G[U_{h'}]$. For all $1 \leq h' \leq h$, for every component $R \in \mathcal{R}_{h'}$, we have defined the disc $\eta(R)$ (given by Theorem 5.5), whose boundary $\gamma(R)$ served as the curve $\Gamma(t)$ for all terminals t lying in R. In Case 2a, discs $\eta(R)$ had the important property that no demand pair in $\mathcal{N}$ is contained in $\eta(R)$. This ensured that whenever a path routing any demand pair in $\mathcal{N}$ intersects R, such a path must also cross $\gamma(R)$. This latter property was crucial in ensuring that, after we map all terminals to the vertices of the shell, the solution value does not change by much. Unfortunately, in Case 2b this is no longer true, and it is possible that discs $\eta(R)$, and even the components R themselves, contain many demand pairs from $\mathcal{N}$. We get around this problem as follows. We construct the shell around x carefully, to ensure that the total number of the terminals contained in each such disc $\eta(R)$ is relatively small. We say that a pair of terminals is bad if there is a path connecting these terminals, which is completely contained in some disc $\eta(R)$, and it is good otherwise. As long as bad terminal pairs exist in our graph, we iteratively route one such pair inside the corresponding disc $\eta(R)$, and discard all other terminals lying in this disc. Since the total number of the terminals contained in each disc is relatively small, the number of the terminals we discard at this step is relatively small compared to the number of the demand pairs we route. If we manage to route a large enough number of demand pairs in this step, then we terminate the algorithm and return this solution. Otherwise, we let $\mathcal{N}'$ be the subset of the demand pairs that have not been routed or discarded yet. Then $|\mathcal{N}'|$ is sufficiently large relatively to $|\mathcal{N}|$, and we have now achieved the property that for all components R, any path connecting a demand pair in $\mathcal{N}'$ that intersects R must also cross $\gamma(R)$. We now proceed to describe the shell construction. Let $\tau^* = 64\Delta_0/\alpha_{\text{WL}}$, and let $\mathcal{T}$ be the set of all terminals participating in the demand pairs in $\mathcal{N}$.

For all integers $i > 0$, let S_i be the set of all vertices $v \notin D_x$, with $d_{\text{GNC}}(v, V(C_x)) \geq i$. We say that a connected component R of $G[S_i]$ is *heavy* iff R contains more than τ^* terminals of $\mathcal{T}$. Let i^* be the largest integer, such that $G[S_{i^*}]$ contains at least one heavy connected component, and let R^* be any such component. We need the following easy observation.

Observation 7.9 $i^* \leq 8\Delta_0 - \Delta - 4$.

Proof: Assume otherwise and let $h = 8\Delta_0 - \Delta - 4$. Then there is some connected component R of $G[S_h]$, containing more than τ^* terminals of $\mathcal{T}$. Let $t \in V(R) \cap \mathcal{T}$ be any such terminal. If $t \in X$, then $d(t, x) \leq \Delta_0$. If $t \in Y$, then $d(t, y) \leq \Delta_0$, and $d(x, y) \leq 5\Delta_0$, so from Observation 5.4, $d(t, x) \leq 6\Delta_0 + \Delta/2 + 1$. Therefore, some vertex $v \in C_t$ has a G-normal curve of length at most $6\Delta_0 + \Delta/2 + 1$ connecting it to a vertex of C_x, and so every vertex in C_t has a G-normal curve of length at most $6\Delta_0 + \Delta/2 + 1 + \Delta/2 + 1 < 8\Delta_0 - \Delta - 4$ connecting it to a vertex of C_x. Therefore, $V(C_t) \subseteq V(G) \setminus S_h$, while $t \in S_h$. Since $V(C_t)$ separates $V(D_t)$ from all remaining vertices of G, and R is a connected component of $G[S_h]$, it follows that $R \subseteq V(D_t)$. But D_t contains fewer than τ^* terminals from the definition of enclosures, a contradiction. □

Let $\kappa = |\mathcal{T} \cap R^*|$, so $\kappa > \tau^*$. The following observation will be useful in order to bound the number of terminals contained in each disc $\eta(R)$. The proof follows immediately from the well-linkedness of the terminals.

Observation 7.10 *Let D be any disc, whose boundary γ is a G-normal curve of length less than $16\Delta_0$. Assume further that at least $\kappa/4$ terminals lie outside of D. Then D contains at most τ^* terminals of $\mathcal{T}$.*

We now provide some further intuition. Let $h' = i^* - 1$, and consider a shell $\mathcal{Z}(x) = (Z_1(x), \ldots, Z_{h'}(x))$ of depth h' around x, where we let F_x be any face incident on a vertex of R^*. We can then define, for each $1 \leq h'' \leq h'$, the set $\mathcal{R}_{h''}$ of connected components of the graph induced by all vertices lying in $D^\circ(Z_{h''}(x)) \setminus D(Z_{h''-1}(x))$, set $\mathcal{R} = \bigcup_{h''=1}^{h'} \mathcal{R}_{h''}$, and use Theorem 5.5 in order to compute the discs

$\eta(R)$ for the components $R \in \mathcal{R}$. Moreover, since all vertices of R^* lie outside each such disc $\eta(R)$, from Observation 7.10, each such disc contains at most τ^* terminals. The problem is that R^* may still contain many terminals, while we need to ensure that most of the terminals lie in the components of $\mathcal{R}$. We get around this problem by extending the shell, and adding two outer cycles to it. First, we consider the outer boundary Γ of the graph R^* (in the drawing where the face containing x is viewed as the outer face), and carefully select some cycle $C \in \Gamma$, so that, if we add C as the outer-most cycle to the shell $\mathcal{Z}(x)$, by setting $Z_{h'+1}(x) = C$, then for every component $R \in \mathcal{R}_{h'+1}$, we still maintain the property that the corresponding disc $\eta(R)$ contains at most τ^* terminals. Finally, we take care of the terminals contained in the disc $D(C)$, that currently do not lie in any component of $\mathcal{R}$. The idea is to carefully select one vertex $\tilde{u} \in V(C)$, and to attach a new cycle C' to $\tilde{u}$, that lies "inside" cycle C, and is then added to the shell as the outermost cycle, so $Z_{h'+2}(x) = C'$. We then view the face whose boundary is C' as the face F_x in the shell construction. Therefore, once $\tilde{u}$ is selected and cycle C' is added to the drawing of G, both the construction of the shell, and the construction of the discs $\eta(R)$ for $R \in \mathcal{R}$ are fixed. We would like to select $\tilde{u}$ in such a way that each resulting disc $\eta(R)$ for $R \in \mathcal{R}_{h'+2}$ contains at most τ^* terminals. We achieve this by discarding a small number of terminals and their corresponding demand pairs. We now describe the construction more formally.

Consider the graph R^*, and its drawing in the plane, induced by the drawing of G on the sphere, where we view the face where x used to be as the outer face. Let Γ be the boundary of the outer face in this drawing of R^*, and let $\mathcal{C}$ be the set of all simple cycles in Γ. Let H be the block-decomposition of Γ. That is, the set $V(H)$ of vertices consists of two subsets: set V_1 of cut vertices of Γ, and set V_2, containing a vertex v_B for every block B (a maximal 2-connected component) of Γ. We add an edge (u, v_B) between vertices $u \in V_1$ and $v_B \in V_2$ iff $u \in V(B)$. It is easy to see that graph H is a tree, and we root it at any vertex. We next define weights for the vertices of H. In order to do so, every terminal in $\mathcal{T} \cap R^*$ will contribute a weight of 1 to one of the vertices of V_2, and the weight of every vertex in V_2 is then the total weight contributed to it. The weights of all vertices in V_1 are 0. Consider some terminal $t \in \mathcal{T} \cap R^*$. If there is some simple cycle $C \subseteq \Gamma$, such that $t \in D(C)$, then t contributes the weight of 1 to the vertex v_B, where $B = C$ (if t belongs to several such cycles, then we select one of these cycles arbitrarily). Otherwise, $t \in V(\Gamma)$, and there is some vertex $v_B \in V_2$, such that block B consists of a single edge e, and t is one of its endpoints. Among all such vertices v_B, we choose one arbitrarily, and contribute the weight of t to v_B.

For every subgraph $H' \subseteq H$, the weight of H' is the total weight of all vertices in H'. Clearly, the weight of H is κ. We need the following simple claim.

Claim 7.11 *There is some vertex $u^* \in V(H)$, such that, if we root H at u^*, then for every child u' of u^*, the weight of the sub-tree of H rooted at u' is at most $\kappa/2$.*

Proof: We root H at any vertex v, and set $u = v$. We then iterate. If the current vertex u has a child u', such that the total weight of all vertices contained in the sub-tree of H rooted at u' is more than $\kappa/2$, then we move u to u'. It is easy to see that when this procedure terminates, we will find the desired vertex u^*. □

Consider the vertex u^* computed by the above claim. If $u^* \in V_1$, then let $v^* = u^*$. Otherwise, $u^* = v_B$ for some vertex $v_B \in V_2$, then let v^* be any vertex of B. We assume without loss of generality that v^* lies on some simple cycle $C \subseteq \Gamma$: otherwise, we create an artificial cycle $C = (v^*, u_1, u_2)$, where u_1 and u_2 are two new vertices. If v^* lies on several such cycles, then we let C be any one of them. Notice that every connected component of $R^* \setminus V(D(C))$ contains at most $\kappa/2$ terminals, and has exactly one neighbor in $V(C)$.

Let $\tilde{u}$ be some vertex in $V(C)$ (that we will select later). We then add a new cycle $C' = (v_1, v_2, v_3)$, containing all new vertices, and an edge $e = (\tilde{u}, v_1)$, and draw C' inside C (we later specify the precise location of this drawing). Let $G' = G \cup C' \cup \{e\}$. We let F_x be the face in the drawing of G' on the sphere, whose boundary is C'. Letting $h = i^*$, we construct a shell $\mathcal{Z}(x) = (Z_1(x), \ldots, Z_h(x))$ of depth $h = i^* \leq 8\Delta_0 - \Delta - 4$ around x, with respect to F_x. As before, we use $Z_0(x)$ to denote C_x. Notice that from our construction, $Z_h(x) = C$. We note that the choice of the vertex $\tilde{u} \in V(C)$ to which the cycle C' is attached does not affect the construction of the shell: for any such choice, the shell will be the same. Notice also that the addition of the new cycles does not affect the routings. So abusing the notation we denote G' by G.

For every $1 \leq h' \leq h$, we let $U_{h'}$ be the set of all vertices in $D^{\circ}(Z_{h'}(x)) \setminus D(Z_{h'-1}(x))$, and let $\mathcal{R}_{h'}$ be the set of all connected components of $G[U_{h'}]$. We let U_{h+1} be the set of all vertices lying outside $D(Z_h(x))$ in the planar embedding of G where F_x is the outer face (equivalently, U_{h+1} is the set of all vertices lying in disc $D^{\circ}(C)$ in the planar embedding of G where the face containing x is viewed as the outer face), and denote by $\mathcal{R}_{h+1}$ the set of all connected components of $G[U_{h+1}]$. We will view the components in $\mathcal{R}_{h+1}$ as type-2 components with respect to the shell. For each such component $R \in \mathcal{R}_{h+1}$, we let $L(R) \subseteq V(C)$ be the set of the neighbors of the vertices of R, and we leave $u(R)$ undefined. We need the following simple observation.

Observation 7.12 *For each $R \in \mathcal{R}_{h+1}$, $|V(R) \cap \mathcal{T}| \leq \tau^*$.*

Proof: Assume otherwise, and let $R \in \mathcal{R}_{h+1}$ be any component with $|R \cap \mathcal{T}| > \tau^*$. We claim that all vertices of R lie in S_{i^*+1}. Indeed, recall that all vertices of R^* lie in S_{i^*}, and $C \subseteq R^*$ is a cycle separating R from all vertices of $V(G) \setminus S_{i^*}$. Therefore, for every vertex $v \in V(R)$, $d_{\mathrm{GNC}}(v, V(C_x)) \geq i^* + 1$ and $v \in S_{i^*+1}$. But then R is a heavy connected component in $G[S_{i^*+1}]$, contradicting the choice of i^*. □

Notice that once $\tilde{u}$ and the face F_x are fixed, we can define, for every component $R \in \mathcal{R}_{h+1}$, the segment $\sigma(R)$ of the cycle $C = Z_h(x)$ exactly as before, and we can define the discs $\eta(R)$ for all components $R \in \bigcup_{h'=1}^{h+1} \mathcal{R}_{h'}$ using Theorem 5.5. (It may be convenient to think of C' as the outer-most cycle of the shell; that is, we add C' to the shell as $Z_{h+1}(x)$). The main theorem summarizing the current step is the following.

Theorem 7.13 *There is an efficient algorithm to compute a vertex $\tilde{u} \in V(C)$, a drawing of the cycle C', and a subset $\mathcal{T}' \subseteq \mathcal{T}$ of at most $4\tau^*$ terminals, such that, in the resulting shell $\mathcal{Z}(x) = (Z_1(x), \ldots, Z_h(x))$, for each component $R \in \bigcup_{h'=1}^{h+1} \mathcal{R}_{h'}$, the resulting disc $\eta(R)$ contains at most τ^* terminals of $\mathcal{T} \setminus \mathcal{T}'$.*

We emphasize that the shell $\mathcal{Z} = (Z_1(x), \ldots, Z_h(x))$ and the sets $\mathcal{R}_{h'}$ of components, for $1 \leq h' \leq h+1$ do not depend on our choice of the vertex $\tilde{u}$ or the drawing of C'. Similarly, the discs $\eta(R)$ for components R lying in sets $\mathcal{R}_{h'}$, for $1 \leq h' \leq h$, given by Theorem 5.5, are also independent of the choice of $\tilde{u}$ or the drawing of C' inside C. The choice of $\tilde{u}$ only influences the discs $\eta(R)$ for $R \in \mathcal{R}_{h+1}$, and so our goal is to select $\tilde{u}$ and $\mathcal{T}'$, and to draw C' inside C in a way that ensures that each such disc $\eta(R)$ contains few terminals of $\mathcal{T} \setminus \mathcal{T}'$.

Proof: Our first step processes the components of $\mathcal{R}_{h+1}$ and to select the vertex $\tilde{u} \in V(C)$. For this step, we will think of G as being embedded on the sphere. Let κ' be the number of the terminals of $\mathcal{T}$ contained in the components of $\mathcal{R}_{h+1}$. If $\kappa' \leq 4\tau^*$, then we add all terminals of $\mathcal{T} \cap \left(\bigcup_{R \in \mathcal{R}_{h+1}} V(R)\right)$ to $\mathcal{T}'$, and terminate the algorithm, setting $\tilde{u}$ to be any vertex of C, and drawing C' anywhere inside C, so the resulting drawing of G is planar. (We show below that this choice satisfies the conditions of the theorem). Therefore, we assume from now on that $\kappa' \geq 4\tau^*$, and in particular $|\mathcal{R}_{h+1}| \geq 1$.

Throughout the algorithm, we maintain a partition of $\mathcal{R}_{h+1}$ into two subsets: $\mathcal{R}'$, containing all components we have processed, and $\mathcal{R}''$, containing all remaining components. At the beginning, $\mathcal{R}' = \emptyset$ and $\mathcal{R}'' = \mathcal{R}_{h+1}$.

Consider any component $R \in \mathcal{R}''$. Recall that all neighbors of the vertices of R must lie on C from our construction, and we denoted the set of these vertices by $L(R)$.

We say that R is a good component, iff there is a segment $\mu(R)$ of C containing all vertices of $L(R)$, whose endpoints, denoted by $a_1(R)$ and $a_2(R)$ belong to $L(R)$, and there is a G-normal curve $\gamma'(R)$, whose endpoints are $a_1(R)$ and $a_2(R)$, such that $\gamma'(R)$ is internally disjoint from $V(G)$, and the following holds: let $\eta'(R)$ be the disc, whose boundary is $\mu(R) \cup \gamma'(R)$, with $R \subseteq \eta'(R)$. Then for all $R' \in \mathcal{R}''$ with $R' \neq R$, R' is disjoint from $\eta'(R)$ (intuitively, these are the components that lie closest to C; see Figure 8).

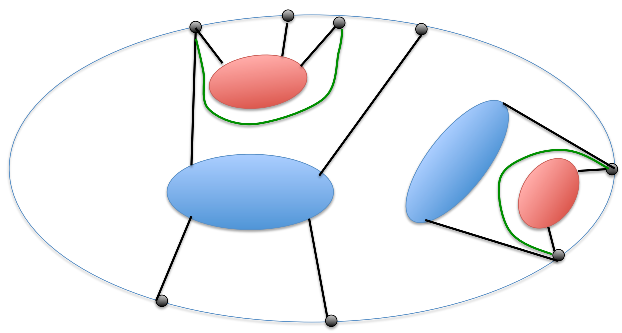


Figure 8: Good components are shown in red, and their corresponding curves $\gamma'(R)$ in green

Claim 7.14 *If $|\mathcal{R}''| \geq 2$, then there are at least two good components in $\mathcal{R}''$.*

Proof: Let μ be the shortest segment of C, such that for some $R \in \mathcal{R}''$, $L(R) \subseteq \mu$. Let $\tilde{\mathcal{R}} \subseteq \mathcal{R}''$ be the set of all components R with $L(R) \subseteq \mu$. If, for any component $R \in \tilde{\mathcal{R}}$, $|L(R)| \geq 3$, then from the definition of μ it is easy to see that R is a good component. If any component $R \in \tilde{\mathcal{R}}$ has $|L(R)| = 1$, then μ contains a single vertex - the unique vertex of $L(R)$, and it is easy to see that R is a good component. Otherwise, every component in $\tilde{\mathcal{R}}$ has $|L(R)| = 2$. We then let R be the component that lies closest to μ. In other words, we choose the unique component $R \in \tilde{\mathcal{R}}$, such that for some vertex $v \in V(R)$, there is a curve γ connecting v to an inner point of μ, so that γ does not contain any vertices of G except for v, and it does not intersect the edges of G. It is immediate to verify that R is a good component. We conclude that there is at least one good component $R \in \mathcal{R}''$, with $L(R) \subseteq \mu$.

Let a, a' be the endpoints of μ, and let μ' be the other segment of C whose endpoints are a and a'. Since $|\mathcal{R}''| \geq 2$, there is at least one component $R' \neq R$ in $\mathcal{R}''$, with $L(R') \subseteq \mu'$. As before, we let μ'' be the shortest (inclusion-wise) segment of μ', such that for some component $R'' \in \mathcal{R}''$, $L(R'') \subseteq \mu''$. Let $\tilde{\mathcal{R}}'$ be the set of all components $R'' \neq R$ in $\mathcal{R}''$ with $L(R'') \subseteq \mu''$. Using the same arguments as above, we can find a second good component in $\mathcal{R}''$. □

We are now ready to describe our algorithm. We start with $\mathcal{R}'' = \mathcal{R}_{h+1}$ and $\mathcal{R}' = \emptyset$. Recall that $|\mathcal{R}''| \geq 1$ must hold. While $|\mathcal{R}''| \geq 2$, let $R, R' \in \mathcal{R}''$ be two distinct good components in $\mathcal{R}''$. Notice that $\mathcal{T} \cap \eta'(R)$ and $\mathcal{T} \cap \eta'(R')$ are completely disjoint. Therefore, either $|\mathcal{T} \cap \eta'(R)| \leq \kappa'/2$, or $|\mathcal{T} \cap \eta'(R')| \leq \kappa'/2$. We assume without loss of generality it is the former. We then move R from $\mathcal{R}''$ to $\mathcal{R}'$, and continue to the next iteration.

Notice that in every iteration of the algorithm, $|\mathcal{R}''|$ decreases by 1. The algorithm terminates when $|\mathcal{R}''| = 1$. Let R be the remaining component in $\mathcal{R}''$. We set $\mathcal{T}' = \mathcal{T} \cap V(R)$, and we set $\tilde{u}$ to be any vertex of $L(R)$. Recall that from Observation 7.12, $|\mathcal{T}'| \leq \tau^*$. We then add the cycle C' that attaches to $\tilde{u}$ with an edge e to G, and we draw C' inside C, so it is drawn next to the edge e. We then obtain a drawing of the resulting graph in the plane where the outer face F_x is the face whose boundary is

C'. We construct the shells, and discs $\eta(R)$ for $R \in \bigcup_{h'=1}^{h+1} \mathcal{R}_{h'}$ as described above. From the choice of $\tilde{u}$, the drawing of C', and the construction of the discs $\eta(R)$, for every $R \in \mathcal{R}'$, $\eta'(R) \subseteq \eta(R)$, and $\eta(R) \cap (D(C') \setminus D(C)) = \eta'(R)$. Therefore, for each $R \in \mathcal{R}'$, at least $\kappa'/2$ terminals of $\mathcal{T}'$ lie outside $\eta(R)$. The following claim will finish the proof of the theorem.

Claim 7.15 *For all $1 \leq h' \leq h+1$, for all $R \in \mathcal{R}_{h'}$, $\eta(R)$ contains at most τ^* terminals of $\mathcal{T} \setminus \mathcal{T}'$.*

Proof: Fix some $1 \leq h' \leq h+1$ and $R \in \mathcal{R}_{h'}$. From our construction, the length of the boundary of $\eta(R)$ is bounded by $2h + 3 + \Delta/2 \leq 2(8\Delta_0 - \Delta) + \Delta/2 < 16\Delta_0$.

Assume first that $h' < h$. Then all vertices of $\mathcal{T} \cap R^*$ lie outside $D(Z_{h'}(x))$, and since $\eta(R) \subseteq D(Z_{h'}(x))$, they also lie outside $\eta(R)$. From Observation 7.10, $\eta(R)$ contains at most τ^* terminals of $\mathcal{T}$.

Assume now that $h' = h$. Notice that for every component $R' \in \mathcal{R}_h$, either R' is disjoint from R^*, or $R' \subseteq R^*$. In the latter case, R' is a type-1 or a type-3 component (see Figure 1(a)), as all type-2 components are disconnected from R^* in $G[S_{i^*}]$. Recall that $R' \in \mathcal{R}_h$ is contained in $\eta(R)$ only if $\sigma(R') \subseteq \sigma(R)$, and this can only happen if R' is a type-2 component. Let $\tilde{\mathcal{R}} \subseteq \mathcal{R}_h$ be the set of all components contained in $\eta(R)$. Then at most one of these components $R' \in \tilde{\mathcal{R}}$ may be contained in R^* - and if such a component exists, then $R' = R$. Our construction of the cycle C and the choice of F_x guarantee that R' contains at most $\kappa/2 + 1$ terminals of $\mathcal{T} \cap V(R^*)$, and so at least $\kappa/2 - 1 \geq \kappa/4$ terminals lie outside $\eta(R)$. Since the length of $\gamma(R)$ is less than $16\Delta_0$, from Observation 7.10, $\eta(R)$ contains at most τ^* terminals of $\mathcal{T}$.

Finally, assume that $h' = h+1$. If $\kappa' \leq 4\tau^*$, then $V(R) \cap (\mathcal{T} \setminus \mathcal{T}') = \emptyset$. Therefore, we assume that $\kappa' > 4\tau^*$. If $R \cap (\mathcal{T} \setminus \mathcal{T}') \neq \emptyset$, then R was added to $\mathcal{R}'$ at some iteration of the algorithm. From the above discussion, at least $\kappa'/2 \geq 2\tau^*$ terminals of $\mathcal{T}$ lie outside of $\eta(R)$, while the length of the boundary of $\eta(R)$ is less than $16\Delta_0$. From the well-linkedness of the terminals, $\eta(R)$ contains at most τ^* terminals of $\mathcal{T}$. □ □

We partition $\mathcal{N}$ into two subsets: set $\mathcal{M}^0$ contains all demand pairs in which the terminals of $\mathcal{T}'$ participate, together with all demand pairs for which either terminal lies in D_x. Then $|\mathcal{M}^0| \leq 4\tau^* + 4\Delta/\alpha_{\text{WL}} \leq 5\tau^*$ (we have used the definition of enclosures to bound the number of the demand pairs of the latter kind, and the fact that $\tau^* = 64\Delta_0/\alpha_{\text{WL}}$). Let $\mathcal{M}^1$ contain the remaining demand pairs, and let $\mathcal{R} = \bigcup_{h'=1}^{h+1} \mathcal{R}_{h'}$. As we noted before, we would like to ensure that for every component $R \in \mathcal{R}$ that contains a terminal of $\mathcal{T}(\mathcal{M}^1)$, whenever a path P routing a demand pair in $\mathcal{M}^1$ intersects R, it must cross $\gamma(R)$ - the boundary of $\eta(R)$. We achieve this property by routing a subset of the demand pairs, and discarding some additional demand pairs, in the following theorem.

Theorem 7.16 *There is an efficient algorithm to compute a partition $(\mathcal{N}_0, \mathcal{N}_1, \mathcal{N}_2)$ of $\mathcal{M}^1$, and a collection $\mathcal{P}^*$ of node-disjoint paths routing at least $|\mathcal{N}_1|/(h+1)$ demand pairs in $\mathcal{N}_1$ in graph G, such that:*

- *$|\mathcal{N}_0| \leq \tau^* |\mathcal{N}_1|$; and*
- *for every component $R \in \mathcal{R}$ with $R \cap \mathcal{T}(\mathcal{N}_2) \neq \emptyset$, for every path P routing a demand pair in $\mathcal{N}_2$, $P \not\subseteq \eta^{\circ}(R)$.*

Proof: We start with $\mathcal{N}_0 = \mathcal{N}_1 = \emptyset$, $\mathcal{P} = \emptyset$, and $\mathcal{N}_2 = \mathcal{M}^1$. Throughout the algorithm, we say that a component $R \in \mathcal{R}$ is a live component iff $R \cap \mathcal{T}(\mathcal{N}_2) \neq \emptyset$. While there is any demand pair $(s,t) \in \mathcal{N}_2$, and any live component $R \in \mathcal{R}$, such that some path P connecting s to t is contained in $\eta^{\circ}(R)$, we do the following. We add P to $\mathcal{P}$, and we move (s,t) from $\mathcal{N}_2$ to $\mathcal{N}_1$. We say that the component R is *responsible for* P, and we say that P is a level-h' path if $R \in \mathcal{R}_{h'}$. Next, for every live component $R' \in \mathcal{R}_{h'}$, such that P intersects $\eta^\circ(R')$, we move all demand pairs $(s',t') \in \mathcal{N}_2$ with $\{s',t'\} \cap \eta^\circ(R') \neq \emptyset$ from $\mathcal{N}_2$ to $\mathcal{N}_0$. The crux of the analysis of this algorithm is in the following claim.

Claim 7.17 *In every iteration of the algorithm, the number of the demand pairs moved to $\mathcal{N}_0$ is at most τ^*.*

Proof: Consider some iteration of the algorithm, where a path P that belongs to level h' was added to $\mathcal{P}$, and let $R \in \mathcal{R}_{h'}$ be the component responsible for it. Recall that $P \subseteq \eta^\circ(R)$, and recall that Theorem 5.5 guarantees that all discs $\eta(R')$ for $R' \in \mathcal{R}_{h'}$ are laminar: that is, for $R', R'' \in \mathcal{R}_{h'}$, either $\eta(R') \subseteq \eta(R'')$, or $\eta(R'') \subseteq \eta(R')$, or $\eta^\circ(R') \cap \eta^\circ(R'') = \emptyset$. Let $\mathcal{R}' \subseteq \mathcal{R}_{h'}$ be the set of all live components R' with $P \cap \eta^\circ(R') \neq \emptyset$. Since $P \subseteq \eta^\circ(R)$, there is some component $R' \in \mathcal{R}'$, such that $P \subseteq \eta^\circ(R')$, and for every component $R'' \in \mathcal{R}'$, $\eta^\circ(R'') \subseteq \eta^\circ(R')$. Therefore, all demand pairs moved in this step from $\mathcal{N}_2$ to $\mathcal{N}_0$ have at least one terminal lying in $\eta^\circ(R')$, and from Theorem 7.13, their number is bounded by τ^*. □

Therefore, once the algorithm terminates, $|\mathcal{N}_0| \leq \tau^*|\mathcal{N}_1|$ must hold. Moreover, for every component $R \in \mathcal{R}$ with $R \cap \mathcal{T}(\mathcal{N}_2) \neq \emptyset$, for every path P routing a demand pair in $\mathcal{N}_2$, $P \not\subseteq \eta^\circ(R)$. Consider now the set $\mathcal{P}$ of paths. Each of these paths belongs to one of $h+1$ levels, and so there is a subset $\mathcal{P}^* \subseteq \mathcal{P}$ of paths that belong to the same level, say h', such that $|\mathcal{P}^*| \geq |\mathcal{P}|/(h+1)$. The paths in $\mathcal{P}^*$ route a subset of the demand pairs in $\mathcal{N}_1$, and it is now enough to show that they are node-disjoint. Assume for contradiction otherwise, and let $P, P' \in \mathcal{P}^*$ be two distinct paths that are not disjoint. Assume that P was added to $\mathcal{P}$ before P'. Let $R' \in \mathcal{R}_{h'}$ be the component responsible for P', so $P' \subseteq \eta^\circ(R')$, and R' was live when R was processed. Then path P must intersect $\eta^\circ(R')$, and so all demand pairs that have a terminal in $\eta^\circ(R')$, including the demand pair routed by P', should have been removed from $\mathcal{N}_2$ during the iteration when P was added to $\mathcal{P}$, a contradiction. □

Recall that $\tau^* = 64\Delta_0/\alpha_{\mathrm{WL}}$. We now consider two cases. First, if $|\mathcal{P}^*| \geq |\mathcal{N}| \cdot \frac{\alpha_{\mathrm{WL}}}{2^{13}\Delta_0^2}$, then we return $\mathcal{N}' = \mathcal{N}$ and $\mathcal{P}^*$ as the set of paths routing a subset of the demand pairs in $\mathcal{N}'$. Therefore, we assume from now on that $|\mathcal{P}^*| < |\mathcal{N}| \cdot \frac{\alpha_{\mathrm{WL}}}{2^{13}\Delta_0^2}$. Let $\mathcal{N}'' = \mathcal{M}^0 \cup \mathcal{N}_0 \cup \mathcal{N}_1$, and let $\mathcal{N}' = \mathcal{N} \setminus \mathcal{N}'' = \mathcal{N}_2$. Then:

$$\begin{aligned}
|\mathcal{N}''| &\leq 5\tau^* + (\tau^*+1)|\mathcal{N}_1| \\
&\leq 5\tau^* + (\tau^*+1)(h+1)|\mathcal{P}^*| \\
&\leq 16\Delta_0\tau^* \cdot |\mathcal{P}^*| \\
&\leq \frac{1024\Delta_0^2}{\alpha_{\mathrm{WL}}} \cdot \frac{|\mathcal{N}|\alpha_{\mathrm{WL}}}{2^{13}\Delta_0^2} \\
&\leq \frac{|\mathcal{N}|}{6}.
\end{aligned}$$

Therefore, $|\mathcal{N}'| \geq 5|\mathcal{N}|/6$. From now on, our goal is to find a set $\mathcal{P}$ of paths, routing $\Omega\left(\frac{\mathsf{OPT}(G,\mathcal{N}')}{\Delta_0^8 \log^3 n}\right)$ demand pairs in $\mathcal{N}'$. From the above discussion, the demand pairs in $\mathcal{N}'$ have the following property.

P1. For every component $R \in \mathcal{R}$ with $R \cap \mathcal{T}(\mathcal{N}') \neq \emptyset$, if P is any path routing any demand pair in $\mathcal{N}'$, then $P \not\subseteq \eta^\circ(R)$.

Step 2: Mapping the Terminals. This step is almost identical to the similar step in Case 2a, except that now we crucially exploit Property (P1). Let $\mathcal{T}' = \mathcal{T}(\mathcal{N}')$, and let $U = \bigcup_{h'=1}^{h} V(Z_{h'}(x))$ be the set of all vertices lying on the cycles of the shell. Our next step is to define a mapping $\beta : \mathcal{T}' \to 2^U$ of the terminals $t \in \mathcal{T}'$ to subsets $\beta(t)$ of at most three vertices of U. For every terminal $t \in \mathcal{T}'$, we also define a corresponding G-normal curve $\Gamma(t)$, as before.

The mapping β and the curves $\Gamma(t)$ are defined as follows. First, for every terminal $t \in \mathcal{T}' \cap U$, we set $\beta(t) = \{t\}$, and we let $\Gamma(t)$ contain the vertex t only. For all remaining terminals $t \in \mathcal{T}'$, t must lie in some component $R \in \mathcal{R}$. If $|L(R)| \leq 2$, then we let $\beta(t) = L(R) \cup \{u(R)\}$ if $u(R)$ is defined, and $\beta(t) = L(R)$ otherwise. If $|L(R)| > 2$, then we let $\beta(t) = \{v\}$, where v is a leg of R, which is not an endpoint of $\sigma(R)$ (in other words, v is not the first and not the last leg of R). We then let $\Gamma(t)$ be the boundary $\gamma(R)$ of the disc $\eta(R)$ given by Theorem 7.13. Recall that the length of $\Gamma(t)$ is bounded by $2h + \Delta/2 + 3 < 16\Delta_0$, as $h \leq 8\Delta_0 - \Delta - 4$.

Let $\tilde{\mathcal{M}} = \bigcup_{(s,t)\in\mathcal{N}'} \beta(s) \times \beta(t)$. In the following two theorems, we relate the values of the solutions to problems $(G, \mathcal{N}')$ and $(G, \tilde{\mathcal{M}})$.

Theorem 7.18 *There is an efficient algorithm, that, given any solution to instance $(G, \tilde{\mathcal{M}})$, that routes κ demand pairs, finds a solution to instance $(G, \mathcal{N}')$, routing at least $\Omega\left(\frac{\kappa}{\Delta_0}\right)$ demand pairs.*

Proof: Let $\mathcal{P}_0$ be any collection of disjoint paths in graph G, routing a subset $\tilde{\mathcal{M}}_0$ of κ demand pairs in $\tilde{\mathcal{M}}$. We assume that $\kappa \geq 64\Delta_0$, as otherwise we can return a routing of a single demand pair in $\mathcal{N}'$. For every demand pair $(\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}_0$, let (s, t) be any corresponding demand pair in $\mathcal{N}'$, that is, $\tilde{s} \in \beta(s)$ and $\tilde{t} \in \beta(t)$.

We build a conflict graph H, whose vertex set is $\left\{v(\tilde{s}, \tilde{t}) \mid (\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}_0\right\}$, and there is a directed edge from $v(\tilde{s}_1, \tilde{t}_1)$ to $v(\tilde{s}_2, \tilde{t}_2)$ iff the unique path $P(\tilde{s}_1, \tilde{t}_1) \in \mathcal{P}_0$ routing the pair $(\tilde{s}_1, \tilde{t}_1)$ intersects $\Gamma(s_2)$ or $\Gamma(t_2)$ (in which case we say that there is a conflict between $(\tilde{s}_1, \tilde{t}_1)$ and $(\tilde{s}_2, \tilde{t}_2)$). Since all paths in $\mathcal{P}_0$ are node-disjoint, and all curves $\Gamma(s), \Gamma(t)$ have lengths at most $16\Delta_0$ each, the in-degree of every vertex in H is at most $32\Delta_0$. Therefore, we can efficiently compute an independent set I of size at least $\frac{\kappa}{32\Delta_0+1} \geq \frac{\kappa}{64\Delta_0}$ in H.

Let $\tilde{\mathcal{M}}_1 = \left\{(\tilde{s}, \tilde{t}) \mid v(\tilde{s}, \tilde{t}) \in I\right\}$, and let $\mathcal{P}_1 \subseteq \mathcal{P}_0$ be the set of paths routing the demand pairs in $\tilde{\mathcal{M}}_1$. Let $\mathcal{M}' = \left\{(s, t) \mid (\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}_1\right\}$. It is now enough to show that all demand pairs in $\mathcal{M}'$ can be routed in G. Consider any demand pair $(\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}_1$, and let $P \in \mathcal{P}_1$ be the path routing $(\tilde{s}, \tilde{t})$. We will extend the path P, so it connects s to t, by appending two paths: $Q(\tilde{s})$ connecting $\tilde{s}$ to s, and $Q(\tilde{t})$ connecting $\tilde{t}$ to t, to it. If $s \in U$, then $s = \tilde{s}$, and we define $Q(\tilde{s}) = \emptyset$. Assume now that $s \notin U$, and let $R \in \mathcal{R}$ be the component in which s lies. Let $Q(\tilde{s})$ be any path that starts from $\tilde{s}$, terminates at s, and except for its first edge, is contained in R. We define the path $Q(\tilde{t})$ similarly. Let $\mathcal{P}^*$ be the set of paths obtained by concatenating the paths in $\mathcal{P}_1$ with the paths in $\left\{Q(\tilde{s}), Q(\tilde{t}) \mid (\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}_1\right\}$. Then the paths in $\mathcal{P}^*$ route all demand pairs in $\mathcal{M}'$, so $|\mathcal{P}^*| \geq \frac{\kappa}{64\Delta_0}$. It is now enough to prove that the paths in $\mathcal{P}^*$ are node-disjoint. We do so in the following claim.

Claim 7.19 *The paths in $\mathcal{P}^*$ are node-disjoint.*

Proof: Observe that all endpoints of all paths of $\mathcal{P}_1$ are distinct. We first prove that for all terminals $\tilde{t}, \tilde{t}' \in \mathcal{T}(\tilde{\mathcal{M}}_1)$, with $\tilde{t} \neq \tilde{t}'$, paths $Q(\tilde{t})$ and $Q(\tilde{t}')$ are node-disjoint. Assume otherwise. Then $Q(\tilde{t}), Q(\tilde{t}') \neq \emptyset$, and $t, t' \notin U$. Let $t, t' \in \mathcal{T}(\tilde{\mathcal{M}}')$ be the terminals corresponding to $\tilde{t}$ and $\tilde{t}'$, respectively, so $Q(\tilde{t})$ connects $\tilde{t}$ to t, and $Q(\tilde{t}')$ connects $\tilde{t}'$ to t'. Let $R, R' \in \mathcal{R}$ be the components to which t and t' belong, respectively. Recall that except for its first edge, $Q(\tilde{t})$ is contained in R, and the same holds for $Q(\tilde{t}')$ and R'. Since $\tilde{t} \neq \tilde{t}'$, but $Q(\tilde{t}) \cap Q(\tilde{t}') \neq \emptyset$, we get that $R = R'$. But then $\tilde{t}' \in L(R) \cup \{u(R)\}$, and so it lies in $\eta(R)$. Let P' be the path of $\mathcal{P}_1$, such that $\tilde{t}'$ is an endpoint of P', and let $\tilde{s}'$ be its other endpoint. Since $\Gamma(t) = \gamma(R)$, and set $\mathcal{P}_1$ is conflict-free, path P' cannot intersect $\gamma(R)$. Therefore, $\tilde{s}'$ must belong to $\eta^\circ(R)$. Let s' be the terminal in $\mathcal{T}(\mathcal{M}')$ corresponding to $\tilde{s}'$, so path $Q(\tilde{s}')$ connects $\tilde{s}'$ to s'. We claim that if $Q(\tilde{s}') \neq \emptyset$, then $Q(\tilde{s}')$ is also contained in $\eta^\circ(R)$. Indeed, if $Q(\tilde{s}') \neq \emptyset$, then s' belongs to some component $R'' \in \mathcal{R}$, and $\tilde{s}' \in L(R'') \cup \{u(R'')\}$. But since $\tilde{s}' \in \eta^\circ(R)$, and since $\gamma(R)$ is a G-normal curve disjoint from $V(R'')$, $R'' \subseteq \eta^\circ(R)$ must hold. We conclude that $Q(\tilde{s}')$ is contained in $\eta^\circ(R)$. By concatenating $Q(\tilde{s}'), P'$ and $Q(\tilde{t}')$, we obtain a path P, connecting s' to t', where P is contained in $\eta^\circ(R)$, violating Property (P1). We conclude that paths $Q(\tilde{t})$ and $Q(\tilde{t}')$ are node-disjoint.

It is now enough to show that for every terminal $\tilde{t} \in \mathcal{T}(\tilde{\mathcal{M}}_1)$, and for every path $P \in \mathcal{P}_1$, such that $\tilde{t}$ is not an endpoint of P, $Q(\tilde{t})$ is disjoint from P. Assume otherwise, and let $\tilde{s}', \tilde{t}'$ be the endpoints of the path P. Since we have assumed that $Q(\tilde{t}) \cap P \neq \emptyset$, $Q(\tilde{t}) \neq \emptyset$, and so the vertex $t \in \mathcal{T}(\mathcal{M}')$ serving as the other endpoint of $Q(\tilde{t})$ must lie in some component $R \in \mathcal{R}$. Notice that since the paths in $\mathcal{P}_1$ have no conflicts, P is disjoint from $\gamma(R)$, and so it must be contained in $\eta^\circ(R)$. Using the same reasoning as above, we conclude that if $Q(\tilde{s}') \neq \emptyset$, then it is contained in $\eta^\circ(R)$, and the same holds for $Q(\tilde{t}')$. Therefore, there is a path P, obtained by concatenating $P', Q(\tilde{s}')$ and $Q(\tilde{t}')$, connecting the pair $(s', t') \in \mathcal{N}'$, with $P' \subseteq \eta^\circ(R)$, contradicting Property (P1). Since the paths in $\mathcal{P}_1$ are node-disjoint, it is now immediate to see that the paths in $\mathcal{P}^*$ must also be node-disjoint. □

□

Theorem 7.20 $\mathsf{OPT}(G, \tilde{\mathcal{M}}) \geq \frac{\mathsf{OPT}(G,\mathcal{N}')}{64\Delta_0}$.

Proof: Let $\mathcal{P}_0$ be the set of paths in the optimal solution to instance $(G, \mathcal{N}')$, and let $\mathcal{M}_0$ be the set of the demand pairs they route. We can assume that $|\mathcal{M}_0| \geq 64\Delta_0$, as otherwise we can route a single demand pair in $\tilde{\mathcal{M}}$.

As before, we define a conflict graph H, whose vertex set is $\{v(s,t) \mid (s,t) \in \mathcal{M}_0\}$, and there is a directed edge from $v(s_1,t_1)$ to $v(s_2,t_2)$ iff the unique path $P(s_1,t_1) \in \mathcal{P}_0$ routing the pair (s_1,t_1) intersects $\Gamma(s_2)$ or $\Gamma(t_2)$ (in which case we say that there is a conflict between (s_1,t_1) and (s_2,t_2)). Since all paths in $\mathcal{P}_0$ are node-disjoint, and all curves $\Gamma(s), \Gamma(t)$ have lengths at most $16\Delta_0$ each, the in-degree of every vertex in H is at most $32\Delta_0$. Therefore, we can efficiently compute an independent set I of size at least $\frac{\mathsf{OPT}(G,\mathcal{N}')}{32\Delta_0+1} \geq \frac{\mathsf{OPT}(G,\mathcal{N}')}{64\Delta_0}$ in H.

Let $\mathcal{M}_1 = \{(s,t) \mid v(s,t) \in I\}$, and let $\mathcal{P}_1 \subseteq \mathcal{P}_0$ be the set of paths routing the demand pairs in $\mathcal{M}_1$. We show that we can route $|\mathcal{M}_1|$ demand pairs of $\tilde{\mathcal{M}}$ in G via node-disjoint paths. Let $\mathcal{T}_1$ be the sets of all terminals participating in the pairs in $\mathcal{M}_1$.

Consider any terminal $t \in \mathcal{T}_1$. We say that t is a *good terminal* if $t \in U$, or t belongs to some component $R \in \mathcal{R}$, such that $|L(R)| \leq 2$. Otherwise, t is a *bad terminal*. Notice that if t is a good terminal, then the path $P \in \mathcal{P}_1$ that contains t as its endpoint must contain a vertex $t' \in \beta(t)$: if $t \in U$, then $\beta(t) = \{t\}$; otherwise, if $t \in R$ for some component $R \in \mathcal{R}$ with $|L(R)| \leq 2$, then $\beta(t) = L(R) \cup \{u(R)\}$ if $u(R)$ is defined, and $\beta(t) = L(R)$ otherwise. In either case, in order to enter R, path P has to visit a vertex of $\beta(t)$ (it is impossible that $P \subseteq R$ due to Property (P1)). Therefore, if t is a good terminal, then some vertex $t' \in P$ belongs to $\beta(t)$.

We transform the paths in $\mathcal{P}_1$ in two steps, to ensure that they connect demand pairs in $\tilde{\mathcal{M}}$. In the first step, for every path $P \in \mathcal{P}_1$ originating at a good terminal $s \in \mathcal{T}_1$, we truncate P at the first vertex $s' \in \beta(s)$, so it now originates at s'. Similarly, if P terminates at a good terminal $t \in T_1$, we truncate P at the last vertex $t' \in \beta(t)$, so it now terminates at t'. Let $\mathcal{P}'_1$ be the resulting set of paths. Notice that the paths in $\mathcal{P}'_1$ remain node-disjoint.

In order to complete our transformation, we need to take care of bad terminals. Let $t \in \mathcal{T}_1$ be any bad terminal. Then $t \in R$ for some component $R \in \mathcal{R}$ with $|L(R)| \geq 3$. Recall that in this case, $\beta(t)$ contains a unique vertex, that we denote by t', which is a leg of R, and it is not one of the endpoints of $\sigma(R)$. We then let $Q(t)$ be any path connecting t to t' in the sub-graph of G induced by $V(R) \cup \{t'\}$. By concatenating the paths in $\{Q(t)\}$ for all bad terminals $t \in \mathcal{T}_1$, and the paths in $\mathcal{P}'_1$, we obtain a collection $\tilde{\mathcal{P}}$ of at least $\frac{\mathsf{OPT}(G,\mathcal{N}')}{64\Delta_0}$ paths, routing demand pairs in $\tilde{\mathcal{M}}$. It now only remains to show that the paths in $\tilde{\mathcal{P}}$ are disjoint. Recall that the paths in $\mathcal{P}'_1$ were node-disjoint.

Claim 7.21 *The paths in $\tilde{\mathcal{P}}$ are node-disjoint.*

Proof: Consider first some pair t_1, t_2 of bad terminals. We show that the paths $Q(t_1)$ and $Q(t_2)$ are disjoint. Let P_1 and P_2 be the paths in $\mathcal{P}'_1$, for which t_1 and t_2 serve as endpoints, respectively. Let $P'_2 \in \mathcal{P}_1$ be the path corresponding to P_2, that is, P_2 is a sub-path of P'_2.

Recall that $t_1 \in R$ for some $R \in \mathcal{R}$, and recall that the disc whose boundary is $\Gamma(t_1)$ contains $R \cup L(R)$. Path P'_2 cannot cross $\gamma(R)$ since the paths in $\mathcal{P}_1$ are are conflict-free, and it is not contained in $\eta^\circ(R)$, since that would violate Property (P1). Therefore, path P'_2 lies completely outside $\eta(R)$, and so does path P_2. Let $R' \in \mathcal{R}$ be the component to which t_2 belongs. Then $R' \cap \eta(R) = \emptyset$ must also hold, since $\gamma(R)$ cannot intersect R', from Theorem 5.5. From our definition of $\beta(t)$ for bad terminals t, $\beta(t_1) \neq \beta(t_2)$, and each such set contains exactly one vertex. It is now easy to see that $Q(t_1)$ and $Q(t_2)$ are disjoint.

Consider now some bad terminal $t \in \mathcal{T}_1$, and let $P \in \mathcal{P}'_1$ be any path, such that t is not an endpoint of P. We next show that $Q(t)$ is disjoint from P. Let t', t'' be the endpoints of path P, and let $R \in \mathcal{R}$ be the component containing t. Let $P' \in \mathcal{P}_1$ be the path corresponding to P, so $P \subseteq P'$. Path P' cannot intersect $\gamma(R)$ since the paths in $\mathcal{P}_1$ are conflict-free, and it is not contained in $\eta^\circ(R)$ due to Property (P1). Therefore, P', and hence P, lie completely outside $\eta(R)$. Since $Q(t) \subseteq \eta(R)$, we get that $Q(t) \cap P = \emptyset$. Since the paths in $\mathcal{P}'_1$ are node-disjoint, it is now immediate to verify that the paths in $\tilde{\mathcal{P}}$ are node-disjoint as well. □ □

For all $1 \leq h', h'' \leq h$, we let $\tilde{\mathcal{M}}_{h',h''}$ be the set of all demand pairs $(\tilde{s}, \tilde{t}) \in \tilde{\mathcal{M}}$ with $\tilde{s} \in Z_{h'}(x)$ and $\tilde{t} \in Z_{h''}(x)$. Since $h \leq 16\Delta_0$, we obtain the following corollary.

Corollary 7.22 *There are some $1 \leq h', h'' \leq h$, such that $\mathsf{OPT}(G, \tilde{\mathcal{M}}_{h',h''}) \geq \Omega\left(\frac{\mathsf{OPT}(G,\mathcal{N}')}{\Delta_0^3}\right)$, and for any solution to instance $(G, \tilde{\mathcal{M}}_{h',h''})$, routing κ demand pairs, we can efficiently obtain a solution to instance $(G, \mathcal{N}')$, routing $\Omega(\kappa/\Delta_0)$ demand pairs.*

It is now enough to prove the following theorem.

Theorem 7.23 *There is an efficient algorithm, that for all $1 \leq h', h'' \leq h$ computes a set $\mathcal{P}_{h',h''}$ of disjoint paths, routing $\Omega\left(\frac{\mathsf{OPT}(G,\tilde{\mathcal{M}}_{h',h''})}{\Delta_0^4 \log^3 n}\right)$ demand pairs of $\tilde{\mathcal{M}}_{h',h''}$ in G.*

Step 3: Reduction to Routing on a Disc. In this step, we complete the proof of Theorem 7.1 by proving Theorem 7.23. We fix some $1 \leq h', h'' \leq h$. If $h' = h''$, then all terminals of $\mathcal{T}(\tilde{\mathcal{M}}_{h',h''})$ lie on $Z_{h'}(x)$, and we obtain an instance of the special case, with $Z = Z_{h'}(x)$, $C = C_x$ and $\hat{\mathcal{M}} = \tilde{\mathcal{M}}_{h',h''}$. We let $\hat{G}'$ be the graph obtained from G by deleting all vertices and edges lying in $D^\circ(Z_{h'})$, and apply the $O(\log n)$-approximation algorithm for $\mathsf{NDP\text{-}Disc}$ to the resulting instance $(\hat{G}', \hat{\mathcal{M}})$. From Theorem 7.7, $\mathsf{OPT}(\hat{G}', \hat{\mathcal{M}}) \geq \Omega\left(\frac{\mathsf{OPT}(G,\hat{\mathcal{M}})}{\Delta_0^2 \log n}\right)$, and so overall we obtain a routing of $\Omega\left(\frac{\mathsf{OPT}(G,\tilde{\mathcal{M}}_{h',h''})}{\Delta_0^2 \log^2 n}\right)$ demand pairs of $\tilde{\mathcal{M}}_{h',h''}$. Therefore, we assume that $h' \neq h''$ from now on. We assume without loss of generality that $h' < h''$, and that all source vertices of $\tilde{\mathcal{M}}_{h',h''}$ lie on $Z_{h'}(x)$ and all destination vertices of $\tilde{\mathcal{M}}_{h',h''}$ lie on $Z_{h''}(x)$.

In order to simplify the notation, we denote $\rho = h''-h'+1$, and we denote cycles $Z_{h'}(x), Z_{h'+1}(x), \ldots, Z_{h''}(x)$ by $\tilde{Z}_1, \tilde{Z}_2, \ldots, \tilde{Z}_\rho$, respectively. We also denote $\tilde{\mathcal{M}}_{h',h''}$ by $\tilde{\mathcal{M}}$, and the sets of all source and all destination vertices of the demand pairs in $\tilde{\mathcal{M}}$ by $\tilde{S}$ and $\tilde{T}$, respectively. We assume that $\mathsf{OPT}(G, \tilde{\mathcal{M}}) \geq 500\Delta_0^4$, since otherwise routing a single demand pair of $\tilde{\mathcal{M}}$ is sufficient. We also assume that for all $1 \leq i < \rho$, no edge connects a vertex of $\tilde{Z}_i$ to a vertex of $\tilde{Z}_{i+1}$, since we can subdivide each such edge with a vertex.

For convenience of notation, for each $1 \leq i < \rho$, we denote the set of vertices of G lying in $D^\circ(\tilde{Z}_{i+1}) \setminus D(\tilde{Z}_i)$ by $\tilde{U}_i$, and we denote by $\tilde{\mathcal{R}}_i$ the set of all **type-1** connected components of $G[\tilde{U}_i]$. Let $R \in \tilde{\mathcal{R}}_i$ be any such connected component. Recall that we have defined a segment $\sigma(R)$ of $\tilde{Z}_i$ containing all vertices of $L(R)$. We view $\sigma(R)$ as directed in the counter-clock-wise direction of $\tilde{Z}_i$, and we let $u'(R) \in L(R)$ be the first vertex of $\sigma(R)$. We then define $\chi(R)$ to be any path that connects $u'(R)$ to $u(R)$, such that all inner vertices of $\chi(R)$ belong to R.

The idea of the proof is to construct a collection $\mathcal{Q}$ of special paths, connecting the vertices of $\tilde{Z}_1$ to the vertices of $\tilde{Z}_\rho$, that we call *staircases*. We use the paths in $\mathcal{Q}$, in order to map all the source vertices in $\tilde{S}$ to some vertices of $\tilde{Z}_\rho$. We then reduce the problem to routing on a disc, by creating a hole $D^\circ(\tilde{Z}_\rho)$ in the sphere, so that all terminals participating in the new set $\hat{\mathcal{M}}$ of demand pairs lie on the boundary $\tilde{Z}_\rho$ of the hole.

In order to define the staircases, it will be convenient to work with a directed graph G', obtained from a sub-graph of G, as follows. First, we add the cycles $\tilde{Z}_1, \tilde{Z}_2, \ldots, \tilde{Z}_\rho$ to G', and for each such cycle $\tilde{Z}_i$, we direct all its edges in the counter-clock-wise direction along $\tilde{Z}_i$. Next, for each $1 \leq i < \rho$, for each component $R \in \tilde{\mathcal{R}}_i$, we add the path $\chi(R)$ to G', and direct all its edges so that the path is directed from $u'(R)$ to $u(R)$. We are now ready to define a staircase.

A staircase is simply a directed path in G', connecting some vertex of $\tilde{Z}_1$ to some vertex of $\tilde{Z}_\rho$, which is internally disjoint from $\tilde{Z}_\rho$. Observe that we can decompose any such staircase Q into $2\rho-2$ segments $\mu_1(Q), \chi_1(Q), \ldots, \mu_{\rho-1}(Q), \chi_{\rho-1}(Q)$, where for $1 \leq i < \rho$, $\mu_i(Q)$ is a directed sub-path of $\tilde{Z}_i$ (possibly consisting of a single vertex), and $\chi_i(Q) = \chi(R)$ for some $R \in \tilde{\mathcal{R}}_i$.

For every vertex $v \in V(\tilde{Z}_1)$, we build a special staircase $Q(v)$, as follows. Intuitively, we will try to minimize the lengths of the segments $\mu_i(Q(v))$. Denote $v = v_1$. For each $1 \leq i < \rho$, we now define the segments $\mu_i(Q(v))$ and $\chi_i(Q(v))$, and the vertex $v_{i+1} \in V(\tilde{Z}_{i+1})$, which is the last vertex of $\chi_i(Q(v))$, assuming that we are given the vertex $v_i \in \tilde{Z}_i$. We let $\mu_i(Q(v))$ be the shortest directed segment of $\tilde{Z}_i$, starting from v_i, that terminates at some vertex v'_i, such that for some component $R \in \tilde{\mathcal{R}}_i$, $u'(R) = v'_i$. If $R \in \tilde{\mathcal{R}}_i$ is a unique component with this property, then we set $\chi_i(Q(v)) = \chi(R)$. Otherwise, let $\mathcal{R}' \subseteq \tilde{\mathcal{R}}_i$ be the set of all components R with $u'(R) = v'_i$. Intuitively, we would like to set $\chi_i = \chi(R^*)$, where R^* is the first component of $\mathcal{R}'$ in the counter-clock-wise order (see Figure 9). In order to define R^* formally, we need the following observation.

Observation 7.24 *There is some component $R \in \tilde{\mathcal{R}}_i$, such that $R \notin \mathcal{R}'$.*

Proof: From our assumption, there is a set $\mathcal{P}$ of at least $500\Delta_0^4$ node-disjoint paths, connecting the vertices of $\tilde{S}$ to the vertices of $\tilde{T}$ in G. For every path $P \in \mathcal{P}$, let v_P be the first vertex of P lying on $\tilde{Z}_{i+1}$. Since we have assumed that no edge connects a vertex of $\tilde{Z}_i$ to a vertex of $\tilde{Z}_{i+1}$, there must be some component $R' \in \tilde{\mathcal{R}}_i$, such that $v_P = u(R')$, and moreover, P must contain some vertex of $L(R')$. We say that R' is responsible for P. Notice that each component of $\mathcal{R}_i$ may be responsible for at most one path in $\mathcal{P}$.

Notice that for all components $R' \in \mathcal{R}'$, except for maybe one, $L(R) = \{v_i'\}$, and so the components of $\mathcal{R}'$ may be responsible for at most two paths in $\mathcal{P}$. But $|\mathcal{P}| \geq 500\Delta_0^4$, so $\tilde{\mathcal{R}}_i \setminus \mathcal{R}' \neq \emptyset$. □

We draw a closed G-normal curve $\gamma \subseteq D^\circ(\tilde{Z}_{i+1}) \setminus D(\tilde{Z}_i)$, so that for each $R' \in \tilde{\mathcal{R}}_i$, $\gamma \cap V(R')$ is a contiguous set of vertices on γ, and all vertices that lie on γ belong to $\bigcup_{R' \in \tilde{\mathcal{R}}_i} V(R')$. We then let σ^* be the shortest segment of γ, containing all vertices of $\bigcup_{R' \in \mathcal{R}'} V(R')$, and no other vertices (in particular, the vertices of all components in $\mathcal{R}_i \setminus \mathcal{R}'$ do not lie on σ^*). Let v^* be the first vertex on σ^* in the counter-clock-wise direction, and let $R^* \in \mathcal{R}'$ be the unique component containing v^*. Finally, we set $\chi_i(Q(v)) = \chi(R^*)$. The final staircase $Q(v)$ is a concatenation of the paths in $\{\mu_i(Q(v)), \chi_i(Q(v))\}_{i=1}^{\rho-1}$. Each such staircase $Q(v)$ defines an undirected path in graph G, that we also denote by $Q(v)$, and we do not distinguish between them.

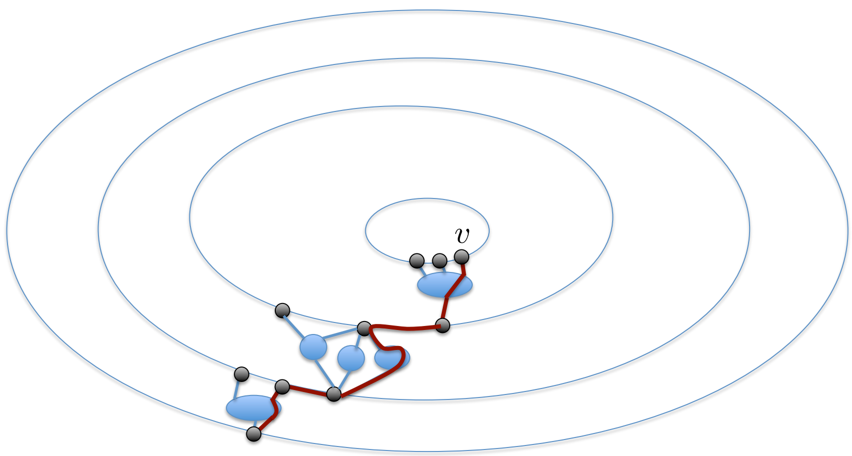


Figure 9: A staircase $Q(v)$ in graph G.

We will use the following lemma in order to bound the number of paths in the optimal solution intersecting any staircase $\mathcal{Q}(v)$.

Lemma 7.25 *Let $\mathcal{P}^*$ be any set of node-disjoint paths, connecting a subset of the demand pairs in $\tilde{\mathcal{M}}$. Then for each $v \in V(\tilde{Z}_1)$, the number of paths in $\mathcal{P}^*$ that intersect $Q(v)$ is at most $O(\Delta_0^2)$.*

Proof: For simplicity of notation, denote $Q = Q(v)$. We construct a collection $\mathcal{D}$ of at most $3\rho \leq O(\Delta_0)$ discs, such that for each disc $\eta \in \mathcal{D}$, its boundary is a G-normal curve of length at most $O(\Delta_0)$. We ensure that every vertex $v' \in V(Q)$ is contained in some disc $\eta \in \mathcal{D}$, and no terminal of $\tilde{T}$ is contained in $\bigcup_{\eta \in \mathcal{D}} \eta^\circ$. It is then easy to see that every path $P \in \mathcal{P}^*$ that intersects Q must also contain at least one vertex on the boundary of some disc in $\mathcal{D}$, and, since the total length of all boundaries of the discs in $\mathcal{D}$ is bounded by $O(\Delta_0^2)$, at most $O(\Delta_0^2)$ paths in $\mathcal{P}^*$ intersect Q.

For simplicity of notation, for all $1 \leq i < \rho$, we denote $\mu_i(Q)$ by μ_i, and $\chi_i(Q)$ by χ_i. We let v_i, v_i' be the first and the last vertex of μ_i, respectively, and we let $R_i \in \tilde{\mathcal{R}}_i$ be the component with $\chi(R_i) = \chi_i(Q)$, so that $v_{i+1} = u(R_i)$, and $v_i' = u'(R_i)$ (see Figure 10). We start with $\mathcal{D} = \emptyset$. For each $1 \leq i < \rho$, we add at most three discs to $\mathcal{D}$.

Fix some $1 \leq i < \rho$. We first add to $\mathcal{D}$ the disc $\eta(R_i)$, given by Theorem 5.5. Notice that this disc contains all vertices of $\chi_i(Q)$, and its boundary is a G-normal curve of length $O(\Delta_0)$. Let $\mathcal{R}'$ be the set of all type-1 and type-2 components R of $G[\tilde{U}_i]$, such that v_i is an inner vertex of $\sigma(R)$. Since the segments in $\{\sigma_R \mid R \in \mathcal{R}'\}$ form a nested set, we can assume that $\mathcal{R}' = \left\{R^1, \ldots, R^q\right\}$ with $\sigma(R^1) \subseteq \sigma(R^2) \subseteq \cdots \subseteq \sigma(R^q)$. Notice that among all components in $\mathcal{R}'$, only R^q may be a type-1 component, and the remaining components are type-2 components. If $\mathcal{R}' \neq \emptyset$, then we let v_i^* be the last vertex on $\sigma(R^q)$ in the counter-clock-wise direction, so $v_i^* \in \mu_i$, and we add the disc $\eta(R^q)$ to $\mathcal{D}$. Otherwise, we let $v_i^* = v_i$. Let $\mu_i' \subseteq \mu_i$ be the segment of μ_i between v_i^* and v_i' (see Figure 10). Notice that all vertices of $\mu_i \setminus \mu_i'$ are contained in the discs we have already added to $\mathcal{D}$. Our final step is to add a disc η_i^*, containing all vertices of μ_i', that is defined as follows. From our construction of the staircases and the set $\mathcal{R}'$, if R is a component of $G[\tilde{U}_i]$, and some inner vertex of μ_i' belongs to $L(R)$, then R is a type-2 component, and $\sigma(R) \subseteq \mu'(R)$. Therefore, we can draw a G-normal curve γ_i with endpoints v_i' and v_i^*, so that γ_i is contained in $D^\circ(\tilde{Z}_{i+1}) \setminus D(\tilde{Z}_i)$, and it is internally disjoint from all vertices of G. From Property (J4) of shells, we can construct curves γ, γ' that connect v_i' and v_i^*, respectively, to some vertices $a, a' \in C_x$, so that $\gamma, \gamma' \subseteq D(\tilde{Z}_i)$, and the length of each curve is bounded by $i \leq 8\Delta_0$. Combining $\gamma_i, \gamma, \gamma'$, and one of the segments of C_x with endpoints a and a', we obtain a closed curve γ_i^* of length $O(\Delta_0)$, such that the disc η_i^*, whose boundary is γ_i^*, contains all vertices of μ_i'. We then add η_i^* to $\mathcal{D}$.

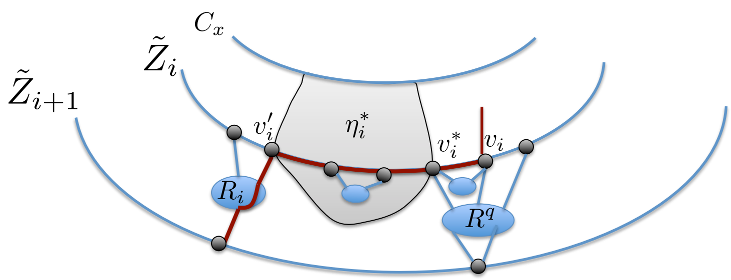


Figure 10: Constructing the disc η_i^*. The staircase is shown in red.

It is easy to verify that all vertices of $\mu_i \cup \chi_i$ lie in the discs $\eta(R_i), \eta(R^q), \eta_i^*$, and for each such disc, its boundary is a G-normal curve of length at most $O(\Delta_0)$. Moreover, from our construction, none of these discs contains a vertex of Z_ρ, except as part of its boundary. We conclude that every vertex of Q belongs to some disc $\eta \in \mathcal{D}$, the boundary of each such disc is a G-normal curve of length at most $O(\Delta_0)$. Every path $P \in \mathcal{P}^*$ intersecting Q must intersect the boundary of at least one disc in $\mathcal{D}$, and since $|\mathcal{D}| = O(\Delta_0)$, the number of such paths is bounded by $O(\Delta_0^2)$. □

Consider now the set $\mathcal{Q} = \left\{Q(v) \mid v \in V(\tilde{Z}_1)\right\}$ of staircases. Notice that once a pair of staircases meet, they always continue together. Therefore, we can partition the set $\mathcal{Q}$ of staircases into equivalence classes $\mathcal{Q}_1, \ldots, \mathcal{Q}_\lambda$, where two staircases belong to the same class iff they terminate at the same vertex. From the above discussion, staircases belonging to distinct equivalence classes must be disjoint. For each such set $\mathcal{Q}_j$, let $V_j \subseteq V(\tilde{Z}_1)$ be the set of all vertices where the staircases of $\mathcal{Q}_j$ originate. We also need the following lemma.

Lemma 7.26 *Let $\mathcal{P}^*$ be any set of node-disjoint paths, connecting a subset of the demand pairs in $\tilde{\mathcal{M}}$. Then for each $1 \leq j \leq \lambda$, the number of paths in $\mathcal{P}^*$ that originate at the vertices of V_j is at most $O(\Delta_0^2)$.*

Proof: Fix some $1 \leq j \leq \lambda$, and let $\mathcal{P}' \subseteq \mathcal{P}^*$ be the set of paths originating at the vertices of V_j. We assume that $\mathcal{Q}_j = \{Q_1, \ldots, Q_r\}$, and the endpoints of the paths Q_i that belong to V_j appear consecutively in this order on $\tilde{Z}_1$. Let $v_1, v_r \in V_j$ be the vertices where the paths Q_1 and Q_r originate. From Property (J4) of shells, we can construct G-normal curves γ, γ', connecting v_1 and v_r, respectively, to some vertices a, a' of C_x, so that the length of each such curve is bounded by $h' \leq 8\Delta_0$, and both curves are contained in $D(\tilde{Z}_1)$. We claim that every path in $\mathcal{P}'$ must contain a vertex of

$V(\gamma) \cup V(\gamma') \cup V(Q_1) \cup V(Q_r) \cup V(C_x)$. Indeed, consider the curve γ^* obtained by the union of the images of Q_1 and Q_r, the curves γ and γ', and one of the segments of C_x with endpoints a and a'. Then for every path of $\mathcal{P}'$, its source lies inside or on the curve γ^*, and its destination lies either on γ^* (when the destination is the common endpoint of all paths in $\mathcal{Q}_j$), or outside the curve γ^*. Therefore, every path in $\mathcal{P}'$ must contain a vertex of $V(\gamma) \cup V(\gamma') \cup V(Q_1) \cup V(Q_r) \cup V(C_x)$. Since γ, γ' and C_x contain $O(\Delta_0)$ vertices each, and since from Lemma 7.25 at most $O(\Delta_0^2)$ paths in $\mathcal{P}^*$ may intersect Q_1 and Q_r, we get that $|\mathcal{P}'| \le O(\Delta_0)^2$. □

We are now ready to complete our reduction. For every source vertex $s \in \tilde{S}$, let s' be the vertex of $\tilde{Z}_\rho$ that serves as the other endpoint of path $Q(s)$. We define a new set of demand pairs $\hat{\mathcal{M}} = \left\{(s', t) \mid (s,t) \in \tilde{\mathcal{M}}\right\}$. Notice that $(G, \hat{\mathcal{M}})$ is now an instance of the special case, where we use $Z = \tilde{Z}_\rho$ and $C = C_x$. We construct a graph $\hat{G}$ from G, by creating a hole $D^\circ(\tilde{Z}_\rho)$ in the sphere, and removing from G all edges and vertices that lie in the interior of $D(\tilde{Z}_\rho)$. We then apply the $O(\log n)$-approximation algorithm for NDP-Disc to the resulting problem, to obtain a routing of at least $\Omega\left(\frac{\mathsf{OPT}(\hat{G},\hat{\mathcal{M}})}{\log n}\right)$ demand pairs. From Theorem 7.7, $\mathsf{OPT}(\hat{G}, \hat{\mathcal{M}}) \ge \Omega\left(\frac{\mathsf{OPT}(G,\hat{\mathcal{M}})}{\Delta_0^2 \log n}\right)$, and so we obtain a set $\mathcal{P}$ of disjoint paths, routing of $\Omega\left(\frac{\mathsf{OPT}(G,\hat{\mathcal{M}})}{\Delta_0^2 \log^2 n}\right)$ demand pairs of $\hat{\mathcal{M}}$ in graph $\hat{G}$. Notice that for all $1 \le j \le \lambda$, if we denote by b_j the unique vertex where all staircases of $\mathcal{Q}_j$ terminate, then at most one demand pair in which b_j participates is routed by $\mathcal{P}$. We construct a set $\mathcal{Q}' \subseteq \mathcal{Q}$ of staircases as follows. For every demand pair $(s', t) \in \hat{\mathcal{M}}$ routed by $\mathcal{P}$, we select one source vertex $s \in \tilde{S}$ with $(s,t) \in \tilde{\mathcal{M}}$, such that the staircase $Q(s)$ terminates at s', and we add $Q(s)$ to $\mathcal{Q}'$. Notice that all staircases in $\mathcal{Q}'$ are disjoint from each other, since all of them belong to different sets $\mathcal{Q}_j$.

Since the paths in $\mathcal{P}$ do not use any vertices in the interior of $D(\tilde{Z}_\rho)$, we can combine them with the paths in $\mathcal{Q}'$, to obtain a routing of $\Omega\left(\frac{\mathsf{OPT}(G,\hat{\mathcal{M}})}{\Delta_0^2 \log^2 n}\right)$ demand pairs of $\tilde{\mathcal{M}}$ in graph G. In order to complete the proof of Theorem 7.23, it is now enough to show that $\mathsf{OPT}(G, \hat{\mathcal{M}}) \ge \Omega\left(\frac{\mathsf{OPT}(G,\tilde{\mathcal{M}})}{\Delta_0^2}\right)$. We do so in the following claim.

Claim 7.27 $\mathsf{OPT}(G, \hat{\mathcal{M}}) \ge \Omega\left(\frac{\mathsf{OPT}(G,\tilde{\mathcal{M}})}{\Delta_0^2}\right)$.

Proof: Let $\mathcal{P}_0$ be a set of paths routing $\kappa_0 = \mathsf{OPT}(G, \tilde{\mathcal{M}})$ demand pairs of $\tilde{\mathcal{M}}$ in G. We show that we can route $\Omega(\kappa_0/\Delta_0^2)$ demand pairs of $\hat{\mathcal{M}}$ in G. Let $\tilde{\mathcal{M}}_0 \subseteq \tilde{\mathcal{M}}$ be the set of the demand pairs routed by set $\mathcal{P}_0$. For each demand pair $(s,t) \in \tilde{\mathcal{M}}_0$, let $P(s,t) \in \mathcal{P}_0$ be the path routing this demand pair. We build a conflict graph H, whose vertex set is $\left\{v(s,t) \mid (s,t) \in \tilde{\mathcal{M}}_0\right\}$, and there is a directed edge from $v(s_1,t_1)$ to $v(s_2,t_2)$ iff one of the following happens: either (i) s_1 and s_2 both belong to the same set V_j, for $1 \le j \le \lambda$; or (ii) path $P(s_2,t_2)$ intersects $Q(s_1)$. From Lemmas 7.25 and 7.26, the in-degree of every vertex in H is at most $O(\Delta_0^2)$. Therefore, there is an independent set I of $\Omega(\kappa_0/\Delta_0^2)$ vertices in H. We let $\tilde{\mathcal{M}}_1 = \{(s,t) \mid v(s,t) \in I\}$, and we let $\mathcal{P}_1 \subseteq \mathcal{P}_0$ be the set of paths routing the demand pairs in $\tilde{\mathcal{M}}_1$.

Let $\hat{\mathcal{M}}' \subseteq \hat{\mathcal{M}}$ contain, for every demand pair $(s,t) \in \tilde{\mathcal{M}}_1$, the pair (s', t). It is now enough to show that all demand pairs in $\hat{\mathcal{M}}'$ can be routed in G. Let $\mathcal{Q}'$ contain all staircases $Q(s)$, where s participates in some demand pair in $\tilde{\mathcal{M}}_1$. Then all staircases in $\mathcal{Q}'$ are disjoint from each other, since they all belong to different sets $\mathcal{Q}_j$, for $1 \le j \le \lambda$. Moreover, for each staircase $Q(s) \in \mathcal{Q}'$, all paths in $\mathcal{P}_1 \setminus \{P(s,t)\}$ are disjoint from $Q(s)$. By concatenating the paths in $\mathcal{P}_1$ and the staircases in $\mathcal{Q}'$, we obtain a collection of node-disjoint paths routing all demand pairs in $\hat{\mathcal{M}}'$ □

8 Proof of Theorem 1.3

We perform a transformation to instance $(G, \mathcal{M})$ as before, to ensure that every terminal participates in at most one demand pair, and the degree of every terminal is 1. The number of vertices in the new instance is bounded by $2n^2$, and abusing the notation we denote this number by n. We use the following analogue of Theorem 4.1.

Theorem 8.1 *There is an efficient algorithm, that, given any semi-feasible solution to (LP-flow2), either computes a routing of at least $\Omega\left(\frac{(X^*)^{1/19}}{\operatorname{poly}\log n}\right)$ demand pairs in $\mathcal{M}$ via node-disjoint paths, or returns a constraint of type (5), that is violated by the current solution.*

We show that the above theorem implies Theorem 1.3. The Ellipsoid Algorithm, in every iteration, applies the above theorem to the current semi-feasible solution (x, f) to (LP-flow2). If the outcome is a solution routing at least $\Omega\left(\frac{(X^*)^{1/19}}{\operatorname{poly}\log n}\right)$ demand pairs, then we obtain the desired routing, assuming that $X^* = \mathsf{OPT}$ is guessed correctly. Otherwise, we obtain a violated constraint of type (5), and continue to the next iteration of the Ellipsoid Algorithm. The algorithm is guaranteed to terminate with a feasible solution after a number of iterations that is polynomial in the number of the LP-variables, so we obtain an efficient algorithm, that returns a solution routing $\Omega\left(\frac{(\mathsf{OPT}(G,\mathcal{M}))^{1/19}}{\operatorname{poly}\log n}\right)$ demand pairs. We now focus on proving Theorem 8.1.

We again process the fractional solution (x, f) to obtain a new fractional solution (x', f'), where every demand pair sends either 0 or w^* flow units, in the same way as described in Section 4. We let $\mathcal{M}' \subseteq \mathcal{M}$ denote the set of the demand pairs (s_i, t_i) with non-zero flow value x'_i in this new solution. As before, the total flow between the demand pairs in $\mathcal{M}'$ is at least $\Omega(X^*/\log k)$ in the new solution, and, if we find a subset $\mathcal{M}'' \subseteq \mathcal{M}'$ of demand pairs with $\mathsf{OPT}(G, \mathcal{M}'') \leq w^*|\mathcal{M}''|/2$, then set $\mathcal{M}''$ defines a violated constraint of type (5) for (LP-flow2). Therefore, we focus on set $\mathcal{M}'$ and for simplicity denote $\mathcal{M} = \mathcal{M}'$.

We decompose the input instance $(G, \mathcal{M})$ into a collection of well-linked instances $\left\{(G_j, \mathcal{M}^j)\right\}_{j=1}^r$ using Theorem 4.2. For each $1 \leq j \leq r$, let $W_j = w^*|\mathcal{M}^j|$ be the contribution of the demand pairs in $\mathcal{M}^j$ to the current flow solution and let $W = \sum_{j=1}^r W_j = \Omega(X^*/\log k)$.

Theorem 4.3 guarantees that for each $1 \leq j \leq r$, we can obtain one of the following:

1. Either a collection $\mathcal{P}^j$ of node-disjoint paths, routing $\Omega(W_j^{1/19}/\operatorname{poly}\log n)$ demand pairs of $\mathcal{M}^j$ in G_j; or
2. A collection $\tilde{\mathcal{M}}^j \subseteq \mathcal{M}^j$ of demand pairs, with $|\tilde{\mathcal{M}}^j| \geq |\mathcal{M}^j|/2$, such that $\mathsf{OPT}(G_j, \tilde{\mathcal{M}}^j) \leq w^*|\tilde{\mathcal{M}}^j|/8$.

We say that instance $(G_j, \mathcal{M}^j)$ is a type-1 instance, if the first outcome happens for it, and we say that it is a type-2 instance otherwise. Let $I_1 = \left\{j \mid (G_j, \mathcal{M}^j) \text{ is a type-1 instance}\right\}$, and similarly, $I_2 = \left\{j \mid (G_j, \mathcal{M}^j) \text{ is a type-2 instance}\right\}$. We consider two cases, where the first case happens when $\sum_{j \in I_1} W_j \geq W/2$, and the second case when $\sum_{j \in I_2} W_j \geq W/2$. In the second case, we let $\mathcal{M}' = \bigcup_{j \in I_2} \tilde{\mathcal{M}}^j$, and by the same reasoning as in Section 4, the following inequality, that is violated by the current LP-solution, is a valid constraint of (LP-flow2):

$$\sum_{(s_i,t_i) \in \mathcal{M}'} x'_i \leq w^*|\mathcal{M}'|/2.$$

We now focus on Case 1, where the number of paths routed for each instance $(G_j, \mathcal{M}^j)$ with $j \in I_1$ is at least $|\mathcal{P}^j| = \Omega\left(\frac{W_j^{1/19}}{\operatorname{poly}\log n}\right)$. Since $\sum_{j \in I_1} W_j \geq W/2 = \Omega\left(X^*/\log k\right)$, the total number of paths routed is:

$$\begin{aligned}
\sum_{j \in I_1} |\mathcal{P}^j| &\geq \sum_{j \in I_1} \Omega\left(\frac{W_j^{1/19}}{\operatorname{poly}\log n}\right) \\
&\geq \sum_{j \in I_1} \Omega\left(\frac{W_j}{W^{18/19} \cdot \operatorname{poly}\log n}\right) \\
&= \Omega\left(\frac{W^{1/19}}{\operatorname{poly}\log n}\right) \\
&= \Omega\left(\frac{(X^*)^{1/19}}{\operatorname{poly}\log n}\right).
\end{aligned}$$

9 Conclusion and Open Problems

In this paper we showed the first approximation algorithm for the NDP-Planar problem, whose approximation factor breaks the $\Omega(n^{1/2})$ barrier of the multicommodity flow LP-relaxation. We introduce a number of new techniques, that we hope will be helpful in obtaining better approximation algorithms for this problem. We note that our initial motivation came from the improved approximation algorithm for NDP-Grid of [CK15]. Even though adapting their main idea to the more general setting of planar graphs is technically challenging, we believe that the work of [CK15] on the much simpler and better structured grid graphs helped crystallize the main conceptual idea that eventually lead to this result. Therefore, we believe that studying the NDP-Grid problem can be very helpful in understanding the more general NDP-Planar problem. The best current approximation algorithm for NDP-Grid achieves an $\tilde{O}(n^{1/4})$-approximation, and it seems likely that this approximation ratio can be improved. We leave open the question of whether the techniques introduced in this paper can help improve the $O(n^{1/2})$-approximation factor of [CKS06] for EDP on planar graphs. Finally, we remark that the complexity of the NDP-Disc and the NDP-Cylinder problems is still not well-understood: we provide an $O(\log k)$-approximation algorithm for both problems, and we are not aware of any results that prove that the optimization versions of NDP-Disc or NDP-Cylinder are NP-hard. We note that the EDP problem is known to be NP-hard for both these settings [Nav12], but we are not aware of any approximation algorithms for it.

References

[ACG+10] Matthew Andrews, Julia Chuzhoy, Venkatesan Guruswami, Sanjeev Khanna, Kunal Talwar, and Lisa Zhang. Inapproximability of edge-disjoint paths and low congestion routing on undirected graphs. *Combinatorica*, 30(5):485–520, 2010.

[AKR03] Eyal Amir, Robert Krauthgamer, and Satish Rao. Constant factor approximation of vertex-cuts in planar graphs. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 90–99. ACM, 2003.

[And10] Matthew Andrews. Approximation algorithms for the edge-disjoint paths problem via Raecke decompositions. In *Proceedings of IEEE FOCS*, pages 277–286, 2010.

[AR95] Yonatan Aumann and Yuval Rabani. Improved bounds for all optical routing. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, SODA '95, pages 567–576, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.

[AZ05] Matthew Andrews and Lisa Zhang. Hardness of the undirected edge-disjoint paths problem. In *STOC*, pages 276–283. ACM, 2005.

[CC] Chandra Chekuri and Julia Chuzhoy. Half-integral all-or-nothing flow. Unpublished Manuscript.

[CE13] Chandra Chekuri and Alina Ene. Poly-logarithmic approximation for maximum node disjoint paths with constant congestion. In *Proc. of ACM-SIAM SODA*, 2013.

[Chu12] Julia Chuzhoy. Routing in undirected graphs with constant congestion. In *Proc. of ACM STOC*, pages 855–874, 2012.

[CK15] 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, August 24-26, 2015, Princeton, NJ, USA*, volume 40 of *LIPIcs*, pages 187–211. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[CKS04] Chandra Chekuri, Sanjeev Khanna, and F Bruce Shepherd. Edge-disjoint paths in planar graphs. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 71–80. IEEE, 2004.

[CKS05] Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In *Proc. of ACM STOC*, pages 183–192, 2005.

[CKS06] 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(1):137–146, 2006.

[CL12] Julia Chuzhoy and Shi Li. A polylogarithimic approximation algorithm for edge-disjoint paths with congestion 2. In *Proc. of IEEE FOCS*, 2012.

[DH05] Erik D. Demaine and Mohammad Taghi Hajiaghayi. Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 682–689. SIAM, 2005.

[EIS76] Shimon Even, Alon Itai, and Adi Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976.

[Kar72] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[KK13] Ken-Ichi Kawarabayashi and Yusuke Kobayashi. An O(log n)-approximation algorithm for the edge-disjoint paths problem in Eulerian planar graphs. *ACM Trans. Algorithms*, 9(2):16:1–16:13, March 2013.

[Kle05] Jon Kleinberg. An approximation algorithm for the disjoint paths problem in even-degree planar graphs. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 627–636, Washington, DC, USA, 2005. IEEE Computer Society.

[KS04] Stavros G. Kolliopoulos and Clifford Stein. Approximating disjoint-path problems using packing integer programs. *Mathematical Programming*, 99:63–87, 2004.

[KT95] Jon M. Kleinberg and Éva Tardos. Disjoint paths in densely embedded graphs. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 52–61, 1995.

[KT98] Jon M. Kleinberg and Éva Tardos. Approximations for the disjoint paths problem in high-diameter planar networks. *J. Comput. Syst. Sci.*, 57(1):61–73, 1998.

[KvL84] MR Kramer and Jan van Leeuwen. The complexity of wire-routing and finding minimum area layouts for arbitrary vlsi circuits. *Advances in computing research*, 2:129–146, 1984.

[Lyn75] James F. Lynch. The equivalence of theorem proving and the interconnection problem. *SIGDA Newsl.*, 5(3):31–36, September 1975.

[Nav12] Guyslain Naves. The hardness of routing two pairs on one face. *Mathematical programming*, 131(1-2):49–69, 2012.

[Räc02] Harald Räcke. Minimizing congestion in general networks. In *Proc. of IEEE FOCS*, pages 43–52, 2002.

[RLWW96] Heike Ripphausen-Lipa, Dorothea Wagner, and Karsten Weihe. Linear-time algorithms for disjoint two-face paths problems in planar graphs. *International Journal of Foundations of Computer Science*, 07(02):95–110, 1996.

[RS86] 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.

[RS90] N. Robertson and P. D. Seymour. Outline of a disjoint paths algorithm. In *Paths, Flows and VLSI-Layout*. Springer-Verlag, 1990.

[RS95] Neil Robertson and Paul D Seymour. Graph minors. XIII. the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.

[RST94] Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a planar graph. *J. Comb. Theory, Ser. B*, 62(2):323–348, 1994.

[RT87] Prabhakar Raghavan and Clark D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, December 1987.

[RZ10] Satish Rao and Shuheng Zhou. Edge disjoint paths in moderately connected graphs. *SIAM J. Comput.*, 39(5):1856–1887, 2010.

[SAN90] Hitoshi Suzuki, Takehiro Akama, and Takao Nishizeki. Finding steiner forests in planar graphs. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '90, pages 444–453, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics.

[SCS11] Loïc Seguin-Charbonneau and F. Bruce Shepherd. Maximum edge-disjoint paths in planar graphs with congestion 2. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 200–209, Washington, DC, USA, 2011. IEEE Computer Society.

A Proofs Omitted from Section 2

A.1 Proof of Observation 2.1

For every terminal $t \in \mathcal{T}$, let $v_t \in V(G)$ be the unique neighbor of t in G. Let (A, C, B) be the cut computed by algorithm $\mathcal{A}_{\text{AKR}}$, whose sparsity $\beta = \frac{|C|}{\min\{|A \cap \mathcal{T}|, |B \cap \mathcal{T}|\} + |C \cap \mathcal{T}|}$ is within factor α_{AKR} of the optimal one. If $\beta > 1$, then we replace cut (A, C, B) with the cut $(\emptyset, \mathcal{T}, V(G) \setminus \mathcal{T})$, obtaining a cut of sparsity 1, and continue with our algorithm, so we assume that $\beta \leq 1$ from now on. While $C \cap \mathcal{T} \neq \emptyset$, let $t \in C \cap \mathcal{T}$ be any terminal in C.

If $|A \cap \mathcal{T}| < |B \cap \mathcal{T}|$, then we move t from C to A, and if $v_t \notin C$, we add v_t to C. Let (A', C', B') be the resulting tri-partition of $V(G)$. It is easy to see that (A', C', B') is a valid vertex cut, and $|C'| \leq |C|$, while $\min\{|A' \cap \mathcal{T}|, |B' \cap \mathcal{T}|\} + |C' \cap \mathcal{T}| = \min\{|A \cap \mathcal{T}|, |B \cap \mathcal{T}|\} + |C \cap \mathcal{T}|$, so the sparsity of the cut does not increase.

The case where $|B \cap \mathcal{T}| < |A \cap \mathcal{T}|$ is dealt with similarly.

Finally, assume that $|A \cap \mathcal{T}| = |B \cap \mathcal{T}|$. If $v_t \in A$, then we move t from C to A, and otherwise we move t from C to B. Let (A', C', B') be the resulting tri-partition of $V(G)$. It is easy to see that (A', C', B') is a valid vertex cut, and $|C'| = |C| - 1$. It is also easy to verify that $\min\{|A' \cap \mathcal{T}|, |B' \cap \mathcal{T}|\} + |C' \cap \mathcal{T}| \geq \min\{|A \cap \mathcal{T}|, |B \cap \mathcal{T}|\} + |C \cap \mathcal{T}| - 1$, and $\min\{|A' \cap \mathcal{T}|, |B' \cap \mathcal{T}|\} + |C' \cap \mathcal{T}| > 0$ hold. Since $\beta \leq 1$, the sparsity of the cut does not increase. Once we process all terminals in set C in this fashion, we obtain a final vertex cut (A, C, B) whose sparsity is at most β, and $C \cap \mathcal{T} = \emptyset$.

A.2 Proof of Claim 2.2

We compute a partition $\mathcal{R}(\tau)$ for every tree $\tau \in F$ separately. The partition is computed in iterations, where in the jth iteration we compute the set $R_j(\tau) \subseteq V(\tau)$ of vertices, together with the corresponding collection $\mathcal{P}_j(\tau)$ of paths. For the first iteration, if τ contains a single vertex v, then we add this vertex to $\mathcal{R}_1(\tau)$ and terminate the algorithm. Otherwise, for every leaf v of τ, let $P(v)$ be the longest directed path of τ, starting at v, that only contains degree-1 and degree-2 vertices, and does not contain the root of τ. We then add the vertices of $P(v)$ to $R_1(\tau)$, and the path $P(v)$ to $\mathcal{P}_1(\tau)$. Once we process all leaf vertices of τ, the first iteration terminates. It is easy to see that all resulting vertices in $R_1(\tau)$ induce a collection $\mathcal{P}_1(\tau)$ of disjoint paths in τ, and moreover if $v, v' \in R_1(\tau)$, and there is a path from v to v' in τ, then v, v' lie on the same path in $\mathcal{P}_1(\tau)$. We then delete all vertices of $R_1(\tau)$ from τ.

The subsequent iterations are executed similarly, except that the tree τ becomes smaller, since we delete all vertices that have been added to the sets $R_j(\tau)$ from the tree.

It is now enough to show that this process terminates after $\lceil \log n \rceil$ iterations. In order to do so, we can describe each iteration slightly differently. Before each iteration starts, we contract every edge e of the current tree, such that at least one endpoint of e has degree 2 in the tree, and e is not incident on the root of τ. We then obtain a tree in which every inner vertex (except possibly the root) has degree at least 3, and delete all leaves from this tree. The number of vertices remaining in the contracted tree after each such iteration therefore decreases by at least factor 2. It is easy to see that the number of iteration in this procedure is the same as the number of iterations in our algorithm, and is bounded by $\lceil \log n \rceil$. For each $1 \le j \le \lceil \log n \rceil$, we then let $R_j = \bigcup_{\mathcal{T} \in F} R_j(\mathcal{T})$.

A.3 Proof of Lemma 2.3

We denote by $\mathcal{T}$ the set of all vertices participating in the demand pairs in $\mathcal{M}$, and we refer to them as terminals. Consider any demand pair $(s,t) \in \mathcal{M}$, and let $\sigma(s,t), \sigma'(s,t)$ be the two segments of C whose endpoints are s and t. We assume without loss of generality that $|\sigma(s,t) \cap \mathcal{T}| \le |\sigma'(s,t) \cap \mathcal{T}|$, and we denote $\delta(s,t) = |\sigma(s,t) \cap \mathcal{T}| - 1$. By possibly renaming the terminals s and t, we assume that s appears before t on $\sigma(s,t)$ as we traverse it in counter-clock-wise direction along C. Our first step is to partition the demand pairs in $\mathcal{M}$ into $\lceil \log \kappa \rceil$ subsets $\mathcal{N}_1, \ldots, \mathcal{N}_{\lceil \log \kappa \rceil}$, as follows. For each $1 \le i \le \lceil \log \kappa \rceil$, $\mathcal{N}_i$ contains all demand pairs (s,t) with $2^{i-1} \le \delta(s,t) < 2^i$. In order to complete the proof of the lemma, it is enough to show that for each $1 \le i \le \lceil \log \kappa \rceil$, we can partition $\mathcal{N}_i$ into four sets of demand pairs, each of which is r-split, for some integer r.

Fix some $1 \le i \le \lceil \log \kappa \rceil$, and assume that $\mathcal{T} = \{v_1, \ldots, v_{2\kappa}\}$, where the vertices are indexed in the circular order of their appearance on C. We let A contain all vertices v_j, where $j = 1$ modulo 2^{i-1}. For convenience, we denote $A = \{a_1, \ldots, a_z\}$, and we assume that the vertices of A appear in this circular order on C, with $a_1 = v_1$. If $|A| = 1$, then $|\mathcal{T}| \le 2^{i-1}$, and so $\delta(s,t) \le 2^{i-2}$ for all $(s,t) \in \mathcal{M}$, and $\mathcal{N}_i = \emptyset$. If $|A| = 2$, then let β, β' be the two segments of C between a_1 and a_2, where β contains a_1 but not a_2, and β' contains a_2 but not a_1. Then for every pair $(s,t) \in \mathcal{N}_i$, one of the two terminals lies on β and the other on β', and so $\mathcal{N}_i$ is 1-split. We assume from now on that $z \ge 3$.

For each $1 \le j \le z-1$, let β_j be the segment of C from a_j and a_{j+1}, as we traverse C in the counter-clock-wise order, so that β_j includes a_j but excludes a_{j+1}. We let β_z be the segment of C from a_z to a_1, as we traverse C in the counter-clock-wise order, so that β_z includes a_z, but not a_1. Then for every segment β_j with $1 \le j < z$, $|\beta_j \cap \mathcal{T}| = 2^{i-1}$, while $|\beta_z \cap \mathcal{T}| \le 2^{i-1}$. Notice that for every demand pair $(s,t) \in \mathcal{N}_i$, one of the following must happen: either (i) $s \in \beta_j$, $t \in \beta_{j+1}$ for some $1 \le j \le z$, where we treat $z+1$ as 1; or (ii) $s \in \beta_z$, $t \in \beta_2$; or (iii) $s \in \beta_{z-1}$, $t \in \beta_1$.

We are now ready to partition $\mathcal{N}_i$ into four subsets. The first subset, $\mathcal{N}_i^1$, contains all pairs $(s,t) \in \mathcal{N}_i$ with $s \in \beta_{z-1}$, $t \in \beta_1$. The second set, $\mathcal{N}_i^2$, contains all pairs $(s,t) \in \mathcal{N}_i$, with $s \in \beta_z$, $t \in \beta_1 \cup \beta_2$. It is immediate to verify that each of these two sets is 1-split. The third set, $\mathcal{N}_i^3$, contains all pairs $(s,t) \in \mathcal{N}_i \setminus (\mathcal{N}_i^1 \cup \mathcal{N}_i^2)$, where $s \in \beta_j$, for an odd index $1 \le j < z$. This set is $\lfloor z/2 \rfloor$-split, with the segments $\beta_1, \ldots, \beta_z$ giving the splitting. The last set, $\mathcal{N}_i^4$, contains all pairs $(s,t) \in \mathcal{N}_i \setminus (\mathcal{N}_i^1 \cup \mathcal{N}_i^2 \cup \mathcal{N}_i^3)$. This set is similarly $\lfloor z/2 \rfloor$-split, since for all $(s,t) \in \mathcal{N}_i^4$, $s \in \beta_j$ for an even index $1 \le j < z$. Overall, we obtain a partition of $\mathcal{M}$ into $4\lceil \log \kappa \rceil$ sets, each of which is r-split for some integer r.

A.4 Proof of Lemma 2.4

Consider any path $P \in \mathcal{P}$. A sub-path Q of P is called a bump, if the two endpoints of Q lie on C, and all the intermediate vertices of Q do not lie on C (notice that Q may be simply an edge of C). Since path P is simple, a bump Q cannot contain s.

We now define a shadow of a bump Q. If Q is an edge of C, then the shadow of Q is Q itself. Assume now that Q is not an edge of C. Let $u, v \in V(C)$ be the two endpoints of Q, and let σ, σ' be the two segments of C with endpoints u and v. Let C_1 be the union of σ and Q, and C_2 the union of σ' and Q. Then one of the two corresponding discs, $D(C_1)$ and $D(C_2)$ contains s - we assume that it is the latter disc. We then let σ be the *shadow* of Q on C (see Figure 11).

We note that all inner points on the image of Q must lie outside $D^{\circ}(C)$, as otherwise, we can find a cycle

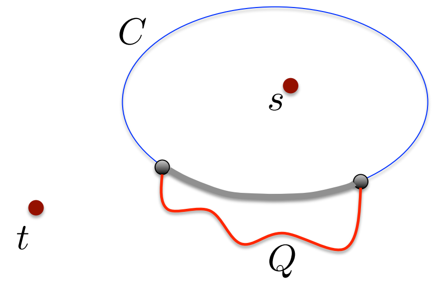


Figure 11: A bump Q and its shadow

C' in H with $D(C')$ containing s and $D(C') \subsetneq D(C)$, contradicting the fact that $C = \text{min-cycle}(H, s)$. We obtain the following two simple observations.

Observation A.1 *Let $P \in \mathcal{P}$ be any path, and let R be the longest segment of P, starting from s and terminating at a vertex of C, such that R does not contain any vertex of C as an inner vertex. Let $v \in V(C)$ be the endpoint of R, and let R' be the sub-path of P from v to t. Then every point p on the image of R' lies outside $D^\circ(C)$.*

Proof: If any such point p lies in $D^\circ(C)$, then for some bump Q of P, some inner point on the image of Q lies in $D^\circ(C)$, and this is impossible, as observed above. □

Observation A.2 *Let $P, P' \in \mathcal{P}$ be two distinct paths, let Q be a bump on P, and let Q' be a bump on P'. Then the shadows of Q and Q' are node-disjoint.*

Proof: Let σ be the shadow of Q, and let a, b be its endpoints. Let σ' be the shadow of Q', and let a', b' be its endpoints. Since the inner points of the images of both Q and Q' lie outside $D^\circ(C)$, and their endpoints are all distinct, if σ and σ' are not disjoint, then either $\sigma \subsetneq \sigma'$, or $\sigma' \subsetneq \sigma$. Assume without loss of generality that the latter is true. Let A be the disc whose boundary is $\sigma \cup Q$. Then $t \notin A$, but some vertex of P' lies in A. Let R' be the longest sub-path of P' starting from s and terminating at a vertex of C, such that R' is internally disjoint from C. Then there is a vertex $v' \in V(Q')$, that does not lie on R'. Let R'' be the sub-path of P' from v' to t. Then path R'' originates in disc A and terminates outside A. But it can only leave A by traveling inside $D^\circ(C)$, which is impossible from Observation A.1. □

We now re-route each path $P \in \mathcal{P}$, as follows. Let u and v be the first and the last vertex of P that belong to C, respectively. Let $\tilde{P}$ be the union of the shadows of all bumps of P. Then $\tilde{P}$ is a path, contained in C, that contains u and v. Let $\tilde{P}'$ be a simple sub-path of $\tilde{P}$ connecting u to v, and let P' be obtained by concatenating the segment of P from s to u, $\tilde{P}'$, and the segment of P from v to t. Notice that path P' is monotone with respect to C, and all paths in the resulting set $\mathcal{P}' = \{P' \mid P \in \mathcal{P}\}$ are internally node-disjoint by Observation A.2.

A.5 Proof of Theorem 2.5

We perform r iterations. At the beginning of the hth iteration, for $1 \leq h \leq r$, we assume that we are given a set $\mathcal{P}_{h-1}$ of κ node-disjoint paths in H', connecting the vertices of A to the vertices of B, so that the paths in $\mathcal{P}_{h-1}$ are internally disjoint from $V(C) \cup V(Y)$, and they are monotone with respect to $Z_1, \ldots, Z_{h-1}$. The output of iteration h is a set $\mathcal{P}_h$ of κ node-disjoint paths in H', connecting the vertices of A to the vertices of B, so that the paths in $\mathcal{P}_h$ are internally disjoint from $V(C) \cup V(Y)$, and they are monotone with respect to $Z_1, \ldots, Z_h$. The output of the algorithm is the set $\mathcal{P}_r$ of paths computed in the last iteration. For simplicity of notation, we denote $Z_0 = C$, even though C is not a cycle of H.

We start with $\mathcal{P}_0 = \mathcal{P}$. It is immediate to see that this is a valid input to iteration 1. Assume now that we are given a set $\mathcal{P}_{h-1}$ of paths, which is a valid input to iteration h. The iteration is then executed as follows. Let H'' be the graph obtained by starting with $H'' = (\bigcup_{h=0}^{r} V(Z_h)) \cup \mathcal{P}_{h-1}$, and then contracting all vertices lying in $D(Z_{h-1})$ into a source vertex s', and all vertices of B into a destination vertex t'. Since $B \subseteq V(Y)$, where Y is a connected sub-graph of G, $Y \cap D(Z_r) = \emptyset$, and the paths in $\mathcal{P}_{h-1}$ are internally disjoint from $V(Y)$, graph H'' is a planar graph. Moreover, $Z_h = \text{min-cycle}(H'', s')$ from the definition of tight concentric cycles. We consider the drawing of H'' in the plane where t' is incident on the outer face. It is easy to see that $Z_h = \text{min-cycle}(H'', s')$ still holds with respect to this new drawing of H''.

From Lemma 2.4, there is a set $\mathcal{Q}$ of κ internally node-disjoint paths in H'', connecting s' to t', that are monotone with respect to Z_h. In order to construct the set $\mathcal{P}_h$ of paths, let $P \in \mathcal{P}_{h-1}$ be any path, and let v_P be the last vertex of P lying on Z_{h-1}. Let P' be the sub-path of P starting from v_P and terminating at the endpoint of P lying in B. Notice that from the monotonicity of the paths in $\mathcal{P}_{h-1}$ with respect to $Z_1, \ldots, Z_{h-1}$, there are exactly κ edges leaving the vertex s' in H'', each edge lying on a distinct path $P \in \mathcal{P}_{h-1}$. If edge e is leaving s' in H'', and e lies on P, then it is incident on v_P, and so exactly one path of $\mathcal{Q}$ originates at v_P. We denote this path by Q_P. For each path $P \in \mathcal{P}_{h-1}$, we let P^* be the path obtained from P by replacing P' with Q_P. Notice that, since the paths in $\mathcal{P}_{h-1}$ are internally disjoint from B, exactly κ edges are incident on t' in graph H', each of which is incident on a distinct vertex of B in H. It is now easy to verify that the set $\mathcal{P}_h$ contains κ node-disjoint paths, connecting the vertices of A to the vertices of B, and the paths in $\mathcal{P}_h$ are internally disjoint from $V(C) \cup V(Y)$ and monotone with respect to $Z_1, \ldots, Z_h$. We return $\mathcal{P}_r$ as the output of the algorithm.

B Proofs Omitted from Section 3

B.1 Proof of Theorem 3.1

The following definition and observation allow us to slightly relax the problem.

Definition B.1 *Let $c \geq 1$ be an integer, and let $\mathcal{M}' \subseteq \mathcal{M}$ be a subset of the demand pairs. Given a constraint $K = (i, a, b, w)$ of type 1 or 2, we say that $\mathcal{M}'$ violates K by a factor of at most c, iff the number of the demand pairs $(s, t) \in \mathcal{M}'$ with either s or t lying in (a, b) is at most cw. Likewise, given a constraint $K = (i, a, b, w)$ of type 3 or 4, we say that $\mathcal{M}'$ violates K by a factor of at most c iff the number of the demand pairs in $\mathcal{M}'$ crossing K is at most cw.*

Observation B.1 *There is an efficient algorithm, that, given a* DPSP *instance $(\sigma, \sigma', \mathcal{M}, \mathcal{K})$ and a non-crossing set $\mathcal{M}' \subseteq \mathcal{M}$ of demand pairs that violates every constraint in $\mathcal{K}$ by a factor of at most c, for any integer $c > 1$, computes a subset $\mathcal{M}'' \subseteq \mathcal{M}'$ of at least $|\mathcal{M}'|/c$ demand pairs, satisfying all constraints in $\mathcal{K}$.*

Proof: Assume that $\mathcal{M}' = \{(s_{j_1}, t_{j_1}), \ldots, (s_{j_z}, t_{j_z})\}$, where $s_{j_1} \prec \cdots \prec s_{j_z}$ and $t_{j_1} \prec \cdots \prec t_{j_z}$. Let $\mathcal{M}'' = \{(s_{j_\ell}, t_{j_\ell}) \mid 1 \leq \ell \leq z \text{ and } \ell \equiv 1 \mod c\}$. Clearly, $\mathcal{M}'' \subseteq \mathcal{M}'$ and $|\mathcal{M}''| \geq |\mathcal{M}'|/c$. We now claim that all constraints in $\mathcal{K}$ are satisfied by $\mathcal{M}''$.

Indeed, consider any constraint $K = (i, a, b, w) \in \mathcal{K}$. Assume first that K is a type-1 constraint. Then at most cw demand pairs in $\mathcal{M}'$ have a source vertex in the interval (a, b). It is easy to see that the number of the demand pairs of $\mathcal{M}''$ that have a source vertex in the interval (a,b) is at most w. If K is a type-2 constraint, the argument is similar. Assume now that K is a type-3 constraint (the case where it is a type-4 constraint is symmetric). Let $\mathcal{M}_K \subseteq \mathcal{M}'$ be the set of all demand pairs of $\mathcal{M}'$ crossing K. The key observation is that the demand pairs of $\mathcal{M}_K$ appear consecutively in the ordered set $\mathcal{M}'$. Since $|\mathcal{M}_K| \leq cw$, it is easy to see that $|\mathcal{M}'' \cap \mathcal{M}_K| \leq w$, and so set $\mathcal{M}''$ satisfies the constraint K. □

Let $r = \lceil \log |\mathcal{M}| \rceil$, and for $1 \leq j \leq r+1$, set $W_j = 2^j$. We partition the constraints of $\mathcal{K}$ into r levels, where for $1 \leq j \leq r$, the jth level contains all constraints (i,a,b,w) with $W_{j-1} \leq w < 2 \cdot W_{j-1} = W_j$. For all $1 \leq i \leq 4$ and $1 \leq j \leq r$, we denote by $\mathcal{S}_j^{(i)}$ the set of all type-i constraints that belong to level j. Let $\mathcal{S}_j = \bigcup_{i=1}^4 \mathcal{S}_j^{(i)}$ be the set of all level-j constraints.

Our algorithm employs dynamic programming. It is convenient to view the algorithm as constructing r dynamic programming tables - one for each level. Consider some level $1 \leq j \leq r$. Let $I = (x,y) \subseteq \sigma$, $I' = (x', y') \subseteq \sigma'$ be a pair of intervals. We say that it is a *good level-j* pair if the following conditions hold for every level-j constraint $K = (i,a,b,w) \in \mathcal{S}_j$:

C1. if K is a type-1 constraint, then I is not contained in (a,b);

C2. if K is a type-2 constraint, then I' is not contained in (a,b);

C3. if K is a type-3 constraint, then either I is not contained in L_a, or I' is not contained in R_b; and

C4. if K is a type-4 constraint, then either I is not contained in R_a, or I' is not contained in L_b.

For each $1 \leq j \leq r$, let $\mathcal{P}_j$ denote the set of all good level-j pairs of intervals. The level-j dynamic programming table, Π_j contains an entry $\Pi_j[I,I']$ for every good level-j pair $(I,I') \in \mathcal{P}_j$ of intervals. The entry will either remain empty, or it will contain a collection of non-crossing demand pairs from $\mathcal{M}$ of cardinality exactly W_j, whose sources lie in I and destinations lie in I'.

We now describe an efficient algorithm that computes the entries of the dynamic programming tables. We start with $j = 1$. For every pair $(I,I') \in \mathcal{P}_1$ of good level-1 intervals, if there are two distinct non-crossing demand pairs $(s,t),(s',t') \in \mathcal{M}$ with $s,s' \in I$ and $t,t' \in I'$, then we set $\Pi_1[I,I'] = \{(s,t),(s',t')\}$. Otherwise, we set $\Pi_1[I,I'] = \emptyset$.

Assume now that we have constructed the tables for levels $1,\ldots,j-1$, and consider the level-j table, for some $1 < j \leq r$, and its entry $\Pi_j(I,I')$ for some good level-j pair $(I,I') \in \mathcal{P}_j$ of intervals, where $I = (x,y)$ and $I' = (x',y')$. If there exist two pairs of vertices $u,v \in I$ and $u',v' \in I'$ such that $u \prec v$, $u' \prec v'$, and both $\Pi_{j-1}[(x,u),(x',u')]$ and $\Pi_{j-1}[(v,y),(v',y')]$ are non-empty, then we set $\Pi_j[I,I'] = \Pi_{j-1}[(x,u),(x',u')] \cup \Pi_{j-1}[(v,y),(v',y')]$. Otherwise, we set $\Pi_j[I,I'] = \emptyset$. This completes the description of the algorithm that computes the entries of the dynamic programming tables. We now proceed to analyze it, starting with the following easy observations.

Observation B.2 *For all $1 \leq j \leq r$ and $(I,I') \in \mathcal{P}_j$, either $|\Pi_j[I,I']| = 0$ or $|\Pi_j[I,I']| = W_j$.*

Proof: The proof is by induction on j. Clearly, the claim holds for $j = 1$ and all $(I,I') \in \mathcal{P}_1$ by our construction. Consider now some $j > 1$, and assume that the claim holds for all values $j' < j$. Consider some entry $\Pi_j(I,I')$ of the level-j dynamic programming table. Our algorithm either sets $\Pi_j[I,I'] = \emptyset$ or $\Pi_j[I,I'] = \Pi_{j-1}[(x,u),(x',u')] \cup \Pi_{j-1}[(v,y),(v',y')]$. The latter only happens when both $\Pi_{j-1}[(x,u),(x',u')], \Pi_{j-1}[(v,y),(v',y')]$ are non-empty, and so by the induction hypothesis, $|\Pi_{j-1}[(x,u),(x',u')]|, |\Pi_{j-1}[(v,y),(v',y')]| = W_{j-1}$, giving us $|\Pi_j[I,I']| = 2 \cdot W_{j-1} = W_j$. □

Observation B.3 *For all* $1 \leq j \leq r$ *and* $(I, I') \in \mathcal{P}_j$, $\Pi_j[I, I']$ *is a non-crossing subset of the demand pairs in* $\mathcal{M}$, *where every* $(s,t) \in T_j[I, I']$ *has* $s \in I$ *and* $t \in I'$.

Proof: The proof is again by induction on j. Clearly, the claim holds for $j = 1$ and all $(I, I') \in \mathcal{P}_1$ from our construction. Consider now some level $j > 1$ and assume that the claim holds for levels $1, \ldots, (j-1)$. Let $(I, I') \in \mathcal{P}_j$ be a good level-j pair of intervals, with $I = (x, y)$ and $I' = (x', y')$. The claim trivially holds when $\Pi_j[I, I'] = \emptyset$, so we assume that $\Pi_j[I, I'] = \Pi_{j-1}[(x, u), (x', u')] \cup \Pi_{j-1}[(v, y), (v', y')]$, where $x \preceq u \prec v \preceq y$ and $x' \preceq u' \prec v' \preceq y'$. From the induction hypothesis, no two demand pairs from $\Pi_{j-1}[(x, u), (x', u')]$ can cross, and the same holds for the demand pairs of $\Pi_{j-1}[(v, y), (v', y')]$. Since every demand pair $(s, t) \in \Pi_{j-1}[(x, u), (x', u')]$ has $s \in (x, u), t \in (x', u')$, and every demand pair $(s', t') \in \Pi_{j-1}[(v, y), (v', y')]$ has $s' \in (v, y), t' \in (v', y')$, it is immediate to verify that no pair of demands $(s, t) \in \Pi_{j-1}[(x, u), (x', u')]$ and $(s', t') \in \Pi_{j-1}[(v, y), (v', y')]$ can cross, and for every demand pair $(s, t) \in T_j[I, I']$, $s \in I$ and $t \in I'$ holds. □

Observation B.4 *For all* $1 \leq j \leq r$ *and* $(I, I') \in \mathcal{P}_j$, *the set* $\Pi_j[I, I']$ *of demand pairs violates every constraint in* $\mathcal{K}$ *by at most factor* 4.

Proof: Fix some $1 \leq j \leq r$ and $(I, I') \in \mathcal{P}_j$, and consider the corresponding table entry $\Pi_j[I, I']$. The claim holds trivially when $|\Pi_j[I, I']| = 0$, so we assume that $|\Pi_j[I, I']| = W_j$. Consider some constraint $K = (i, a, b, w) \in \mathcal{S}_{j'}^{(i)}$, for some level $1 \leq j' \leq r$. if $j' \geq j$, then $w \geq W_{j-1}$ must hold, while $|\Pi_j[I, I']| = W_j$, so the constraint is violated by the factor of at most 4.

Consider now the case where $j' < j$. Assume first that $i = 1$. Note that $\Pi_j[I, I']$ is the union of exactly $2^{j-j'}$ non-empty level-j' table entries, each of which contains a set of demand pairs of cardinality exactly $W_{j'}$. Let $\mathcal{R}$ be the set of all these level-j' table entries. We claim that there are at most two table entries $\Pi_{j'}(\hat{I}, \hat{I}') \in \mathcal{R}$ with $\hat{I} \cap (a, b) \neq \emptyset$. Indeed, assume for contradiction that there are 3 such distinct entries in $\mathcal{R}$, say $\Pi_{j'}[I_1, I_1'], \Pi_{j'}[I_2, I_2']$, and $\Pi_{j'}[I_3, I_3']$, with $I_1 \cap (a, b), I_2 \cap (a, b), I_3 \cap (a, b) \neq \emptyset$. Then at least one of the intervals I_1, I_2, I_3 must be contained in (a, b), violating our condition for good level-j' pairs of intervals. We conclude that the number of the demand pairs in $\Pi_j[I, I']$ with a source vertex in (a, b) is bounded by $2 \cdot W_{j'} = 4 \cdot W_{j'-1} \leq 4 \cdot w$. The proof for the case where $i = 2$ is analogous.

Assume now that K is a type-3 constraint. As before, $\Pi_j[I, I']$ is the union of exactly $2^{j-j'}$ non-empty level-j' table entries, each of which stores a set of demand pairs of cardinality exactly $W_{j'}$. Let $\mathcal{R}$ be the set of these level-j' table entries. We claim that there are at most two entries $\Pi_{j'}(\hat{I}, \hat{I}') \in \mathcal{R}$, with $\hat{I} \cap L_a \neq \emptyset$ and $\hat{I}' \cap R_b \neq \emptyset$. Indeed, assume otherwise, and let $\Pi_{j'}[I_1, I_1'], \Pi_{j'}[I_2, I_2'], \Pi_{j'}[I_3, I_3']$ be three distinct entries in $\mathcal{R}$, such that for each $1 \leq \ell \leq 3$, $I_\ell \cap L_a \neq \emptyset$ and $I_\ell' \cap R_b \neq \emptyset$. Assume that I_1, I_2, I_3 appear on σ in this order, and recall that from our construction they are disjoint. Then it is easy to see that $I_2 \subseteq L_a$ and $I_2' \subseteq R_b$ must hold, contradicting our definition of good level-j' pairs of terminals. Therefore, there are at most two entries $\Pi_{j'}(\hat{I}, \hat{I}') \in \mathcal{R}$, with $\hat{I} \cap L_a \neq \emptyset$ and $\hat{I}' \cap R_b \neq \emptyset$. Demand pairs participating in solutions corresponding to other entries in $\mathcal{R}$ cannot cross K, and so the number of the demand paris crossing K is at most $2 \cdot W_{j'} = 4 \cdot W_{j'-1} \leq 4 \cdot w$. The case where K is a type-4 constraint is treated similarly. □

From the discussion so far, every entry of every dynamic programming table contains a non-crossing set of demand pairs, that violates every constraint of $\mathcal{K}$ by at most factor 4. Let $\tilde{\mathcal{M}}$ be the largest-cardinality set of demand pairs stored in any entry any of the tables, and let OPT be the value of the optimal solution to the DPSP problem instance. The following theorem is central to our analysis.

Theorem B.5 *If* $\mathsf{OPT} \geq 2$, *then* $|\tilde{\mathcal{M}}| \geq \mathsf{OPT}/2$.

We can now apply Observation B.1 to compute a subset $\tilde{\mathcal{M}}' \subseteq \tilde{\mathcal{M}}$ of at least $|\tilde{\mathcal{M}}|/4 \geq \mathsf{OPT}/8$ non-crossing demand pairs satisfying all constraints in $\mathcal{K}$ (if $|\tilde{\mathcal{M}}| = 0$, then $\mathsf{OPT} \leq 1$ must hold, and finding an optimal solution is trivial). In order to complete the proof of Theorem 3.1, it now remains to prove Theorem B.5.

Proof of Theorem B.5. Denote $\kappa = \mathsf{OPT}$, and let $\mathcal{M}^* = \{(s_1, t_1), \ldots, (s_\kappa, t_\kappa)\}$ be the optimal solution to the DPSP instance, where $s_1 \prec \ldots \prec s_\kappa$ and $t_1 \prec \ldots \prec t_\kappa$. Let r' be the largest value for which $\kappa/W_{r'} \geq 1$ (this is well-defined since we have assumed that $\mathsf{OPT} \geq 2$). Let $\mathcal{M}^{**} \subseteq \mathcal{M}^*$ contain the first $W_{r'}$ demand pairs of $\mathcal{M}^*$, so $|\mathcal{M}^{**}| \geq |\mathcal{M}^*|/2$. We will show that some entry of the level-r' dynamic programming table stores a solution whose cardinality is at least $W_{r'}$.

For every level $1 \leq j \leq r'$, we define a partition $\mathcal{S}_j$ of $\mathcal{M}^{**}$ into $2^{r'-j}$ subsets, each containing exactly W_j consecutive demand pairs from $\mathcal{M}^{**}$. For every set $S \in \mathcal{S}_j$ of the partition, we define a pair $(I(S), I'(S))$ of intervals, with $I(S) \subseteq \sigma$ and $I'(S) \subseteq \sigma'$, as follows. Let $(s,t),(s',t')$ be the first and the last demand pairs of S, respectively (this is well-defined since the demand pairs are non-crossing). Then $I(S) = (s, s')$ and $I'(S) = (t, t')$. We denote by $\mathcal{Q}_j$ the resulting collection of $2^{r'-j}$ pairs of intervals. Note that for every pair $(I_1, I'_1), (I_2, I'_2) \in \mathcal{Q}_j$, $I_1 \cap I_2 = \emptyset$ and $I'_1 \cap I'_2 = \emptyset$. We need the following claim.

Claim B.6 *For every $1 \leq j \leq r'$, every pair $(I, I') \in \mathcal{Q}_j$ of intervals is a good level-j pair.*

Proof: Fix some $1 \leq j \leq r'$ and some pair $(I, I') \in \mathcal{Q}_j$ of intervals. From our definition of the pairs of intervals in $\mathcal{Q}_j$, there are exactly W_j demand pairs (s,t) in $\mathcal{M}^*$ with $s \in I$ and $t \in I'$. Consider any level-j constraint $K = (i, a, b, w) \in \mathcal{S}_j$, and recall that $W_{j-1} \leq w < W_j$ must hold.

Assume first that K is a type-1 constraint. Then I cannot be contained in (a, b), since then $\mathcal{M}^*$ would have $W_j > w$ demand pairs whose sources lie in I, and hence in (a,b). The analysis for type-2 constraints is similar.

Assume now that K is a type-3 constraint, and assume for contradiction that $I \subseteq L_a$ and $I' \subseteq R_b$. Then $\mathcal{M}^*$ has $W_j > w$ demand pairs (s,t) with $s \in I$ and $t \in I'$, each of which crosses the constraint K, a contradiction. The case where K is a type-4 constraint is proved similarly. □

From the above claim, for all $1 \leq j \leq r'$ and $(I, I') \in \mathcal{Q}_j$, there is an entry $\Pi_j[I, I']$ in the level-j dynamic programming table. It is now enough to show that each such entry contains a solution of cardinality W_j.

Claim B.7 *For each $1 \leq j \leq r'$, for every pair $(I(S), I'(S)) \in \mathcal{Q}_j$ of intervals, entry $\Pi_j[I(S), I'(S)]$ contains a solution of value W_j.*

Proof: The proof is by induction on j. The claim clearly holds for $j = 1$, since there are two distinct non-crossing demand pairs $(s,t),(s',t')$ with $s, s' \in I(S)$, $t, t' \in I'(S)$ - the two demand pair lying in S. Assume now that the claim holds for levels $1, \ldots, j-1$, for some $1 < j \leq r'$, and we would like to prove it for j.

Our definition of the partitions $\mathcal{S}_\ell$ of $\mathcal{M}^{**}$ ensures that there are exactly two distinct sets $S_1, S_2 \in \mathcal{S}_{j-1}$, with $S_1, S_2 \subseteq S$ (and in fact $S_1 \cup S_2 = S$). From the induction hypothesis, the entries of Π_{j-1} corresponding to pairs $(I(S_1), I'(S_1))$ and $(I(S_2), I'(S_2))$ each contain W_{j-1} demand pairs, and so $\Pi_j[I(S), I'(S)]$ must contain W_j demand pairs. □

Recall that $\mathcal{S}_{r'}$ contains a single set of demand pairs - the set $\mathcal{M}^{**}$. We conclude that the corresponding entry of the level-r' dynamic programming table contains $W_{r'} \geq \mathsf{OPT}/2$ demand pairs. □

B.2 Approximation Algorithm for NDP-Disc

In this section we prove Theorem 1.1 for NDP-Disc. The proof builds on the work of Robertson and Seymour [RS86], who gave a precise characterization of the instances NDP-Disc, where all demand pairs can be routed simultaneously. Many of the definitions below are from [RS86]. Let Σ be a disc, whose boundary is denoted by Γ, and let G be any graph drawn on Σ. Suppose we are given a set $\mathcal{M} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of pairs of vertices of G, called demand pairs, and let $\mathcal{T}$ be the set of all vertices participating in the demand pairs in $\mathcal{M}$, that we refer to as terminals. We identify the graph G with its drawing. A *region* of G is a connected component of $\Sigma \setminus G$. We say that the drawing of G is *semi-proper* if no inner point on an image of an edge of G lies on Γ, and we say that it is *proper* with respect to $\mathcal{M}$, if additionally $V(G) \cap \Gamma = \mathcal{T}$.

Suppose we are given a planar graph G, together with a set $\mathcal{M}$ of demand pairs, such that G is drawn properly (with respect to $\mathcal{M}$) on Σ. Let W be a set of points on Γ, constructed as follows. First, we add to W all points corresponding to the vertices of $\mathcal{T}$. Next, for every segment β of $\Gamma \setminus \mathcal{T}$, we add one arbitrary point $p \in \beta$ to W. A vertex v of G is *peripheral* if $v \in \Gamma$, and a region of G is peripheral, if it contains a segment of $\Gamma \setminus \mathcal{T}$. Given two points $x, y \in W$, we denote by $\Delta_{\mathcal{M}}(x, y)$ the total number of the demand pairs $(s, t) \in \mathcal{M}$, where either $\{s, t\} \cap \{x, y\} \neq \emptyset$, or s and t belong to different segments of $\Gamma \setminus \{x, y\}$. We need the following definition.

Definition B.2 *Given G, $\mathcal{M}$ and W as above, for any $x, y \in W$, an (x, y)-chain is a sequence $A_1, A_2, \ldots, A_r$, such that:*

- *for all $1 \leq i < r$, one of A_i, A_{i+1} is a vertex of G, the other is a region, and they are incident;*
- *if A_1 is a vertex then $A_1 = x$; if A_1 is a region then $x \in A_1$;*
- *similarly, if A_r is a vertex then $A_r = y$, and if A_r is a region then $y \in A_r$; and*
- *for all $1 \leq i \leq r$, A_i is peripheral if and only if $i = 1$ or $i = r$.*

The length of an (x, y)-chain is the number of its terms that are vertices. The redundancy of an (x, y)-chain is its length minus $\Delta_{\mathcal{M}}(x, y)$.

The following theorem, proved by Roberston and Seymour [RS86] characterizes routable sets of demand pairs for the NDP-Disc problem.

Theorem B.8 (Theorem 3.6 in [RS86]) *Let G be a planar graph properly drawn on a disc Σ, with respect to a set $\mathcal{M}$ of demand pairs, and let W be defined as above. Then there is a set of node-disjoint paths, routing all demand pairs in $\mathcal{M}$ if and only if: (i) $\mathcal{M}$ is a non-crossing set of demand pairs; and (ii) for all $x, y \in W$, every (x, y)-chain has a non-negative redundancy. Moreover, there is an efficient algorithm to determine whether these conditions hold, and if so, to find a routing of the demand pairs in $\mathcal{M}$.*

Notice that if G is a graph drawn on a disc Σ, such that all terminals appear on the boundary Γ of the disc, then, by slightly altering Γ, we can ensure that the drawing of G is proper with respect to $\mathcal{M}$. Therefore, we assume from now on that we are given a proper drawing φ of G on Σ with respect to $\mathcal{M}$. We prove Theorem 1.1 for NDP-Disc in three steps. In the first step, we prove a stronger version of the theorem for the case where G is connected and the set $\mathcal{M}$ of terminals is 1-split. In the second step, we prove the theorem for the case where G is 2-connected, and we make no assumptions on the demand pairs. In the third step we prove the theorem without any additional assumptions.

B.2.1 Special Case: G is Connected and $\mathcal{M}$ is 1-Split

In this section, we prove the following theorem.

Theorem B.9 *There is an efficient algorithm, that, given a connected planar graph G with a set $\mathcal{M}$ of demand pairs, and a proper drawing of G on the disc Σ with respect to $\mathcal{M}$, where $\mathcal{M}$ is 1-split with respect to the disc boundary, computes a routing of $\mathsf{OPT}(G, \mathcal{M})/8$ demand pairs of $\mathcal{M}$ via node-disjoint paths in G.*

Recall that if $\mathcal{M}$ is 1-split with respect to the boundary Γ of the disc Σ, then we can partition Γ into two disjoint segments, σ and σ', such that for every demand pair $(s, t) \in \mathcal{M}$, one of the vertices s, t lies on σ, and the other on σ'. We assume without loss of generality that all source vertices of the demand pairs in $\mathcal{M}$ lie on σ, and all destination vertices lie on σ'. Let $\mathcal{T}$ be the set of all vertices participating in the demand pairs in $\mathcal{M}$, that we refer to as terminals. We denote by S and T the sets of the source and the destination vertices of the demand pairs in $\mathcal{M}$, respectively. As before, we construct a set W of points: start with $W = \mathcal{T}$; then for every segment of $\Gamma \setminus \mathcal{T}$, add one arbitrary point of that segment to W.

The idea is to reduce this problem to the DPSP problem. We let $\tilde{\sigma}$ be a directed path, whose vertices correspond to the points of $W \cap \sigma$, ordered in their counter-clock-wise order on Γ. We let $\tilde{\sigma}'$ be a directed path, whose vertices correspond to the points of $W \cap \sigma'$, ordered in their clock-wise order on Γ. Therefore, all vertices of S lie on $\tilde{\sigma}$, and all vertices of T lie on $\tilde{\sigma}'$. Moreover, $\mathcal{M}'$ is a non-crossing set of demand pairs with respect to $\tilde{\sigma}$ and $\tilde{\sigma}'$ iff it is non-crossing with respect to Γ.

Suppose we are given any pair (x, y) of vertices of W. Let $\gamma(x, y)$ denote the shortest G-normal curve connecting x to y in the drawing of G on Σ, so that $\gamma(x, y)$ is contained in the disc Σ. Curve $\gamma(x, y)$ can be found efficiently by considering the graph G' dual to G, deleting the vertex corresponding to the outer face of G from it, and computing shortest paths between appropriately chosen vertices of G' (that correspond to the faces of G incident on x and y). Let $\ell(x, y)$ be the length of $\gamma(x, y)$. Notice that, since G is connected, if $\Delta_{\mathcal{M}}(x, y) \geq 1$, then $\ell(x, y) \geq 1$. We now construct a collection $\mathcal{K}$ of constraints of the DPSP probem, as follows.

For every pair x, y of vertices of $W \cap \tilde{\sigma}$, with $\Delta_{\mathcal{M}}(x, y) \geq 1$, we add a type-1 constraint $(1, x, y, \ell(x, y))$ to $\mathcal{K}$. Similarly, for every pair x, y of vertices of $W \cap \tilde{\sigma}'$, with $\Delta_{\mathcal{M}}(x, y) \geq 1$, we add a type-2 constraint $(2, x, y, \ell(x, y))$ to $\mathcal{K}$. For every pair $x \in W \cap \tilde{\sigma}$, $y \in W \cap \tilde{\sigma}'$ of vertices, with $\Delta_{\mathcal{M}}(x, y) \geq 1$, we add a type-3 constraint $(3, x, y, \ell(x, y))$ and a type-4 constraint $(4, x, y, \ell(x, y))$ to $\mathcal{K}$. This finishes the definition of the DPSP problem instance. The following observation is immediate.

Observation B.10 *Let $\mathcal{M}' \subseteq \mathcal{M}$ be any set of demand pairs that can be routed via disjoint paths in G. Then $\mathcal{M}'$ is a valid solution to the DPSP problem instance.*

We apply the 8-approximation algorithm for the DPSP problem, to obtain a set $\mathcal{M}'$ of non-crossing demand pairs (with respect to $\tilde{\sigma}$ and $\tilde{\sigma}'$) satisfying all constraints in $\mathcal{K}$, with $|\mathcal{M}'| \geq |\mathsf{OPT}(G, \mathcal{M})|/8$. As observed above, the pairs in $\mathcal{M}'$ are non-crossing with respect to Γ. It now only remains to show that all demand pairs in $\mathcal{M}'$ can be routed in G. The algorithm from Theorem B.8 can then be used to find the routing. The following theorem will finish the proof of Theorem B.9.

Theorem B.11 *Let $\mathcal{M}' \subseteq \mathcal{M}$ be a set of non-crossing demand pairs (with respect to $\tilde{\sigma}$ and $\tilde{\sigma}'$), that satisfy all constraints in $\mathcal{K}$. Then all demand pairs in $\mathcal{M}'$ can be routed in G.*

Proof: Let $\mathcal{T}' = \mathcal{T}(\mathcal{M}')$, and let φ be the current proper drawing of G with respect to $\mathcal{M}$. Notice that φ is not necessarily a proper drawing of G with respect to $\mathcal{M}'$, since the vertices of $\mathcal{T} \setminus \mathcal{T}'$ may lie on Γ. We can obtain a proper drawing φ' of G with respect to $\mathcal{M}'$ by moving all such terminals inside the disc, so they no longer lie on Γ. It is immediate to verify that the demand pairs in $\mathcal{M}'$ remain non-crossing with respect to Γ in the new drawing.

As before, we construct a set W' of points of Γ, by first adding all points corresponding to the terminals of $\mathcal{T}'$ to W'. Additionally, for every segment β of $\Gamma \setminus \mathcal{T}'$, we add an arbitrary point of β to W'. It now remains to show that for every pair x, y of points in W', for every (x, y)-chain in φ', the redundancy of the chain (with respect to $\mathcal{M}'$) is non-negative.

Assume otherwise. Let $x, y \in W'$ be any pair of points, and let $\mathcal{A} = (A_1, \ldots, A_r)$ be an x-y chain in the drawing φ', such that the redundancy of $\mathcal{A}$ is negative. We modify the chain $\mathcal{A}$, by replacing A_1 and A_r with elements A'_1 and A'_r, as follows.

If A_1 is a vertex of $\mathcal{T}'$, then we let $A'_1 = A_1$, and we let $x' = A'_1$. Otherwise, A_1 is a region of G in the drawing φ'. Let $v = A_2$, so $v \in V(G)$. Then v must lie on a boundary of some peripheral region R of G in the drawing φ. Let $A'_1 = R$, let $\beta(R)$ be the segment of Γ that serves as part of the boundary of the region R, and let $x' \in W$ be the point lying on the interior of $\beta(R)$. Similarly, if A_r is a vertex of $\mathcal{T}'$, then we let $A'_r = A_r$, and we let $y' = A'_r$. Otherwise, A_r is a region of G in the drawing φ'. Let $v' = A_{r-1}$, so $v' \in V(G)$. Then v' must lie on a boundary of some peripheral region R' of G of the drawing φ. Let $A'_r = R'$, let $\beta(R')$ be the segment of Γ that serves as part of the boundary of the region R', and let $y' \in W$ be the point lying on the interior of $\beta(R)$. We then obtain a sequence $\mathcal{A}' = (A'_1, A_2, \ldots, A_{r-1}, A'_r)$ (it may not be a valid chain for the drawing φ, since some of the elements A_i, for $1 < i < r'$, may be peripheral with respect to φ). We can then construct a G-normal curve γ' in the original drawing φ of G, connecting x' to y', such that $\gamma' \cap V(G)$ only contains the vertices that participate in $\mathcal{A}'$, and γ' is contained in Σ. Let ℓ denote the length of the chain $\mathcal{A}$, and let $\Delta = |\Delta_{\mathcal{M}'}(x, y)|$. We can assume that $\Delta \geq 1$, since otherwise it is immediate to verify that $\Delta \leq \ell$. Then $\ell(x', y') \leq \ell$, from the definition of $\ell(x', y')$. Notice that one of the segments of $\Gamma \setminus \{x, x'\}$ contains no terminals of $\mathcal{T}'$, and the same holds for one of the segments of $\Gamma \setminus \{y, y'\}$. We now consider three cases.

Assume first that both x' and y' lie on $\tilde{\sigma}$. Then pair $(s, t) \in \Delta_{\mathcal{M}'}(x, y)$ iff s belongs to the sub-path (x', y') of $\tilde{\sigma}$. Since we assumed that $\Delta \geq 1$, we get that $\Delta_{\mathcal{M}}(x', y') \geq 1$, so constraint $(1, x', y', \ell(x', y'))$ belongs to $\mathcal{K}$, and we get that $\Delta \leq \ell(x', y') \leq \ell$, a contradiction.

The case where $x', y' \in \tilde{\sigma}'$ is dealt with similarly.

Assume now that $x' \in \tilde{\sigma}$ and $y' \in \tilde{\sigma}'$. As before, since we assumed that $\Delta \geq 1$, we get that $\Delta_{\mathcal{M}}(x', y') \geq 1$. Consider the two corresponding type-3 and type-4 constraints $K = (3, x', y', \ell(x', y')), K'' = (4, x', y', \ell(x', y')) \in \mathcal{K}$. Let $L_{x'}, R_{x'}$ be the segments of $\tilde{\sigma}$ from its first endpoint to x', and from x' to its last endpoint, respectively, where both segments include x'. Define segments $L_{y'}, R_{y'}$ of $\tilde{\sigma}'$ similarly. Recall that a demand pair (s, t) crosses K iff $s \in L_{x'}$ and $t \in R_{y'}$, and it crosses K' iff $s \in R_{x'}$ and $t \in L_{y'}$. Let $\mathcal{M}_1 \subseteq \mathcal{M}'$ be the set of the demand pairs crossing $\mathcal{K}$, and let $\mathcal{M}_2 \subseteq \mathcal{M}'$ be the set of the demand pairs crossing $\mathcal{K}'$. Since $\mathcal{M}'$ is a non-crossing set of demand pairs, it is easy to verify that either $\mathcal{M}_1 \setminus \{(x', y')\} = \emptyset$, or $\mathcal{M}_2 \setminus \{(x', y')\} = \emptyset$. We assume without loss of generality that it is the latter. Notice that if $(x', y') \in \mathcal{M}'$, then it belongs to both $\mathcal{M}_1$ and $\mathcal{M}_2$. Constraint $(3, x', y', \ell(x', y'))$ then ensures that $|\mathcal{M}_1 \cup \mathcal{M}_2| \leq |\mathcal{M}_1| \leq \ell(x', y') \leq \ell$. It is easy to verify that $\Delta_{\mathcal{M}'}(x, y) \subseteq \mathcal{M}_1 \cup \mathcal{M}_2$, and so $\Delta \leq \ell$, a contradiction. $\square$

B.2.2 Special Case: G is 2-connected

The goal of this section is to prove the following theorem.

Theorem B.12 *There is an efficient algorithm, that, given any 2-connected planar graph G, and a set $\mathcal{M}$ of k demand pairs for G, such that G is properly drawn on a disc with respect to $\mathcal{M}$, returns an $O(\log k)$-approximate solution to problem $(G, \mathcal{M})$.*

Proof: Let $|\mathcal{M}| = k$, and let $z = 4\lceil \log k \rceil$. We use Lemma 2.3 to compute a partition $\mathcal{M}^1, \ldots, \mathcal{M}^z$ of $\mathcal{M}$ into z disjoint subsets, so that for each $1 \leq i \leq z$, $\mathcal{M}^i$ is r_i-split for some integer r_i. We prove the following lemma.

Lemma B.13 *There is an efficient algorithm, that, given an index $1 \leq i \leq z$, computes a solution to instance $(G, \mathcal{M}^i)$ routing $\Omega(\mathsf{OPT}(G, \mathcal{M}^i))$ demand pairs.*

Theorem B.12 then easily follows from Lemma B.13, since there is some index $1 \leq i \leq z$, for which $\mathsf{OPT}(G, \mathcal{M}^i) = \Omega(\mathsf{OPT}(G, \mathcal{M})/\log k)$. In the rest of this section, we focus on proving Lemma B.13.

For simplicity, we denote $\mathcal{M}^i$ by $\mathcal{M}$ and r_i by r. We assume that we are given a partition $\{\mathcal{M}_1, \ldots, \mathcal{M}_r\}$ of $\mathcal{M}$, and a collection $\sigma_1, \ldots, \sigma_{2r}$ of disjoint segments of Γ, such that for all $1 \leq j \leq r$, for every demand pair $(s, t) \in \mathcal{M}_j$, $s \in \sigma_{2j-1}$ and $t \in \sigma_{2j}$.

Clearly, for each $1 \leq j \leq r$, set $\mathcal{M}_j$ is 1-split. Therefore, we can use the algorithm from Section B.2.1 in order to compute a set $\mathcal{P}_j$ of disjoint paths, routing a subset $\mathcal{M}'_j \subseteq \mathcal{M}_j$ of demand pairs, with $|\mathcal{M}'_j| \geq \Omega(\mathsf{OPT}(G, \mathcal{M}_j))$. Since $\sum_{j=1}^{r} \mathsf{OPT}(G, \mathcal{M}_j) \geq \mathsf{OPT}(G, \mathcal{M})$, we get that $\sum_{j=1}^{r} |\mathcal{M}'_j| \geq \Omega(\mathsf{OPT}(G, \mathcal{M}))$.

For every $1 \leq j \leq r$, we compute a subset $\mathcal{M}''_j \subseteq \mathcal{M}'_j$ of demand pairs, as follows. If $|\mathcal{M}'_j| \leq 3$, then we let $\mathcal{M}''_j$ contain any demand pair from $\mathcal{M}'_j$. Otherwise, we assume that $\mathcal{M}'_j = \left((s^j_1, t^j_1), \ldots, (s^j_{q_j}, t^j_{q_j})\right)$, where the vertices $s^j_1, s^j_2, \ldots, s^j_{q_j}$ appear in this counter-clock-wise order on σ_{2j-1}. We then add to $\mathcal{M}''_j$ all demand pairs (s^j_a, t^j_a), where $a = 1$ modulo 3. Notice that $\sum_{j=1}^{r} |\mathcal{M}''_j| \geq \sum_{j=1}^{r} \Omega(|\mathcal{M}'_j|) \geq \Omega(\mathsf{OPT}(G, \mathcal{M}))$. It is now enough to prove that all demand pairs in $\mathcal{M}'' = \bigcup_{j=1}^{r} \mathcal{M}''_j$ can be routed in G. It is easy to verify that the demand pairs in $\mathcal{M}''$ are non-crossing, since for each $1 \leq j \leq r$, the demand pairs in $\mathcal{M}'_j$ are non-crossing, and pairs that belong to different sets $\mathcal{M}_j$ cannot cross. Let $\mathcal{T}'' = \mathcal{T}(\mathcal{M}'')$, and let φ'' is a proper drawing of G in Σ with respect to $\mathcal{M}''$, obtained from the original drawing φ by moving all terminals of $\mathcal{T} \setminus \mathcal{T}''$ to the interior of the disc. We define a set W of points on Γ as before: it contains all terminals of $\mathcal{T}''$, and for every segment of $\Gamma \setminus \mathcal{T}''$, W contains an arbitrary point in the interior of the segment.

Let $(x, y) \in W$ be any pair of points, and let $\mathcal{A} = (A_1, A_2, \ldots, A_p)$ be any (x, y)-chain. Let ℓ denote the length of $\mathcal{A}$. It is now enough to prove that $|\Delta_{\mathcal{M}''}(x, y)| \leq \ell$. Assume first that for some $1 \leq j \leq z$, $x, y \in \sigma_{2j-1} \cup \sigma_{2j}$. Then $\Delta_{\mathcal{M}''}(x, y)$ only contains demand pairs from $\mathcal{M}''_j$. Since all demand pairs in $\mathcal{M}''_j$ can be routed in G via node-disjoint paths, $|\Delta_{\mathcal{M}''}(x, y)| \leq \ell$.

Assume now that $x \in \sigma_{2j-1} \cup \sigma_{2j}$ and $y \in \sigma_{2j'-1} \cup \sigma_{2j'}$ for some $j \neq j'$. Then $\Delta_{\mathcal{M}''}(x, y)$ only contains demand pairs from $\mathcal{M}''_j \cup \mathcal{M}''_{j'}$. Let $\mathcal{N}_1 = \Delta_{\mathcal{M}''}(x, y) \cap \mathcal{M}''_j$ and $\mathcal{N}_2 = \Delta_{\mathcal{M}''}(x, y) \cap \mathcal{M}''_{j'}$. Let $\mathcal{N}'_1 = \Delta_{\mathcal{M}'}(x, y) \cap \mathcal{M}'_j$, and let $\mathcal{N}'_2 = \Delta_{\mathcal{M}'}(x, y) \cap \mathcal{M}'_{j'}$. Since both $\mathcal{M}'_j$ and $\mathcal{M}'_{j'}$ can be routed in G via node-disjoint paths, $|\mathcal{N}'_1|, |\mathcal{N}'_2| \leq \ell$. From our selection of the sets $\mathcal{M}''_j$, $\mathcal{M}''_{j'}$ of demand pairs, $|\mathcal{N}_1| \leq \max\{|\mathcal{N}'_1|/2, 1\}$, and $|\mathcal{N}_2| \leq \max\{|\mathcal{N}'_2|/2, 1\}$. Since graph G is 2-vertex connected, we can assume that either $|\Delta_{\mathcal{M}''}(x, y)| \leq 1$, or $\ell \geq 2$. In the former case, $\ell \geq |\Delta_{\mathcal{M}''}(x, y)|$ since the graph is connected. In the latter case, we are now guaranteed that $|\Delta_{\mathcal{M}''}(x, y)| = |\mathcal{N}'_1| + |\mathcal{N}'_2| \leq \ell$. □

B.2.3 The General Case

In this section, we complete the proof of Theorem 1.1 for NDP-Disc. We assume that the input graph G is connected, since otherwise we can solve the problem separately on each connected component of

G.

We use a block-decomposition of G. Recall that a block of G is a maximal 2-node-connected component of G. A block-decomposition of G is a tree τ, whose vertex set consists of two subsets, $V(\tau) = V_1 \cup V_2$, where V_1 is the set of all cut-vertices of G, and V_2 contains a vertex v_B for every block B of G. There is an edge between a vertex $u \in V_1$ and $v_B \in V_2$ iff $u \in v_B$. We choose an arbitrary vertex $r \in V_1$ as the root of τ. We assume that the value of the optimal solution is at least 10, as otherwise we can route a single demand pair. We then discard from $\mathcal{M}$ all demand pairs in which r participates – this changes the value of the optimal solution by at most 1.

Over the course of the algorithm, we will gradually change the tree τ, by deleting vertices from it. Given any vertex u in the current tree τ, we let τ_u denote the subtree of τ rooted at u, and we let G_u be the subgraph of G induced by the union of all blocks B with $v_B \in V(\tau_u)$. For every block B, we denote by $p(B)$ the unique vertex of B that serves as the parent of the vertex v_B in τ. As the tree τ changes, so do the trees τ_u and the graphs G_u.

If any block B of G contains all the terminals, then we can apply the algorithm from Theorem B.12 to instance $(B, \mathcal{M})$ of $\mathsf{NDP\text{-}Disc}$ to obtain an $O(\log k)$-approximate solution. Otherwise, for every block B, if we denote by $\Gamma(B)$ the set of all cut vertices of G that belong to B, and by $\mathcal{T}(B)$ the set of all terminals lying in B, then we can draw B inside a disc, so that the vertices of $\Gamma(B) \cup \mathcal{T}(B)$ lie on its boundary.

We gradually construct a solution $\mathcal{P}$ to the $\mathsf{NDP\text{-}Disc}$ instance $(G, \mathcal{M})$, starting from $\mathcal{P} = \emptyset$. Throughout the algorithm, we ensure that all paths in $\mathcal{P}$ are disjoint from the vertices of G_r, where r is the root vertex of τ, and G_r is computed with respect to the current tree τ. Clearly, the invariant holds at the beginning of the algorithm.

We maintain a collection $\mathcal{B}$ of blocks, that is initialized to $\mathcal{B} = \emptyset$. For every block $B \in \mathcal{B}$, we will define two subsets, $\mathcal{M}_B, \mathcal{M}'_B \subseteq \mathcal{M}$ of demand pairs, so that eventually, $\{\mathcal{M}_B, \mathcal{M}'_B \mid B \in \mathcal{B}\}$ is a partition of $\mathcal{M}$.

In every iteration, we consider all vertices $v_B \in V_2$ that belong to the current tree τ, such that $G_{v_B} \setminus p(B)$ contains at least one demand pair $(s,t) \in \mathcal{M}$. Among all such vertices, we choose one that is furthest from the root of the tree, breaking ties arbitrarily. Let v_B be the selected vertex. We add B to $\mathcal{B}$, and we define two new subsets of demand pairs $\mathcal{M}_B, \mathcal{M}'_B$, as follows. Set $\mathcal{M}_B$ contains all demand pairs (s,t) with $s,t \in G_{v_B} \setminus p(B)$. Set $\mathcal{M}'_B$ contains all demand pairs (s,t) with exactly one of s,t lying in $G_{v_B} \setminus p(B)$, while the other terminal must belong to G_r. Notice that any path connecting any demand pair in $\mathcal{M}'_B$ must use the vertex $p(B)$.

Next, we construct a new instance of the $\mathsf{NDP\text{-}Disc}$ problem, on the graph B. The corresponding set of demand pairs, that we denote by $\mathcal{N}_B$, is defined as follows. Consider any demand pair $(s,t) \in \mathcal{M}_B$. We define a new demand pair (s',t') representing (s,t), with $s',t' \in V(B)$, and define two paths: $Q(s)$ connecting s to s', and $Q(t)$ connecting t to t'. If $s \in V(B)$, then we let $s' = s$, and $Q(s) = \{s\}$. Otherwise, we let $s' \in V_1$ be the unique child of the vertex v_B in τ, such that $s \in G_{s'}$. Notice that s' must be a vertex of B. We then let $Q(s)$ be any simple path connecting s to s' in graph $G_{s'}$. We define the vertex $t' \in B$, and a path $Q(t)$ connecting t to t' similarly. Notice that it is possible that $s' = t'$. Let $\mathcal{N}_B = \{(s',t') \mid (s,t) \in \mathcal{M}_B\}$ be the resulting set of demand pairs. All vertices participating in the demand pairs in $\mathcal{N}_B$ belong to B. Consider the NDP instance $(B, \mathcal{N})$. It is immediate to verify that we can draw B in a disc, so that all vertices participating in the demand pairs in $\mathcal{M}(B)$ lie on the boundary of the disc, and clearly B is 2-connected. We need the following immediate observation.

Observation B.14 $\mathsf{OPT}(B, \mathcal{N}_B) \geq \mathsf{OPT}(G, \mathcal{M}_B)$.

Proof: Let $\tilde{\mathcal{P}}$ be the optimal solution to instance $\mathsf{OPT}(G, \mathcal{M}_B)$. We can assume that all paths in $\tilde{\mathcal{P}}$ are simple. Let $P \in \tilde{\mathcal{P}}$ be any such path, and assume that P connects some demand pair $(s,t) \in \mathcal{M}_B$. Then it is easy to see that $s', t' \in P$, and moreover, the segment of P between s' and t' is contained in B. By appropriately truncating every path in $\tilde{\mathcal{P}}$, we can obtain a solution to instance $(B, \mathcal{N}_B)$ of the NDP problem of the same value. □

Since B is 2-vertex connected, and can be drawn in a disc with all vertices participating in the demand pairs in $\mathcal{N}_B$ lying on the disc boundary, we can apply the algorithm from Theorem B.12 to instance $(B, \mathcal{N}_B)$, to compute a set $\mathcal{P}(B)$ of node-disjoint paths, routing a subset of at least $\Omega(\mathsf{OPT}(B, \mathcal{N}_B)/\log k)$ demand pairs of $\mathcal{M}_B$ in B. We can assume that $|\mathcal{P}(B)| \geq 1$, since otherwise we can route any demand pair in $\mathcal{N}_B$. We would like to ensure that all paths in $\mathcal{P}(B)$ avoid the vertex $p(B)$. If $|\mathcal{P}(B)| = 1$, then, since B is 2-vertex connected, we can re-route the unique path in $\mathcal{P}(B)$ inside B, so that its endpoints remain the same, but it avoids the vertex $p(B)$ (since $G_{v_B} \setminus p(B)$ contains at least one demand pair, we can ensure that the endpoints of the path are distinct from $p(B)$). Otherwise, we discard from $\mathcal{P}(B)$ the path that uses vertex $p(B)$, if such exists. By concatenating the paths in $\mathcal{P}(B)$ with the paths in $\{Q(s), Q(t) \mid (s,t) \in \mathcal{M}(B)\}$, we obtain a collection $\mathcal{P}'(B)$ of at least $\Omega(\mathsf{OPT}(B, \mathcal{N})/\log k)$ node-disjoint paths, connecting demand pairs in $\mathcal{M}(B)$. We add the paths in $\mathcal{P}'(B)$ to $\mathcal{P}$, and delete from τ all vertices of τ_{v_B}. Since we have ensured that the paths in $\mathcal{P}'(B)$ are disjoint from $p(B)$, the invariant that the paths in $\mathcal{P}$ are disjoint from the new graph G_r continues to hold. The algorithm terminates when no demand pair $(s,t) \in \mathcal{M}$ is contained in G_r. We claim that the resulting collection $\{\mathcal{M}(B), \mathcal{M}'(B) \mid B \in \mathcal{B}\}$ of sets of demand pairs partitions $\mathcal{M}$. Indeed, consider any demand pair $(s,t) \in \mathcal{M}$, and consider the last iteration i when both $s, t \in G_r$. Let v_B be the vertex that was processed in the following iteration. If both $s, t \in G_{v_B} \setminus p(B)$, then (s,t) was added to $\mathcal{M}_B$. Otherwise, exactly one of s, t belongs to $G_{v_B} \setminus p(B)$, while the other belongs to G_r, so (s,t) was added to $\mathcal{M}'_B$. We now obtain a set $\mathcal{P}$ of disjoint paths, routing a subset of vertices in $\mathcal{M}$. We show that $|\mathcal{P}| \geq \Omega(\mathsf{OPT}(G, \mathcal{M})/\log k)$. Let $\mathcal{P}^*$ be the optimal solution to instance $\mathsf{OPT}(G, \mathcal{M})$, and let $\mathcal{M}^*$ be the set of the demand pairs routed by $\mathcal{P}^*$. For every block $B \in \mathcal{B}$, let $\tilde{\mathcal{M}}(B) = \mathcal{M}^* \cap \mathcal{M}_B$, and let $\tilde{\mathcal{M}}'_B = \mathcal{M}^* \cap \mathcal{M}'_B$.

From Observation B.14, set $\mathcal{P}'(B)$ of paths routes at least $\Omega(\mathsf{OPT}(B, \mathcal{N}_B)/\log k) \geq \Omega(\mathsf{OPT}(G, \mathcal{M}_B)/\log k) \geq \Omega(|\tilde{\mathcal{M}}_B|/\log k)$ demand pairs. Therefore, $|\mathcal{P}| = \sum_{B \in \mathcal{B}} |\mathcal{P}'(B)| \geq \sum_{B \in \mathcal{B}} \Omega(|\tilde{\mathcal{M}}_B|/\log k)$. On the other hand, as observed above, for every block $B \in \mathcal{B}$, $|\tilde{\mathcal{M}}'_B| \leq 1$ (since all paths routing the pairs in $\tilde{\mathcal{M}}'_B$ contain vertex $p(B)$), while $|\mathcal{P}'(B)| \geq 1$. Therefore, $|\mathcal{P}| \geq \sum_{B \in \mathcal{B}} |\tilde{\mathcal{M}}'_B|$. Overall, $|\mathcal{P}| \geq \sum_{B \in \mathcal{B}} \Omega((|\tilde{\mathcal{M}}_B| + |\tilde{\mathcal{M}}'_B|)/\log k) = \Omega(|\mathcal{M}^*|/\log k)$.

B.3 Approximation Algorithm for NDP-Cylinder

In this section we prove Theorem 1.1 for NDP-Cylinder. Recall that in the NDP-Cylinder problem, we are given a cylinder Σ, obtained from the sphere, by removing two open discs from it. We denote the boundaries of the two discs by Γ_1 and Γ_2, respectively. We assume that we are given a graph G, drawn on Σ, and a set $\mathcal{M}$ of demand pairs. We denote by S and T the sets of all source and all destination vertices participating in the demand pairs in $\mathcal{M}$. We say that a drawing of G is proper with respect to S and T iff the vertices of S lie on Γ_1, the vertices of T lie on Γ_2, and no other edges or vertices of G intersect Γ_1 or Γ_2. We can assume without loss of generality that we are given a proper drawing of the input graph G on Σ with respect to S and T. We also assume that the graph G is connected, as otherwise we can solve the problem for each connected component of G separately. We assume that we know the value OPT of the optimal solution to instance $(G, \mathcal{M})$, and a demand pair $(s^*, t^*) \in \mathcal{M}$ that is routed by some optimal solution to instance $(G, \mathcal{M})$. We can make these assumptions by solving the problem for every possible value of OPT between 1 and $|\mathcal{M}|$, and every choice of $(s^*, t^*) \in \mathcal{M}$. It is enough to show that the algorithm returns the desired solution when the value OPT and the pair

(s^*, t^*) are guessed correctly. We can also assume that $\mathsf{OPT} > 10$, since otherwise routing a single demand pair gives a desired solution.

We define a set W_1 of points on Γ_1, as follows. First, we add to Γ_1 all points corresponding to the vertices of S. Next, for every segment of $\Gamma_1 \setminus W_1$, we add an arbitrary point on the segment to W_1. We define a set W_2 of points on Γ_2 similarly, using T instead of S. Our first step is to compute the shortest G-normal curve $\gamma^* \subseteq \Sigma$, connecting a point of W_1 to a point of W_2. We consider two cases.

Assume first that the length of γ^* is less than $\mathsf{OPT}/2$. Then we can cut the cylinder Σ along the curve γ^*, deleting from G all vertices lying on γ^*, to obtain a disc Σ', and a drawing of G on Σ', where all terminals of $S \cup T$ lie on the boundary of the disc. It is easy to see that the value of the optimal solution of the resulting problem instance is at least $\mathsf{OPT}/2$. We can now apply the $O(\log k)$-approximation algorithm for NDP-Disc from Section B.2 to obtain an $O(\log k)$-approximate solution to the new NDP-Disc instance, which in turn gives an $O(\log k)$-approximate solution to the original instance of NDP-Cylinder.

We assume from now on that the length of γ^* is at least $\mathsf{OPT}/2$. We reduce the problem to DPSP. Let σ be cycle, whose vertices are W_1, and they are connected in the order of their appearance on Γ_1. We delete the edge of σ incident on s^*, that appears after s^* in the counter-clock-wise traversal of Γ_1, and direct all edges of the resulting path away from s^*. We define a path σ' similarly - start with the cycle, whose vertices are W_2, and they are connected in the order of their appearance on Γ_2. Delete the edge incident on t^*, that appears after t^* in the counter-clock-wise traversal of Γ_2, and direct all edges of the resulting path away from t^*. Our next step is to define a set $\mathcal{K}$ of constraints for the DPSP problem instance. The instance we construct will only contain type-1 and type-2 constraints.

Let a^* be the last vertex of σ. The first constraint that we add to $\mathcal{K}$ is $(1, s^*, a^*, \mathsf{OPT}/2)$. This constraint ensures that overall we will not attempt to route more than $\mathsf{OPT}/2$ demand pairs.

Consider now any pair $x, y \in W_1$ of points. Let $\beta_1(x, y)$ and $\beta_2(x, y)$ be the two segments of Γ_1 whose endpoints are x and y. For $i \in \{1, 2\}$, we let $\ell_i(x, y)$ be the smallest number of vertices that need to be removed from G, in order to disconnect all vertices of $\beta_i(x, y) \cap S$ from the vertices of T - this value can be computed efficiently using standard minimum cut algorithms. Let $w_i = \ell_i(x, y)$. We assume w.l.o.g. that x lies before y on σ. If s^* is not an inner vertex on $\beta_i(x, y)$, then we add the constraint $(1, x, y, w_i)$ to $\mathcal{K}$. Otherwise, we add two constraints: $(1, s^*, x, w_i)$ and $(1, y, a^*, w_i)$ to $\mathcal{K}$. For every pair of points $(x, y) \in W_1$, we therefore add at most three type-1 constraints to $\mathcal{K}$.

We process all pairs of points $(x, y) \in W_2$, and add corresponding type-2 constraints to $\mathcal{K}$ similarly, except that we use t^* instead of s^*. This finishes the description of the DPSP instance. We start with the following easy observation.

Observation B.15 *Let $\mathcal{P}^*$ be the optimal solution to the* NDP *instance $(G, \mathcal{M})$, such that (s^*, t^*) is routed by $\mathcal{P}^*$, and let $\mathcal{M}^*$ be the set of the demand pairs routed by $\mathcal{P}^*$. Let $\mathcal{M}^{**} \subseteq \mathcal{M}^*$ be any subset of $\lfloor |\mathcal{M}^*|/2 \rfloor$ demand pairs. Then $\mathcal{M}^{**}$ is a feasible solution to the* DPSP *instance $(\sigma, \sigma', \mathcal{M}, \mathcal{K})$.*

Proof: Since we assume that the demand pair (s^*, t^*) is routed by $\mathcal{P}^*$, and since the demand pairs in $\mathcal{M}^*$ must be non-crossing with respect to Γ_1 and Γ_2, due to the way in which we have defined the paths σ and σ', set $\mathcal{M}^{**}$ must be non-crossing with respect to σ and σ'.

Recall that we have added the constraint $(1, s^*, a^*, |\mathsf{OPT}|/2)$ to $\mathcal{K}$, where s^* and a^* are the endpoints of σ. Since $|\mathcal{M}^{**}| \leq |\mathcal{M}^*|/2 = |\mathsf{OPT}|/2$, set $\mathcal{M}^{**}$ satisfies this constraint.

Consider now any pair (x, y) of points in W_1, and fix some $i \in \{1, 2\}$. Since set $\mathcal{M}^*$ of demand pairs is routable in G, the number of the source vertices of the demand pairs in $\mathcal{M}^*$ that lie on $\beta_i(x, y)$ is at most ℓ_i, as the value of the minimum cut separating the vertices of $S \cap \beta_i(x, y)$ from the vertices of T is ℓ_i. It is now easy to verify that all type-1 constraints in $\mathcal{K}$ corresponding to the pair (x, y) are satisfied by $\mathcal{M}^*$, and hence by $\mathcal{M}^{**}$. Type-2 constraints are dealt with similarly. □

Our next step is to use the algorithm from Theorem 3.1, in order to compute a set $\mathcal{M}' \subseteq \mathcal{M}$ of non-crossing (with respect to σ and σ') demand pairs, satisfying all constraints in $\mathcal{K}$, with $|\mathcal{M}'| \geq \Omega(\mathsf{OPT}(G, \mathcal{M})/\log k)$. We assume that $\mathcal{M}' = \{(s_1, t_1), \ldots, (s_r, t_r)\}$, where $s_1, \ldots, s_r$ appear in this circular order on Γ_1, and if $(s^*, t^*) \in \mathcal{M}'$, then $s_1 = s^*$. If $|\mathcal{M}'| \leq 10$, then a routing of a single demand pair gives a feasible solution to the NDP problem instance and achieves the desired approximation ratio. Therefore, we assume that $|\mathcal{M}'| > 10$. We let $\mathcal{M}''$ contain all demand pairs (s_j, t_j), where $j = 0$ modulo 8, and $1 \leq j \leq r$. Notice that $\mathcal{M}''$ excludes the pair (s^*, t^*). Let S'' and T'' be the sets of the source and the destination vertices of the demand pairs in $\mathcal{M}''$. We need the following theorem.

Theorem B.16 *There is a set $\mathcal{P}$ of $|S''|$ node-disjoint paths in G, connecting the vertices of S'' to the vertices of T''.*

We prove Theorem B.16 below, after we complete the proof of Theorem 1.1 using it. Denote $|\mathcal{M}''| = \kappa^*$, and recall that from our constraints, $\kappa^* \leq |\mathsf{OPT}|/4$. Our first step is to construct a collection $\mathcal{Z} = (Z_1, \ldots, Z_{\kappa^*})$ of κ^* tight concentric cycles around Γ_1, where we consider a planar drawing of G, whose outer face contains Γ_2. In order to do so, we denote $\Gamma_1 = Z_0$, and perform κ^* iteration, where in the ith iteration we construct the cycle Z_i. In order to execute the ith iteration, for $1 \leq i \leq \kappa^*$, we contract $D(Z_{i-1})$ into a single vertex s, to obtain a new graph H_i. We view the face of H_i containing Γ_2 as the outer face in the planar drawing of H_i, and we then let $Z_i = \text{min-cycle}(H, s)$. Since the length of the shortest G-normal curve connecting a point of Γ_1 to a point of Γ_2 is at least $|\mathsf{OPT}|/2 > \kappa^*$, it is easy to verify that we can successfully complete the construction of the set $\mathcal{Z}$ of κ^* cycles, so that all cycles are disjoint from the vertices lying on Γ_2.

Our next step is to re-route the paths in $\mathcal{P}$, so that they become monotone with respect to $\mathcal{Z}$. In order to do so, we construct a graph H, as follows. We start with the union of the paths in $\mathcal{P}$ and the cycles in $\mathcal{Z}$. We then add a new cycle Y connecting the vertices of T'' in the order in which they appear on Γ_2. We can now use Theorem 2.5 to find a collection $\mathcal{P}'$ of κ^* node-disjoint paths in H, connecting vertices of S'' to vertices of T'', that are monotone with respect to $\mathcal{Z}$. It is easy to see that the paths of $\mathcal{P}'$ are contained in G.

We assume that $\mathcal{P}' = \{P_1, P_2, \ldots, P_{\kappa^*}\}$, and for each $1 \leq i \leq \kappa^*$, we denote by $a_i \in S''$ and $b_i \in T''$ the endpoints of P_i. We assume that $a_1, a_2, \ldots, a_{\kappa^*}$ appear in this circular order on Γ_1. Consider the source vertex $a_1 \in S''$, and let $(a_1, b_{1+z}) \in \mathcal{M}''$ be the demand pair in which a_1 participates. We can assume without loss of generality that $z \leq \kappa^*/2$, since if this is not the case, we can re-order the vertices $a_1, \ldots, a_{\kappa^*}$ in the opposite direction around Γ_1. Observe that for all $1 \leq j \leq \lfloor \kappa^*/2 \rfloor$, pair $(a_j, b_{j+z}) \in \mathcal{M}''$, since the demand pairs in $\mathcal{M}''$ are non-crossing. We now show how to route all demand pairs in $\{(a_j, b_{j+z})\}_{1 \leq z \leq \lfloor \kappa^*/2 \rfloor}$. Fix some $1 \leq j \leq \lfloor \kappa^*/2 \rfloor$. We view the paths in $\mathcal{P}'$ as directed from S'' to T''. Let P'_j be the sub-path of P_j from a_j to the first vertex v_j of P_j lying on Z_{κ^*-j+1}. Let P''_{j+z} be the sub-path of P_{j+z}, from the last vertex v'_{j+z} of P_{j+z} lying on Z_{κ^*-j+1} to b_{j+z}. Finally, let Q_j be the segment of Z_{κ^*-j+1} between v_j to v'_{j+z}, that intersects the paths $P_j, P_{j+1}, \ldots, P_{j+z}$, but no other paths of $\mathcal{P}'$. By combining P'_j, P''_{j+z} and Q_j, we obtain a path P^*_j, connecting a_j to b_{j+z}. We then set $\mathcal{P}^* = \left\{P^*_j \mid 1 \leq j \leq z\right\}$. It is immediate to verify that the paths in $\mathcal{P}^*$ are node-disjoint. Therefore, we obtain a solution routing $\Omega(\mathsf{OPT}(G, \mathcal{M}))$ demand pairs in $\mathcal{M}$. It now only remains to prove Theorem B.16.

Proof of Theorem B.16. Assume for contradiction that there is no such set of paths. Denote $|S''| = \kappa$ and recall that $\kappa < \mathsf{OPT}/2$. Then there is a set Y of at most $\kappa - 1$ vertices, so that $G \setminus Y$ contains no path connecting a vertex of S to a vertex of T. Consider the drawing of G on the sphere Σ'', obtained from the drawing of G on the cylinder Σ, by adding back the two caps with the boundaries Γ_1 and Γ_2. We can then construct a simple closed G-normal curve γ of length at most $\kappa - 1$ in Σ'', so that all vertices of S'' lie in one of the discs of Σ'' with boundary γ, and all vertices of T'' lie on the other disc (but the vertices of S'' and T'' may lie on γ). Notice that γ has to cross Γ_1 or Γ_2. Indeed, otherwise, since there are OPT node-disjoint paths connecting the vertices of Γ_1 to the vertices of Γ_2, all such paths would have to cross γ, and so the length of γ should be at least $\mathsf{OPT} > \kappa$, a contradiction. Moreover, since the length of the shortest G-normal curve connecting a point of Γ_1 to a point of Γ_2 is at least $\mathsf{OPT}/2 > \kappa$, curve γ may not intersect both Γ_1 and Γ_2. We assume without loss of generality that γ crosses Γ_1, and not Γ_2.

Let $\mathcal{R}$ be the set of segments of γ, obtained by deleting all points of γ that lie outside the cylinder Σ (that is, the points that lie in the interior of the cap whose boundary is Γ_1). All curves in $\mathcal{R}$ are mutually disjoint. For each curve $\gamma' \in \mathcal{R}$, let $S(\gamma') \subseteq S''$ be the set of the source vertices that γ' separates from Γ_2 in the cylinder Σ. Then $\bigcup_{\gamma' \in \mathcal{R}} S(\gamma') = S''$ must hold, and so there must be some curve $\gamma^* \in \mathcal{R}$, such that the length of γ^* is less than $|S(\gamma^*)|$. Let ℓ^* denote the length of γ^*.

Let x', y' be the endpoints of the curve γ^*, so $x', y' \in \Gamma_1$. If $x' \in W_1$, then we let $x = x'$. Otherwise, we let x be the closest to x' point of $W_1 \setminus S''$ on Γ_1. We define point y' for y similarly. Consider the two segments $\beta_1(x, y)$ and $\beta_2(x, y)$ of Γ_1, whose endpoints are x and y. One of the segments, say $\beta_1(x, y)$ must contain all points of $S(\gamma^*)$. Since the vertices lying on γ^* separate all vertices of $S(\gamma^*)$ from the vertices of T, $\ell_1(x, y) \le \ell^*$. Assume without loss of generality that x lies before y on σ.

Assume first that s^* does not lie on $\beta_1(x, y)$, and consider the corresponding constraint $K = (1, x, y, w_1) \in \mathcal{K}$. As observed above, $w_1 \le \ell^*$. Due to the way we have selected the subset $\mathcal{M}'' \subseteq \mathcal{M}'$ of the demand pairs, we are guaranteed that $|S(\gamma^*)| \le w_1/2 \le \ell^*$, a contradiction.

Assume now that s^* lies on $\beta_1(x, y)$. Using the same reasoning as above, $w_1 \le \ell^*$. Let β'_1, β''_1 be the segments of $\beta_1(x, y)$ between x and s^*, and between s^* and y, respectively, where the last segment excludes s^*. Let Δ_1, Δ'_1 be the number of the source vertices lying on β_1 and β''_1 respectively, that participate in the demand pairs in $\mathcal{M}'$. Since the constraints $(1, s^*, x, w_1)$ and $(1, y, a^*, w_1)$ belong to $\mathcal{K}$, $\Delta_1 + \Delta'_1 \le 2\ell^*$. Due to the way we have selected the subset $\mathcal{M}'' \subseteq \mathcal{M}'$ of the demand pairs, we are guaranteed that $|S(\gamma^*)| \le \max\{w_1, 1\} \le \ell^*$, a contradiction. □

C Proofs Omitted from Section 4

C.1 Proof of Theorem 4.2

Let $\tau = \frac{w^*}{512 \cdot \log k}$. The algorithm is iterative and maintains a set U of vertices. We start with $U = \emptyset$, and in every iteration we add vertices to U. The algorithm terminates when no vertices have been added to U in an iteration. Each iteration is executed as follows:

Let $\mathcal{H}$ denote the set of all connected components of $G \setminus U$, and for each $H \in \mathcal{H}$, let $\mathcal{T}_H \subseteq \mathcal{T}(\mathcal{M})$ denote the set of all terminals contained in $V(H)$. For each $H \in \mathcal{H}$ with $|V(H) \cap \mathcal{T}_H| > 3$, we use Observation 2.1 to compute a vertex cut (A, C, B) in H whose sparsity φ with respect to $\mathcal{T}_H$ is within a factor α_{AKR} from the value of the sparsest cut, and $C \cap \mathcal{T}_H = \emptyset$. If $\varphi < \tau$, then we add the vertices of C to U. This finishes the description of the iteration execution.

Consider the set $\{G_1, \ldots, G_r\}$ of all components of $G \setminus U$ once the algorithm terminates. For all $1 \le j \le r$, let $\mathcal{M}^j = \{(s_i, t_i) \in \mathcal{M} \mid s_i, t_i \in V(G_j)\}$. The algorithm output is $\left\{(G_j, \mathcal{M}^j)\right\}_{j=1}^r$.

Clearly, the algorithm is efficient and terminates after at most $n + 1$ iterations, since the size of U

increases after each iteration, except for the last one. It is easy to see that no edge connects a vertex of G_j to a vertex of $G_{j'}$ for any $1 \le j \ne j' \le r$, from the definition of the graphs G_j.

We now verify that for all $1 \le j \le r$, $(G_j, \mathcal{M}^j)$ is a well-linked instance. Fix some $1 \le j \le r$, and let $\mathcal{T}', \mathcal{T}'' \subseteq \mathcal{T}(\mathcal{M}^j)$ be two disjoint equal-sized subsets of $\mathcal{T}(\mathcal{M}^j)$. Assume for contradiction that there are fewer than $\alpha_{\mathrm{WL}} \cdot |\mathcal{T}'|$ node-disjoint paths in $\mathcal{M}^j$ connecting the vertices of $\mathcal{T}'$ to the vertices of $\mathcal{T}''$. Then by Menger's Theorem, there exists a set $Z \subseteq V(G_j)$ of fewer than $\alpha_{\mathrm{WL}} \cdot |\mathcal{T}'|$ vertices in G^j, such that there is no path from $\mathcal{T}'$ to $\mathcal{T}''$ in $G_j \setminus Z$. Note that we may assume that $Z \cap \mathcal{T}(\mathcal{M}^j) = \emptyset$. Otherwise, since the terminals have degree 1 and form an independent set in G, we may simply replace each terminal in Z with its unique neighbor. Consider a vertex cut (A', C', B') of G_j, defined as follows: $C' = Z$, A' is the union of the vertices of all components of $G_j \setminus Z$ intersecting $\mathcal{T}'$, and $B' = V(G_j) \setminus (A' \cup C')$. This is a valid vertex cut, with $\mathcal{T}' \subseteq A'$ and $\mathcal{T}'' \subseteq B'$. The sparsity of cut (A', C', B') with respect to $\mathcal{T}(\mathcal{M}^j)$ is at most $\frac{|Z|}{\min\{|\mathcal{T}_H \cap A'|, |\mathcal{T}_H \cap B'|\}} < \frac{\alpha_{\mathrm{WL}}|\mathcal{T}'|}{\min\{|\mathcal{T}'|, |\mathcal{T}''|\}} = \alpha_{\mathrm{WL}}$. Therefore, the algorithm from Observation 2.1 should have returned a cut of sparsity less than $\alpha_{\mathrm{AKR}} \cdot \alpha_{\mathrm{WL}} = \tau$, a contradiction.

We now show that $|U| = \left|V(G) \setminus \left(\bigcup_{j=1}^r V(G_j)\right)\right| \le \frac{w^* \cdot |\mathcal{M}|}{64}$. Consider a single iteration of the algorithm. Let H be any component of $G \setminus U$, and let $\mathcal{T}_H$ be the set of all terminals contained in $V(H)$. Suppose the algorithm computes a vertex cut (A, C, B) in H with respect to $\mathcal{T}_H$ of sparsity $\varphi < \tau$, and adds the vertices of C to U. Assume without loss of generality that $|A \cap \mathcal{T}_H| \le |B \cap \mathcal{T}_H|$. Since we have assumed that $C \cap \mathcal{T}_H = \emptyset$, the sparsity of the cut $\varphi = \frac{|C|}{|A \cap \mathcal{T}_H|}$. Moreover, since $\varphi < \tau$, $|C| < \tau \cdot |A \cap \mathcal{T}_H|$ must hold. We charge the value of τ to every terminal in $A \cap \mathcal{T}_H$, so that the total amount charged to the terminals of $A \cap \mathcal{T}_H$ is $\tau \cdot |A \cap \mathcal{T}_H| \ge |C|$. This charging scheme is repeated whenever a set of vertices is added to U throughout the different iterations and components considered by the algorithm. Note that a terminal may be charged during multiple iterations, but at most once per iteration. Clearly, the sum of the total charges to all of the terminals is at least $|U|$. Also, each terminal $t \in \mathcal{T}(\mathcal{M})$ can be charged at most $\lfloor \log 2k \rfloor$ times, since whenever t is charged and U is updated, the number of terminals in the component containing t in $G \setminus U$ falls by at least a factor 2. Therefore, the total charge to all terminals is at most $\tau \cdot |\mathcal{T}(\mathcal{M})| \cdot \log 2k \le (\frac{w^*}{512 \cdot \log k}) \cdot (\log 2k) \cdot |\mathcal{T}(\mathcal{M})| \le \frac{w^* \cdot |\mathcal{M}|}{64}$, and $|U|$ is also bounded by this amount.

Finally, we verify that $\sum_{j=1}^r |\mathcal{M}^j| \ge 63|\mathcal{M}|/64$. Recall that for each demand pair $(s_i, t_i) \in \mathcal{M}$, the current LP-solution sends w^* flow units between s_i and t_i. Let $\tilde{\mathcal{M}} = \mathcal{M} \setminus \left(\bigcup_{j=1}^r \mathcal{M}^j\right)$. If $(s_i, t_i) \in \tilde{\mathcal{M}}$, then all of the w^* flow units between s_i and t_i must pass through U. Since $|U| \le \frac{w^* \cdot |\mathcal{M}|}{64}$, $|\tilde{\mathcal{M}}| \le \frac{|\mathcal{M}|}{64}$, and the theorem follows.

C.2 Proof of Lemma 4.4

We use the notion of treewidth, which is usually defined via tree decompositions. A tree decomposition of a graph H consists of a tree τ and a collection $\{\beta_v \subseteq V(H)\}_{v \in V(\tau)}$ of vertex subsets, called bags, such that the following two properties are satisfied: (i) for each edge $(a, b) \in E(H)$, there is some node $v \in V(\tau)$ with both $a, b \in \beta_v$ and (ii) for each vertex $a \in V$, the set of all nodes of τ whose bags contain a form a non-empty connected subtree of τ. The *width* of a given tree decomposition is $\max_{v \in V(\tau)} \{|\beta_v|\} - 1$, and the treewidth of a graph H, denoted by $\mathrm{tw}(H)$, is the width of a minimum-width tree decomposition for H.

Claim C.1 *For each* $1 \le j \le r$*,* $\mathrm{tw}(G_j) \ge \frac{W_j}{2^{12} \cdot \alpha_{\mathrm{AKR}} \cdot \log k}$.

We first prove the lemma assuming Claim C.1. It is well known that any planar graph of large treewidth contains a large grid as a minor [RST94, DH05]. We use the following theorem.

Theorem C.2 (Theorem 1.2 in [DH05]) *For any fixed graph H, every H-minor-free graph of treewidth w has an $(\Omega(w) \times \Omega(w))$ grid as a minor.*

Therefore, in particular, every planar graph of treewidth w contains an $(\Omega(w) \times \Omega(w))$ grid as a minor. So G_j must contain a grid minor of size $(\Omega(W_j/\log k) \times \Omega(W_j/\log k))$. Since all terminals have degree 1 in G, the number of the non-terminal vertices, $N_j \geq \Omega(W_j^2/\log^2 k)$. It now remains to prove Claim C.1.

Proof of Claim C.1. For convenience, we let $\kappa = |\mathcal{T}(\mathcal{M}^j)|$. Assume for contradiction that $\text{tw}(G_j) < \frac{W_j}{2^{12}\cdot\alpha_{\text{AKR}}\cdot\log k}$ and consider a tree decomposition τ of width less than $\frac{W_j}{2^{12}\cdot\alpha_{\text{AKR}}\cdot\log k}$. Note that τ cannot be a singleton vertex, as $\kappa > \frac{W_j}{2^{12}\cdot\alpha_{\text{AKR}}\cdot\log k}$, since $W_j = w^* \cdot \kappa/2$ and $w^* < 1$. For any given subtree τ' of τ, we let $\beta(\tau') = \bigcup_{u\in V(\tau')} \beta_u$. We say that a vertex $v \in V(\tau)$ is good iff every component of $G \setminus \beta_v$ contains at most $\kappa/2$ terminals.

Claim C.3 *There is a good vertex in τ.*

Proof: Note that for a vertex $v \in V(\tau)$, if $\tau_1, \ldots, \tau_\ell$ are the connected subgraphs of $\tau \setminus \{v\}$, then every connected component C of $G_j \setminus \beta_v$ must have $V(C) \subseteq \beta(\tau_p) \setminus \beta_v$ for some $1 \leq p \leq \ell$. Also note that the sets $\{\beta(\tau_1) \setminus \beta_v, \ldots, \beta(\tau_\ell) \setminus \beta_v\}$ are pairwise vertex disjoint.

Root the tree τ at any vertex v_0, and start with $v = v_0$. While the current vertex v has a child v_i, such that the sub-tree τ_i of τ rooted at v_i has $|(\beta(\tau_i) \setminus \beta_v) \cap \mathcal{T}(\mathcal{M}^j)| > |\mathcal{T}(\mathcal{M}^j)|/2$, we set $v = v_i$, and continue to the next iteration. It is immediate to verify that when the algorithm terminates, the final vertex v is good. □

Let $v \in V(\tau)$ be a good vertex, and let $C_1, \ldots, C_a$ denote the connected components of $G_j \setminus \beta_v$. For all $1 \leq p \leq a$, let $\kappa_p = |\mathcal{T}(\mathcal{M}^j) \cap C_p|$. Note that $|\beta_v| \leq \frac{W_j}{2^{12}\cdot\alpha_{\text{AKR}}\cdot\log k} = \frac{w^*\cdot\kappa}{2^{13}\cdot\alpha_{\text{AKR}}\cdot\log k} \leq \kappa/4$, and hence $|\mathcal{T}(\mathcal{M}^j) \setminus \beta_v| = \sum_{p=1}^{a} \kappa_p \geq 3\kappa/4$. We claim that there exist two disjoint subsets $\mathcal{T}', \mathcal{T}'' \subseteq \mathcal{T}(\mathcal{M}^j) \setminus \beta_v$ such that $|\mathcal{T}'| = |\mathcal{T}''| = \lceil \kappa/8 \rceil$, while $\mathcal{T}'$ and $\mathcal{T}''$ are separated by β_v in G_j. Let $1 \leq b < a$ be the smallest index for which $\sum_{p=1}^{b} \kappa_p \geq \kappa/8$. Then $\sum_{p=1}^{b} \kappa_p \leq (1/8 + 1/2)\kappa = 5\kappa/8$, and so $\sum_{p=b+1}^{a} \kappa_p \geq (3/4 - 5/8)\kappa = \kappa/8$. We then let $\mathcal{T}' \subseteq \left(\bigcup_{p=1}^{b} C_p \cap \mathcal{T}(\mathcal{M}^j)\right)$ and $\mathcal{T}'' \subseteq \left(\bigcup_{p=b+1}^{a} C_p \cap \mathcal{T}(\mathcal{M}^j)\right)$, respectively, be subsets of size $\lceil \kappa/8 \rceil$, so $\mathcal{T}' \cap \mathcal{T}'' = \emptyset$, and β_v separates $\mathcal{T}'$ from $\mathcal{T}''$ in G_j.

Since the terminals are α_{WL}-well-linked in G_j, there is a set of at least $\alpha_{\text{WL}} \cdot |\mathcal{T}'| = \frac{w^*\cdot|\mathcal{T}'|}{512\cdot\alpha_{\text{AKR}}\cdot\log k} \geq \frac{w^*\cdot\kappa}{2^{12}\cdot\alpha_{\text{AKR}}\cdot\log k} = \frac{W_j}{2^{12}\cdot\alpha_{\text{AKR}}\cdot\log k}$ node-disjoint paths from $\mathcal{T}'$ to $\mathcal{T}''$ in G_j. Since $\mathcal{T}'$ and $\mathcal{T}''$ are separated by β_v in $G \setminus \beta_v$, each path must intersect at least one distinct vertex of β_v. However, we have assumed that $|\beta_v| < \frac{W_j}{2^{12}\cdot\alpha_{\text{AKR}}\cdot\log k}$, a contradiction. □

D Table of Parameters

α_{WL}	$\frac{w^*}{512 \cdot \alpha_{\text{AKR}} \cdot \log k} = \Theta(w^*/\log k)$	well-linkedness parameter, where k is the number of the demand pairs in the original instance.
Δ	$\lceil W^{2/19} \rceil$	Minimum distance between terminals in distinct terminal sets of $\mathcal{X}$.
Δ_0	$\Theta(\Delta \log n)$	Maximum distance between terminals in each terminal set of $\mathcal{X}$.
τ	$W^{18/19}$	Threshold for light and heavy clusters in $\mathcal{X}$
Δ_1	$\lfloor \Delta/6 \rfloor$	Depth of shells in Case 1
Δ_2	$\lfloor \Delta_1/3 \rfloor$	Depth of inner shells in Case 1