Decision Algorithms for Unsplittable Flow and the Half-Disjoint Paths Problem

Jon M. Kleinberg*

Abstract

We consider the *bounded unsplittable flow problem*: given terminal pairs in a network, with associated real-valued demands in the range $[0, \frac{1}{2}]$, find a single flow path for each pair so that no more than 1 unit of demand is routed through any vertex. Thus, the setting is not directly comparable to that of the classical *disjoint paths problem* (when all demands are equal to 1) — we must deal with connections having varied, real-valued amounts of demand, but we impose the *boundedness restriction* that each connection can consume at most half the capacity of any vertex.

Our main result is a polynomial-time algorithm for the bounded unsplittable flow problem, in an arbitrary graph, when the number of terminal pairs is a fixed constant. Our algorithm is conceptually much simpler than Robertson and Seymour's corresponding algorithm for the disjoint paths problem with a constant number of terminal pairs; and we can decide the routability of a non-trivially super-constant number of terminal pairs (up to $\Omega((\log\log n)^{2/15})$) in polynomial time. We also obtain polynomial-time algorithms for several natural optimization problems derived from the bounded unsplittable flow problem, when the number of terminal pairs is sufficiently small, and algorithms with better bounds for the case of planar graphs. The results all carry over to problems involving edge capacities. Our approach makes use of several of the ideas underlying the Robertson-Seymour algorithm, together with some new algorithmic components.

The results add to a growing body of work suggesting that versions of our *boundedness restriction* — while often relatively mild from the point of view of the underlying motivation — can have very interesting qualitative effects on the tractability of basic routing problems.

1 Introduction

Disjoint Paths and Unsplittable Flow. In the *vertex-disjoint paths problem* we are given a graph G, and a set of k pairs of vertices in G (the *terminals*), and we must decide whether there exist mutually vertex-disjoint paths connecting the given pairs of terminals. This is a central problem in combinatorial optimization and algorithmic graph theory (see e.g. the surveys [3, 10, 15]), and is one of Karp's original NP-complete problems [5]. It has attracted interest in the contexts of transportation networks, VLSI layout, and, recently, virtual circuit routing in high-speed networks; it is an underlying component, and serves as a common theme, in the range of problems that are formulated and studied in these settings.

Motivated by networks carrying heterogeneous traffic with vastly different bandwidth requirements, one can define the *unsplittable flow problem* (UFP), which generalizes the disjoint paths problem. In one basic version of the UFP, we are given a graph G, a set of k pairs of vertices in G

*Department of Computer Science, Cornell University, Ithaca NY 14853. Email: kleinber@cs.cornell.edu. Supported in part by an Alfred P. Sloan Research Fellowship and by NSF Faculty Early Career Development Award CCR-9701399.

(the *terminals*), and a nonnegative *demand* $\rho_i \leq 1$ associated with each given pair of terminals. We now must decide whether it is possible to choose a single path for each pair of terminals so that the cumulative demand sent by these paths through any vertex is at most 1; if this is possible, we say that the set of terminal pairs is *realizable*. Thus, the disjoint paths problem is simply the special case in which all demands ρ_i are equal to 1. Versions of this model were studied by Schrijver, Seymour, and Winkler [16] in the context of SONET ring routing, by the author in the *single-source* case [6], and more recently in work of Srinivasan [17] and Kolliopoulos and Stein [8].

The special case of the UFP in which all demands are at most $\frac{1}{2}$ — we will call this the *bounded unsplittable flow problem* — often exhibits qualitatively different behavior from the general UFP. It remains NP-complete even in the natural special case of the *half-disjoint paths problem*, when all demands are equal to $\frac{1}{2}$ (by a simple reduction from the disjoint paths problem), but there are several contexts in which it appears to be algorithmically more "manageable." For example, in work of the author and Éva Tardos on the UFP (with edge capacities) in a family of graphs generalizing the $n \times n$ grid [7], the natural multicommodity flow relaxation of the problem provides very strong information for designing approximation algorithms when all demands are at most $\frac{1}{2}$, but the relaxation is known to be extremely weak when the demands are as large as 1 (see e.g. [4]). Thus, by imposing a fairly mild restriction — that each connection consume at most half the capacity of each vertex — one can effect fairly dramatic changes on the "global" structure of the routing problem. This has similarities with, and often technical connections to, a large set of *half-integrality results* for the edge-disjoint paths problem: good characterizations are known for the multicommodity flow problem with half-integer flow values in a large number of special cases for which the corresponding edge-disjoint paths problem is intractable [3]; in many such special cases, allowing half-integrality also greatly reduces the computational complexity of the problem [9].

The Robertson-Seymour Algorithm. One of the major results in the area of graph algorithms is Robertson and Seymour's polynomial-time algorithm for the disjoint paths problem when the number of terminal pairs is a fixed constant [10, 12]. The Robertson-Seymour algorithm (henceforth, the *R-S algorithm*) is quite involved, and the new techniques underlying it have found applications in a wide variety of algorithmic problems. Two aspects of its complexity are of interest to us here. First, the proof of its correctness requires the full power of the machinery from the *Graph Minors* series of papers of Robertson and Seymour, and thus runs to several hundred pages [10]. Second, its running time has the form $O(f(k) \cdot n^3)$, where f is an extremely rapidly growing function; thus, in an n-vertex graph, it is in fact possible to solve the disjoint paths problem for $g(n)$ terminal pairs in polynomial time, where g is an extremely slowly growing function. (I.e. g is chosen so that $f(g(n)) = n^{O(1)}$.) It is a non-trivial task even to determine the function g explicitly (see e.g. the discussion at the conclusion of [12]), for that too requires tracing its (implicit) construction through much of the Graph Minors series.

The present work. We begin by considering the bounded unsplittable flow problem in an arbitrary graph, when the number of terminal pairs is a fixed constant. Thus, our setting is not directly comparable to that of the R-S algorithm. We must deal with terminal pairs that have varied, real-valued amounts of demand associated with them. However, we can make use of the fact that all demands are at most $\frac{1}{2}$; in particular, we are not required to solve instances of the disjoint paths problem. (Note, however, that our setting does contain the *half-disjoint paths problem*.)

Our first result is a polynomial-time algorithm for this problem. The algorithm has several appealing features in the context of the discussion above. First, its description and proof of correctness are much simpler than those of the R-S algorithm — specifically, we are able to provide them in full in the present paper, with the exception of a single result from [11, 13], that a graph with an excluded grid minor has bounded tree-width (more on this below). Second, our algorithm can also handle a super-constant number of terminal pairs in polynomial time — for an n-vertex graph, we can efficiently solve an instance of the bounded UFP with $h(n)$ terminal pairs, for a function h. It turns out that it is not difficult to determine the function h explicitly in our case, and it is also easy to show (using e.g. [12]) that the number $h(n)$ is significantly larger than the corresponding number $g(n)$ associated with the R-S algorithm.

Theorem 1.1 *There is a polynomial-time algorithm to decide the feasibility of an instance of the bounded unsplittable flow problem on an arbitrary n-vertex graph with $k \leq \frac{1}{40}(\log\log n)^{2/15}$ terminal pairs.*

When the input graph is planar, we can give a better bound.

Theorem 1.2 *There is a polynomial-time algorithm to decide the feasibility of an instance of the bounded unsplittable flow problem on an n-vertex planar graph with $k \leq O\left(\left(\frac{\log n}{\log\log n}\right)^{1/3}\right)$ terminal pairs.*

Thus, our bound on k for planar graphs is within a polynomial function of the threshold at which one could well expect the bounded UFP to become intractable simply due to the fact that it contains the NP-complete *partition problem.* (The *partition problem* asks, for a set of k numbers, whether they can be partitioned into two sets whose sums are the same.)

Several observations can be made at this point. First, our main results tie in closely with the earlier discussion about the effect of restricting all demands to be at most $\frac{1}{2}$: they suggest that the unsplittable flow problem with a small number of terminal pairs is a setting in which this restriction makes the routing problem significantly more tractable, both conceptually and quantitatively — in terms of the number of terminal pairs that can be efficiently routed.

The algorithms presented here apply also to the version of the unsplittable flow problem in which capacities are placed on the edges rather than the vertices. Indeed, we can reduce an edge-capacitated problem to vertex-capacitated problem by subdividing each edge once and replacing each vertex with a sufficiently large grid. The details are straightforward, and we do not go into them further here; we note that the reduction transforms planar graphs into planar graphs, and so Theorem 1.2 carries over as well.[1]

Using calls to our decision algorithm, we also obtain the first polynomial-time algorithms for the following standard optimization problems derived from the bounded unsplittable flow problem, when k is in the range specified by Theorems 1.1 and 1.2.

(i) **Maximum Routable Demand**. Find a realizable subset of the terminal pairs of maximum total demand.

(ii) **Minimum Capacitated Congestion**. Find the minimum $c \geq 1$ such that there is a collection of paths that connects the terminal pairs and sends a demand of at most c through each vertex.

(iii) **Routing in Rounds**. Partition the terminal pairs into as few "rounds of communication" as possible, where the set of terminal pairs assigned to each round must be realizable.

[1]It might be worth noting at this point that for the *edge-disjoint paths problem* with a fixed number of terminal pairs, there is no known polynomial-time algorithm that is simpler than the R-S algorithm. For example, consider that the vertex-disjoint paths problem and the edge-disjoint path problem coincide in the case of 3-regular graphs, for which both problems appear to be nearly as difficult as they are on arbitrary graphs.

Finally we mention a result of Schrijver [14] that provides a method, different from the R-S algorithm, to solve the disjoint paths problem in *planar graphs* when the number of terminal pairs is a fixed constant k. His algorithm has a number of interesting properties; but it is of less concern to us here, since the bound on its running time contains k in the exponent, and thus it does not seem to provide a solution to the disjoint paths problem (in planar graphs) for any super-constant number of terminal pairs.

Overview of the Algorithms. Our approach makes use of several of the ideas underlying the R-S algorithm, together with some new algorithmic components. Thus, we first sketch the basic structure of the R-S algorithm. An *H-minor* in a graph G is a subgraph of G that can be contracted to a graph isomorphic to H. Precise definitions of certain other terms used in this informal discussion will be given in the following sections. At a high level, the R-S algorithm is based on the following dichotomy: either the input graph G has *bounded tree-width*, or it has a large *grid minor* [11, 13]. In the former case, one can apply dynamic programming to a recursive decomposition of the graph by separators (see also [1]). The latter case again breaks into a dichotomy. If G also has a large *clique minor*, then one can use this as a "crossbar structure," to link up a subset of the terminals in any desired way; if G has a large grid minor but no large clique minor, then it contains a somewhat smaller grid minor that satisfies a complicated set of conditions, based on restricting the way in which edges can "jump" over it. This latter point is by far the most difficult, and it is possible to prove in this case [10] that the middle vertex of such a grid minor can be deleted without affecting the realizability of the terminal pairs.

Our algorithm is based on the same initial dichotomy, that either G has bounded tree-width or a large grid minor; it is for this that we need the result of [11, 13]. The case of bounded tree-width is somewhat more involved in our setting, since paths carrying small amounts of demand can cross a separator, and each other, in complicated ways — but this is still not very difficult; we discuss it in Section 5. The advantage of bounding the maximum demand becomes clear in the other half of the dichotomy, when G contains a large grid minor. Since each demand is at most $\frac{1}{2}$, any vertex of G can carry at least two paths in any routing of the terminal pairs. As a result, it turns out to be possible to use a grid minor directly as a type of "crossbar structure." There are some complications in doing this, since grids are not as easy to use in this context as cliques; in particular one can place terminals on a grid in such a way that it is not possible to link them by paths that respect the vertex capacities. In Sections 3 and 4, we show how a grid minor can be used as a restricted type of crossbar, and the sense in which this is sufficient for our purposes. The effect of this is that we are not forced to deal with the distinction between "flat" grid minors and arbitrary grid minors, as was the case in [12], and this eliminates a large amount of complexity both in the proof and in the overall bounds.

Again, it is worth making some observations. First, the main bounds needed in analyzing the algorithm here are low-order polynomials, with the exception of the exponential bound on the relation between tree-width and grid minors in arbitrary graphs that is provided by [13]. Any improvement in this bound translates directly to a corresponding improvement in the bound on k in Theorem 1.1.

In the remainder of the paper, we use *branch-width* [12] in place of tree-width; and we use subgraphs of G that we call *hexes* in place of grid minors. This follows the approach of [12], and seems to make for simpler proofs; at a conceptual level, the effect of this substitution is not significant.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph on n vertices, and let $\mathcal{T} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ be a set of pairs of vertices of G, which we refer to as *terminal pairs*. Associated with each terminal pair (s_i, t_i) is a non-negative, real-valued *demand* $\rho_i \leq 1$. Let $\Omega_{\mathcal{T}}$ denote $\{s_1, t_1, s_2, t_2, \ldots, s_k, t_k\}$, the set of *terminals*, and let $\mathcal{D}_{\mathcal{T}}$ denote $\{\rho_1, \ldots, \rho_k\}$, the (multi)set of *demand values*. (We omit the subscript $\mathcal{T}$ when it is clear from context.)

The entities above constitute an instance of the *unsplittable flow problem* (UFP). They constitute an instance of the *bounded unsplittable flow problem* if $\rho_i \leq \frac{1}{2}$ for each i. A *routing* of $\mathcal{T}$ is a set of k paths, the i^{th} of which has endpoints s_i and t_i, and carries a demand of ρ_i. A *feasible routing* of $\mathcal{T}$ is a routing such that, for each $v \in V$, the total demand associated with paths containing v is at most 1. We say that $\mathcal{T}$ is *realizable* in G if there is a feasible routing of $\mathcal{T}$.

We say that $\mathcal{T}$ is *initially feasible* if for all $v \in \Omega_{\mathcal{T}}$,

$$\sum_{i: v \in \{s_i, t_i\}} \rho_i \leq 1.$$

If $\mathcal{T}$ is not initially feasible, then it is (for obvious reasons) not realizable.

Finally, a *separation* of G is a pair of sets (A, B) so that $A \cup B = V$ and there are no edges from $A - B$ to $B - A$. The *order* of (A, B) is $|A \cap B|$.

3 Realizability in a Blocked Hex

We first define the notion of a *hex*, the alternative to a grid minor that we use here. One can view a hex essentially as being a subdivision of an $m \times m$ portion of a hexagonal tiling; it can be constructed by having a set of m "parallel" paths all mutually edge-intersect a second set of m "parallel" paths.

Definition 3.1 *We call a graph $\mathcal{H}$ a* hex *(of order m) if it can be written as the union of subgraphs*

$$I_1, I_2, \ldots, I_m, J_1, J_2, \ldots, J_m$$

such that

(i) Each of I_i and I_j is a path.

(ii) For $1 \leq i < j \leq m$, I_i and I_j are vertex-disjoint, and J_i and J_j are vertex-disjoint.

(iii) Each intersection $I_i \cap J_j$ is a path with at least one edge.

(iv) I_i meets the paths J_j in the order $J_1, \ldots, J_m$; J_j meets the paths I_i in the order $I_1, \ldots, I_m$.

We refer to the paths I_i as the rows *of the hex, and the paths J_j as the* columns*. The sub-paths $I_i \cap J_j$ will be called* super-nodes *and the degree-3 vertices of $\mathcal{H}$ the* branch-vertices*.*

Note that the branch-vertices of $\mathcal{H}$ can be naturally paired off: there are two in each super-node. Of the two branch-vertices associated with $I_i \cap J_j$, the one closer to the J_1-end of I_i will be called the *left branch-vertex*, and the other will be called the *right branch-vertex*.

We write I_i^{rs} $(r < s)$ to denote the sub-path of I_i between its first meeting with J_r and its last meeting with J_s. We define J_j^{rs} analogously. A *ring* in $\mathcal{H}$ is a union of paths $I_p^{rs} \cup I_q^{rs} \cup J_r^{pq} \cup J_s^{pq}$, for indices p, q, r, s with $q - p, s - r > 0$; the *order* of the ring is $\min(q - p, s - r)$, and we say the ring is *balanced* if $q - p = s - r$. The ring for which the indices p, q, r, s are $1, m, 1, m$ is the *perimeter* of the hex. A ring C *encloses* a vertex v if every path from v to the perimeter of $\mathcal{H}$ passes through

C. If C is balanced, we say that the *sub-hex* of $\mathcal{H}$ with perimeter C is the subgraph of $\mathcal{H}$ induced on C and the vertices enclosed by C; this latter set constitutes the *interior* of the sub-hex. Some of the terminology for hexes carries over to sub-hexes in the natural way.

Our goal in this section is to show how a hex may be used as a "crossbar" — we wish to give a certain sufficient condition on a set of terminal pairs in a hex that guarantees its realizability. Unfortunately, in the setting for which we need this result, we are not actually provided with a full hex, but rather with a hex from which some vertices have been deleted. Thus, we first define a *blocked hex*, which captures this notion.

Lemma 3.2 *Let $\mathcal{H}$ be a hex of order m, and let (A, B) be a separation of $\mathcal{H}$ of order $q < m$. Then one of A or B contains all but at most q super-nodes from every row and column of $\mathcal{H}$. In particular, one of A or B meets every row and column of $\mathcal{H}$.*

Proof. If the claim of the lemma does not hold, then A contains a set X of $q+1$ vertices from distinct super-nodes from a single row or column of $\mathcal{H}$, and B contains a set Y of $q+1$ vertices from distinct super-nodes from a single row or column of $\mathcal{H}$. But there exist $q+1$ vertex-disjoint paths between any such pair of sets X and Y, which contradicts the assumption that (A, B) is a separation of order q. ■

Given $\mathcal{H}$ and (A, B) as in Lemma 3.2, suppose that B is the set containing all but at most q super-nodes from every row and column of $\mathcal{H}$, and write $\Gamma = A \cap B$. We call the pair $\mathcal{B} = (\mathcal{H}[B], \Gamma)$ a *blocked hex* of order (m, q), with *boundary* Γ. $\mathcal{B}$ inherits the notions of *row* and *column* from $\mathcal{H}$; thus, it has m rows and columns, some of which meet Γ. The *sub-hexes* and *rings* of $\mathcal{B}$ are the subgraphs that constitute sub-hexes and rings in the ambient graph $\mathcal{H}$.

A set of vertices X in $\mathcal{B}$ is *r-linked* ($r \geq |X|$) if, for every ring C in $\mathcal{B}$ of order at least r, there are $|X|$ vertex-disjoint paths from X to C. In such a context, we say that a ring is *large* if it has order at least r.

Now, given any set $\mathcal{T}$ of up to k terminal pairs in a sufficiently large blocked hex $\mathcal{B} = (\mathcal{K}, \Gamma)$, we wish to show that if the terminal set $\Omega_{\mathcal{T}}$ is $4k$-linked, and $\mathcal{T}$ is initially feasible with all demands in $[0, \frac{1}{2}]$, then $\mathcal{T}$ is realizable. To do this, we choose a hex subgraph of $\mathcal{K}$ and locate a large "empty" portion H' of this subgraph, containing no terminals. We then link the terminals to the ring that bounds H', and connect them together in the interior of H'.

This construction is carried out in Theorem 3.5; we begin with two basic lemmas.

Lemma 3.3 *Let $\mathcal{H}$ be a hex of order $\ell + 4$, with ℓ divisible by 4, and let X denote the set of left branch-vertices in the super-nodes $I_i \cap J_1$, for $i = 1, \ldots, \ell$. Let $\mathcal{T}$ be an initially feasible set of terminal pairs for which all demands are in $[0, \frac{1}{2}]$, $|\Omega_{\mathcal{T}}| \leq \ell/4$, and $\Omega_{\mathcal{T}} \subseteq X$. Then $\mathcal{T}$ is realizable in $\mathcal{H}$.*

Proof. Let v_j be the left branch-vertex in the super-node $I_j \cap J_{1+\ell/4}$, and let $Y = \{v_1, v_4, \ldots, v_{3\ell/4-2}\}$. One can easily verify that $\Omega_{\mathcal{T}}$ can be linked to Y by vertex-disjoint paths in the portion of $\mathcal{H}$ between columns 1 and $1 + \ell/4$. Now, let $\mathcal{T}_j$ denote the set of terminal pairs for which at least one endpoint was routed to the vertex $v_{1+3(j-1)}$. Since the total demand of terminal pairs in $\mathcal{T}_j$ is at most 1, we can partition it into three sets $\mathcal{T}_j^1, \mathcal{T}_j^2, \mathcal{T}_j^3$ such that the sum of the demands in each set is at most $\frac{1}{2}$. Define M_j^i ($i = 1, 2, 3$) to be the subgraph of $\mathcal{H}$ consisting of the path from $\Omega_{\mathcal{T}}$ to $v_{1+3(j-1)}$ constructed above, together with the subgraph

$$J_{\ell/4+1}^{1+3(j-1),1+3(j-1)+(i-1)} \cup I_{1+3(j-1)+(i-1)}^{1+\ell/4,\ell} \\ \cup J_{\ell/4+2+3(j-1)+(i-1)}.$$

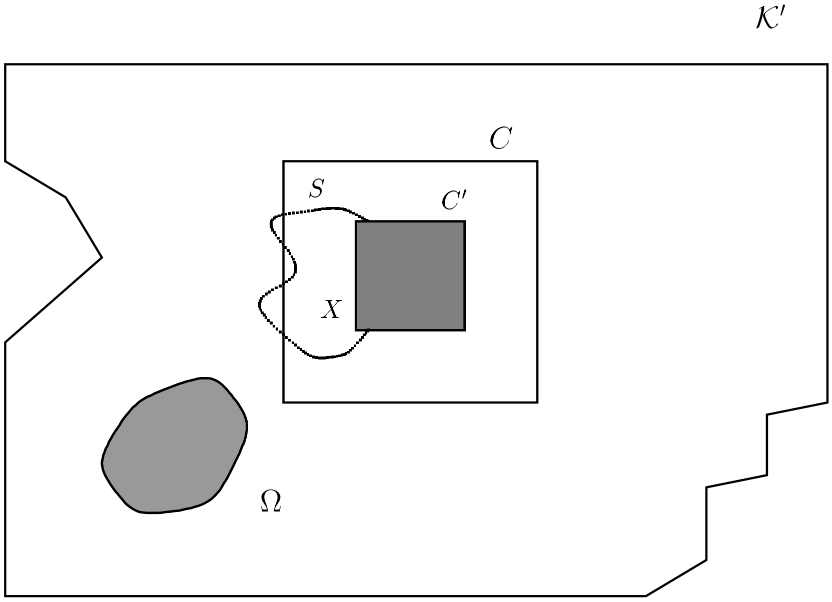

Figure 1: The proof of Theorem 3.5

Note that each M^i_j is connected, and that M^i_j meets M^p_q for all i, j, p, q. Now, if the terminal pair (s_i, t_i) belongs to the sets $\mathcal{T}^i_j$ and $\mathcal{T}^q_p$, we construct an s_i-t_i path lying entirely in $M^i_j \cup M^p_q$. One can verify that, with this choice of paths, no vertex carries more than 1 unit of demand; thus, this constitutes a feasible routing of $\mathcal{T}$ in $\mathcal{H}$. ■

Lemma 3.4 *Let $\mathcal{H}$ be a hex of order $12m$, and let H denote the graph obtained by deleting the interior of the sub-hex of order $8m$ at the center of $\mathcal{H}$. Let C denote the perimeter of the deleted sub-hex, and let C' denote the perimeter of $\mathcal{H}$. Let $S \subseteq V(H)$ have size at most $2m$. Then at most one component of $H - S$ contains both a vertex of C and a vertex of C'.*

Proof. Suppose that components A and $A' \neq A$ both meet C and C'. Then A contains a path P from C to C', and A' contains a path P' from C to C'.

Let $C_0, \ldots, C_{4m}$ be disjoint "concentric" rings in H, with $C = C_0$ and $C' = C_{4m}$. For each $i = 0, \ldots, 4m$, some sub-path of C_i has one end on each of P and P', and is otherwise disjoint from $P \cup P'$. Thus, there exist $4m + 1$ vertex-disjoint paths in H with one end on $P \subseteq A$ and the other on $P' \subseteq A'$; this contradicts the assumption that $|S| \leq 2m$. ■

Theorem 3.5 *Let $\mathcal{B} = (\mathcal{K}, \Gamma)$ be a blocked hex of order (m, q), where $m - q \geq 24k^{3/2}$. Let $\mathcal{T}$ be an initially feasible set of at most k terminal pairs in $\mathcal{K}$ for which all demands are in $[0, \frac{1}{2}]$ and $\Omega_{\mathcal{T}}$ is $4k$-linked. Then $\mathcal{T}$ is realizable in $\mathcal{K}$.*

Proof. We write $\Omega = \Omega_{\mathcal{T}}$, and $\ell = |\Omega|$; note that $\ell \leq 2k$. At least $m - q$ rows and $m - q$ columns of $\mathcal{K}$ contain no vertex of Γ, and these define a hex $\mathcal{H}'$ of order $m - q$ that is a subgraph of $\mathcal{K}$. We divide $\mathcal{H}'$ into $4k$ sub-hexes of order $12k$, with mutually disjoint interiors. Extending the notion of *enclosure* to blocked hexes in the obvious way, we see that each vertex of $\mathcal{K}$ is enclosed by the perimeter of at most one of these sub-hexes. Thus there exists one of the sub-hexes whose perimeter encloses no vertex of Ω; choose such a sub-hex H with perimeter C. Let H' denote the sub-hex of order $8k$ at the center of H, with perimeter C' and interior Y', and let $\mathcal{K}'$ denote the graph obtained from $\mathcal{K}$ by deleting Y'.

Let $X \subseteq C'$ denote the set of left branch-vertices in the leftmost column of H'. We claim that Ω can be linked to X by ℓ vertex-disjoint paths in $\mathcal{K}'$. For suppose not. Then there is a set $S \subseteq V(\mathcal{K}')$, $|S| < \ell$, such that there is no path in $\mathcal{K}' - S$ from Ω to X. First, we observe that there is a component A of $\mathcal{K}' - S$ containing vertices from both Ω and C'; for otherwise Ω would not be $4k$-linked in $\mathcal{K}$. Since C encloses no vertex of Ω, it follows that A also contains a vertex of C. There must also be a component A' of $\mathcal{K}' - S$ containing vertices from both C and X, since $|S| < 2k$ and there are $8k$ vertex-disjoint paths from C to X in $\mathcal{H}' - Y'$ and hence in $\mathcal{K}'$. Since S separates Ω from X, we know that $A \neq A'$; but this gives two distinct components of $H - Y'$, each containing vertices from C and C', which contradicts Lemma 3.4.

Now we describe a feasible routing of $\mathcal{T}$ in $\mathcal{K}$. We first link the set Ω to X by vertex-disjoint paths, using only vertices in $\mathcal{K}'$; this is possible, as argued in the previous paragraph. We then connect the terminal pairs as they appear on X, using vertices in $\mathcal{H}'[H'] \subseteq \mathcal{K}[H']$; this is possible by Lemma 3.3. ■

4 Irrelevant Vertices in a Hex

We use the routing algorithm in the proof of Theorem 3.5 only implicitly, for the purpose of guaranteeing that certain vertices in G are *irrelevant* to the realizability of $\mathcal{T}$. Suppose that G contains a hex subgraph, of order greater than a bound to be determined below. We first locate a minimum set X of vertices separating $\Omega_{\mathcal{T}}$ from a large ring of this hex subgraph, and satisfying some additional conditions. We then show that the portion of the hex that X separates from $\Omega_{\mathcal{T}}$ is in fact a blocked hex, and that X is $4k$-linked to it. An extension of Theorem 3.5 can then be used to show that for certain vertices v in the blocked hex, paths in any feasible routing of $\mathcal{T}$ that pass through X can be re-routed to avoid v. It will follow that v is *irrelevant*, and can be deleted without affecting the realizability of $\mathcal{T}$.

This overall strategy is analogous to one that is used in the analysis of the R-S algorithm [12], for showing how to make use of a large clique minor in a graph.

We begin with a basic fact about separations.

Lemma 4.1 *Let G be a graph, (A, B) a separation of G, and (A', B') a separation of $G[B]$ such that $A \cap B \subseteq A'$. Then $(A \cup A', B')$ is a separation of G, and $(A \cup A') \cap B' \subseteq A' \cap B'$.*

Proof. First note that $A \cup A' \cup B' = V(G)$. Also, if e is an edge of G with one endpoint in $B' - (A \cup A')$, then its other endpoint cannot lie in $A' - B'$ since (A', B') is separation of $G[B]$; and its other endpoint cannot lie in $A - B'$ since $B' \subseteq B$ and (A, B) is a separation of G. Thus $(A \cup A', B')$ is a separation of G.

To prove the second statement, observe that $A \cap B' \subseteq A \cap B \subseteq A'$ and hence $A \cap B' \subseteq A' \cap B'$. Thus

$$(A \cup A') \cap B' \subseteq (A \cap B') \cup (A' \cap B') \subseteq A' \cap B'. \quad ■$$

As a consequence, we have the following lemma.

Lemma 4.2 *Let $\mathcal{B} = (\mathcal{K}, \Gamma)$ be a blocked hex, and let (A, B) be a separation of $\mathcal{K}$ of order $q < m$ such that $\Gamma \subseteq A$ and B contains $q+1$ super-nodes from some row or column of $\mathcal{K}$. Then $(\mathcal{K}[B], A \cap B)$ is a blocked hex of order (m, q).*

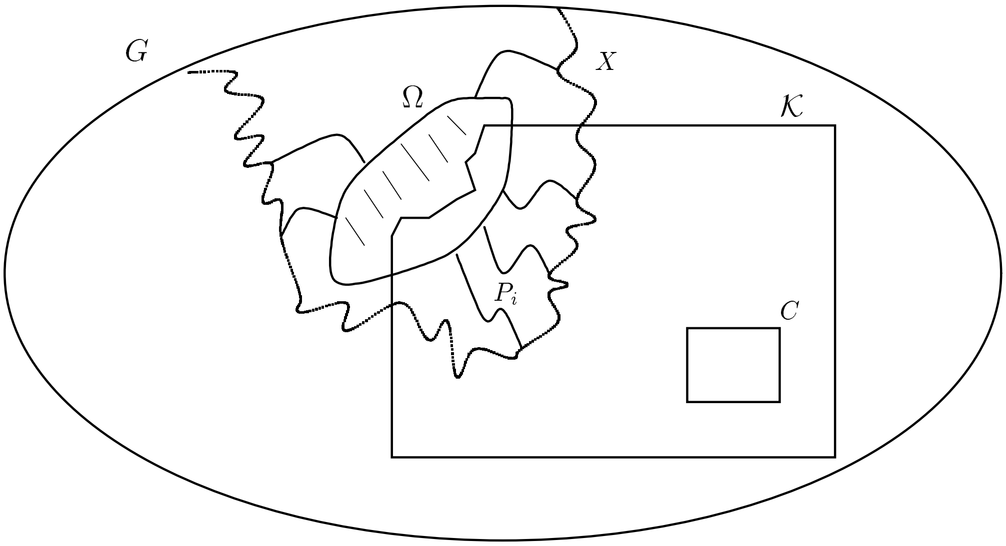

Figure 2: The proof of Theorem 4.3

Theorem 3.5 applies to the case of a blocked hex; but to implement the approach discussed above, we need to extend it to the case of a graph G that only *contains* a blocked hex $(\mathcal{K}, \Gamma)$. In the natural way, we also extend the notion of a set of vertices X in G being *r-linked* to $\mathcal{K}$, even when X does not necessarily lie in $\mathcal{K}$.

The difficulty is that when $\Omega_{\mathcal{T}} \not\subseteq V(\mathcal{K})$, a set of paths linking $\Omega_{\mathcal{T}}$ to a ring of $\mathcal{K}$ might destroy certain portions of $\mathcal{K}$ that we were hoping to preserve for other parts of the overall routing. This is handled by an induction argument; a similar theorem for clique minors was proved in [12] and subsequently, in a somewhat different form, in [2].

Theorem 4.3 *Let G be a graph containing a blocked hex $\mathcal{B} = (\mathcal{K}, \Gamma)$ of order (m, q) as a subgraph, where $m \geq 24k^{3/2} + 2k$. Let $\mathcal{T}$ be an initially feasible set of at most k terminal pairs for which all demands are in $[0, \frac{1}{2}]$, $\Omega_{\mathcal{T}}$ is $4k$-linked to $\mathcal{K}$, and $\Gamma \subseteq \Omega_{\mathcal{T}}$. Then $\mathcal{T}$ is realizable in G.*

Proof. Suppose by way of contradiction that the claim of the theorem is false, and choose a counterexample G, $\mathcal{B}$, $\mathcal{T}$ with $|V(G)| + |E(G)|$ minimum. We write $\Omega = \Omega_{\mathcal{T}}$, and $\ell = |\Omega|$. First, suppose that there is a ring C of $\mathcal{K}$ of order at least $4k$, and a separation (A, B) of G of order ℓ such that $\Omega \subseteq A$ and $C \subseteq B$. Now, define $X = A \cap B$ and $G' = G[B]$. Since Ω is linked to C by ℓ vertex-disjoint paths, it is also linked to X by ℓ vertex-disjoint paths $P_1, \ldots, P_\ell$. Define a set of terminal pairs $\mathcal{T}'$ on X as they are specified by the endpoints of these Ω-X paths. Finally, let $A' = A \cap V(\mathcal{K})$, $B' = B \cap V(\mathcal{K})$, $\mathcal{K}' = \mathcal{K}[B']$, and $\Gamma' = A' \cap B'$. Then $\Gamma \subseteq \Omega \cap V(\mathcal{K}) \subseteq A'$, and B' contains $2q \geq q + 1$ super-nodes from a single row of $\mathcal{K}$ since it contains C. Thus by Lemma 4.2, $(\mathcal{K}', \Gamma')$ is a blocked hex of order (m, q'), where $q' = \max(q, \ell)$, and

$$\Gamma' = A' \cap B' \subseteq A \cap B = X.$$

Also, X is $4k$-linked to $\mathcal{K}'$; for if (A'', B'') is a separation of G' of order $< \ell$, with $X \subseteq A''$ and $C' \subseteq B''$ for some large ring C' of $\mathcal{K}'$, then $(A \cup A'', B'')$ is a separation of order $< \ell$ with $\Omega \subseteq A \cup A''$ and $C' \subseteq B''$, contradicting the initial assumption that Ω is $4k$-linked to $\mathcal{K}$. So we have that G'

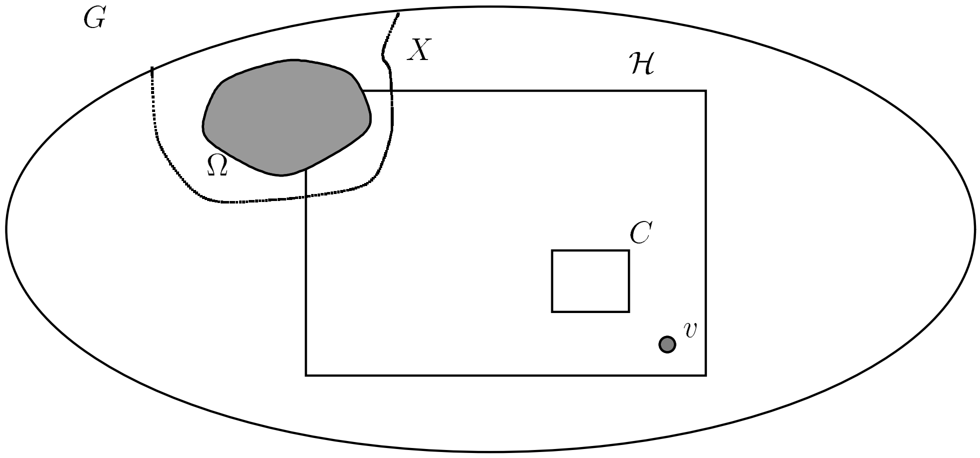

Figure 3: The proof of Theorem 4.4

contains a blocked hex $(\mathcal{K}', \Gamma')$, and a set of at most k terminal pairs $\mathcal{T}'$ for which $\Omega_{\mathcal{T}'} = X$ is $4k$-linked to $\mathcal{K}'$ and $\Gamma' \subseteq X = \Omega_{\mathcal{T}'}$.

If $(A, B) \neq (\Omega, V)$ then the graph G' is smaller than G. Hence by our assumption that G is a minimum counterexample, $\mathcal{T}'$ is realizable in G'; and concatenating the paths $P_1, \dots, P_\ell$ to the feasible routing of $\mathcal{T}'$, we would obtain a feasible routing of $\mathcal{T}$ in G. Thus, we may suppose that if (A, B) is a separation of G of order ℓ with $\Omega \subseteq A$ and $C \subseteq B$ for a large ring C of $\mathcal{K}$, then $A = \Omega$.

Now suppose that some edge of G is not an edge of $\mathcal{K}$ — say, $e \notin E(\mathcal{K})$ — and consider the effect of deleting e from G. If Ω is no longer $4k$-linked to $\mathcal{K}$, there is a separation (A, B) of order $< \ell$ that disconnects Ω from some large ring C. Clearly e could not have both endpoints in Ω, for then Ω would not be $4k$-linked to $\mathcal{K}$ in G; thus one of the endpoints of e does not belong to Ω. If we add this endpoint to both A and B, we obtain a separation (A', B') of G of order $\leq \ell$ for which neither A nor B is equal to Ω, contradicting our assumption above. Thus it follows that if $e \notin E(\mathcal{K})$, then Ω is still $4k$-linked to $\mathcal{K}$ after the deletion of e; since $G - e$ is smaller than G, Ω is realizable in $G - e$ and hence in G, a contradiction.

Hence we may assume that every edge of G is an edge of $\mathcal{K}$. Similarly, we may assume that G contains no isolated vertex, for we may delete such a vertex without affecting our hypotheses. Thus we are left with the case that $G = \mathcal{K}$. Since $\Gamma \subseteq \Omega$, $\mathcal{B}$ has order (m, q) with $q \leq 2k$, and hence by Theorem 3.5 $\mathcal{T}$ is realizable in G, again a contradiction. ■

We say that a vertex v is *irrelevant* to the realizability of $\mathcal{T}$ in G if (i) $\mathcal{T}$ is not realizable in G, or (ii) $\mathcal{T}$ is realizable in both G and $G - \{v\}$.

Theorem 4.4 *Let G be a graph containing a hex $\mathcal{H}$ of order $r = 24k^{3/2} + 2k + 1$. Let $\mathcal{T}$ be an initially feasible set of k terminal pairs for which all demands are in $[0, \frac{1}{2}]$. Then there is a polynomial-time algorithm that finds a vertex v that is irrelevant to the realizability of $\mathcal{T}$ in G.*

Proof. As before, we write $\Omega = \Omega_{\mathcal{T}}$ and $\ell = |\Omega|$. We find a separation (A, B) of minimum order such that $\Omega \subseteq A$, $C \subseteq B$ for some ring C in $\mathcal{H}$ of order at least $4k$, and so that A is maximal subject to these conditions. (Such a separation can be found efficiently by a standard maximum flow algorithm; see e.g. (6.2) of [12].) Let $G' = G[B]$, $\mathcal{K} = \mathcal{H}[B]$, $X = A \cap B$, $\Gamma = V(\mathcal{H}[X])$, and $\ell' = |X| \leq \ell$. Since $C \subseteq B$, B contains at least $2\ell \geq \ell + 1$ super-nodes from a single row of $\mathcal{H}$;

hence $(\mathcal{K}, \Gamma)$ is a blocked hex of order (r, q) in G', with $q \leq \ell$. We choose any vertex $v \in \mathcal{K} - X$, and claim that such a v is irrelevant to the realizability of $\mathcal{T}$ in G.

To establish this, we first form $G'' = G' - \{v\}$. G'' contains a blocked hex $(\mathcal{K}', \Gamma)$ of order $(r-1, q)$. Consider any set $\mathcal{T}'$ of at most k terminal pairs obtained by assigning demands in $[0, \frac{1}{2}]$ to pairs of vertices in X, so that each vertex of X receives a non-zero amount of demand, and no vertex in X receives more than one unit of demand. We claim that G'', $(\mathcal{K}', \Gamma)$, and $\mathcal{T}'$ satisfy the hypotheses of Theorem 4.3 — it will then follow that $\mathcal{T}'$ is realizable in G''. In particular, we must verify that X is $4k$-linked to $\mathcal{K}'$. If not, then there is a separation (S, T) of order $< \ell'$ with $X \subseteq S$ and $C' \subseteq T$, where C' is a large ring of $\mathcal{K}'$. If we add v to both S and T, we obtain a separation (S', T') of order $\leq \ell'$ in G', with $X \subseteq S'$ and $C' \subseteq T'$. But now $(A \cup S', T')$ is a separation of order $\leq \ell'$ in G, with $\Omega \subseteq A \cup S'$, $C' \subseteq T'$; and A is properly contained in $A \cup S'$ since $v \in S' - A$. This contradicts our original choice of the separation (A, B).

Now, suppose that $\mathcal{T}$ is realizable in G, and consider any feasible routing of $\mathcal{T}$ by paths $P_1, \ldots, P_k$. Define P_i' to be the sub-path of P_i between its end at s_i and its first meeting with a vertex $u_i \in X$; define P_i'' to be the sub-path of P_i between its end at t_i and its last meeting with a vertex $v_i \in X$. Note that the paths P_i' and P_i'' lie in the graph $G[A]$. By the argument above, there is a feasible routing of the set of pairs $\{(u_i, v_i)\}$ in the graph G'', by paths $Q_1, \ldots, Q_k$. (To satisfy the above hypotheses, we must also add small "artificial demands" to vertices in X not met by any of $P_1, \ldots, P_k$; but we need not consider the paths associated with these artificial demands for the result.) If we define R_i to be the concatenation of P_i', Q_i, and P_i'', we see that the paths $R_1, \ldots, R_k$ constitute a feasible routing of $\mathcal{T}$ in G that avoids the vertex v. Thus v is irrelevant, as desired. ■

5 Bounded Branch-Width

We will see via the main result of [13] that any graph with sufficiently large *branch-width* (defined below) must contain a large hex subgraph, and therefore an irrelevant vertex by Theorem 4.4. In this section, we show how the realizability of $\mathcal{T}$ can be decided when the branch-width of G is bounded.

The main issue here will be to keep track of the combinatorially distinct ways in which paths in a routing of $\mathcal{T}$ can loop back and forth through the vertices of a small separator. This is captured by the notion of a *passage*. If $X \subseteq V$ and $\mathcal{D} = \{\rho_1, \ldots, \rho_\ell\}$ is a (multi)set of reals in $[0, 1]$, then we define a *passage* Λ on $(X, \mathcal{D})$ as follows: for each $\rho_i \in \mathcal{D}$, we choose a forest Λ_i of maximum degree 2 on the set X. The edges of these graphs Λ_i will be called the *edges of* Λ. We note the following lemma.

Lemma 5.1 *There are at most $((2|X|)!)^{|\mathcal{D}|}$ distinct passages on $(X, \mathcal{D})$.*

The passage Λ induces an instance of the UFP on G in a natural way: the terminal pairs $\mathcal{T}(\Lambda)$ consist of the edges of Λ. We carry over the notions of *routings* and *realizability* to Λ; they are defined in terms of $\mathcal{T}(\Lambda)$.

If $\mathcal{T}$ is a set of terminal pairs on G, we say that a routing of $\mathcal{T}$ is *primitive* if its paths meet the vertices in $\Omega_{\mathcal{T}}$ only at their endpoints. We say $\mathcal{T}$ is *primitively realizable* if there is a primitive routing of $\mathcal{T}$.

Lemma 5.2 *Let $\mathcal{T}$ have terminal set Ω and demand set $\mathcal{D}$. Then $\mathcal{T}$ is realizable in G if and only if there is a primitively realizable passage Λ on $(\Omega, \mathcal{D})$, for which each Λ_i is a path with endpoints s_i, t_i.*

Let $\mathcal{R}_G(X,\mathcal{D})$ denote the set of all primitively realizable passages on $(X,\mathcal{D})$. We now show that we can build up such sets inductively, when G can be partitioned by a small separator.

Lemma 5.3 *Let $X \subseteq V$ have size k, and let $\mathcal{D}$ be a set of reals in [0,1], with $|\mathcal{D}| = \ell$. Then given $\mathcal{R}_G(X,\mathcal{D})$, and $X' \subseteq X$, we can construct $\mathcal{R}_G(X',\mathcal{D})$ in time $k^{O(k\ell)}$.*

Proof. The difficulty is that a passage on $(X,\mathcal{D})$ may not be primitively realizable, but the corresponding passage on $(X',\mathcal{D})$ may be primitively realizable. For each passage Λ on $(X',\mathcal{D})$, we decide whether it is primitively realizable as follows. For each graph Λ_i, we try all possible subdivisions of it by elements of $X - X'$ that result in a maximum degree of 2; this produces $k^{O(k)}$ possible graphs for each $i = 1,\ldots,\ell$. For each of the $k^{O(k\ell)}$ passages Λ' on $(X,\mathcal{D})$ that we produce from Λ in this way, we see whether $\Lambda' \in \mathcal{R}_G(X,\mathcal{D})$. ■

Lemma 5.4 *Let (A,B) be a separation of G, and let $X \supseteq A \cap B$, with $|X| = k$. Let $\mathcal{D}$ be a set of reals in [0,1], with $|\mathcal{D}| = \ell$. Then given $\mathcal{R}_{G[A]}(X \cap A,\mathcal{D})$ and $\mathcal{R}_{G[B]}(X \cap B,\mathcal{D})$, we can construct $\mathcal{R}_G(X,\mathcal{D})$ in time $k^{O(k\ell)}$.*

Proof. For each passage Λ on $(X,\mathcal{D})$ in G, we must decide whether it is primitively realizable. First, for each $v \in X$, let $S_v \subseteq \{1,\ldots,\ell\}$ denote the set of indices i such that v is not isolated in Λ_i. Consider the following *endpoint constraint*: for each v, $\sum_{i\in S_v} \rho_i \leq 1$. If the endpoint constraint is violated, then clearly Λ is not primitively realizable. If the endpoint constraint is satisfied, and Λ is not primitively realizable, then every primitive routing of Λ must violate the capacity constraint at some vertex in $V - X$; for the paths in any primitive realization of Λ cannot use X except at their endpoints.

If Λ has a feasible primitive routing, each edge f of Λ must be realized by a path that has no internal vertex in X; in particular, this means that it is realized by a path lying completely in one of $G[A]$ or $G[B]$. A *labeling* of Λ will be a choice, for each edge f of Λ, of the subgraph in which it is contained. Consider a fixed labeling of the edges of Λ; let Λ' be the passage on $(X \cap A,\mathcal{D})$ in $G[A]$ defined by the edges labeled A, and Λ'' be the analogous passage on $(X \cap B,\mathcal{D})$ in $G[B]$. Then if the endpoint constraint of the previous paragraph is satisfied, Λ has a feasible primitive routing consistent with the given labeling if and only if Λ' and Λ'' are each primitively realizable in the subgraphs $G[A]$ and $G[B]$ respectively.

Thus, we have the following algorithm to decide if $\Lambda \in \mathcal{R}_G(X,\mathcal{D})$: for each labeling of Λ, construct Λ' and Λ'', and determine whether they are contained in $\mathcal{R}_{G[A]}(X \cap A,\mathcal{D})$ and $\mathcal{R}_{G[B]}(X \cap B,\mathcal{D})$ respectively. As there are $2^{O(k\ell)}$ labelings, this takes time $2^{O(k\ell)}$. By Lemma 5.1, the total time over all passages is thus $k^{O(k\ell)}$. ■

Now we use Lemma 5.4 as part of a method to solve the UFP when G has a decomposition in terms of small separators. The overall approach is a type of dynamic programming over this separator decomposition; a framework for developing such algorithms is presented by Arnborg and Proskurowski [1], and it also forms a component of the R-S algorithm. Given $\mathcal{T}$, with terminal set Ω and demand set $\mathcal{D}$, we solve the more general problem of determining $\mathcal{R}_G(\Omega,\mathcal{D})$; by Lemma 5.2, we can determine the realizability of $\mathcal{T}$ from this information in time roughly proportional to $|\mathcal{R}_G(\Omega,\mathcal{D})|$.

A branch-decomposition, following [12], is a ternary tree T together with an assignment τ of a distinct edge of G to each leaf of T. Removing an edge from T partitions T, and hence the edge set E of G, into two pieces; the *width* of (T,τ) is the maximum, over all such partitions (E_1,E_2), of number of vertices of G incident to an edge in both E_1 and E_2. The *branch-width* of G is the minimum width of a branch-decomposition of G.

Given a branch-decomposition (T,τ) of width ω, we proceed as follows. We fix an arbitrary internal node α of T, and root T at α. For each node β of T, let B_β denote the set of endpoints of edges in E that are assigned to the subtree of T below β. Let A_β denote the set of endpoints of edges in E that are assigned to leaves of T not below β. Let $W_\beta = A_\beta \cap B_\beta$ and $X_\beta = (Z \cap B_\beta) \cup W_\beta$; note that $|X_\beta| \leq k+\omega$. We construct $\mathcal{R}_{G[B_\beta]}(X_\beta, \mathcal{D})$ by working up T from its leaves; when β is a leaf node this is easy (since $|B_\beta| = 2$), and when $\beta = \alpha$ we are done. (At α we must combine three sets rather than two; but we can do this in two steps.)

Now, if β has children β' and β'', we observe that $B_\beta = B_{\beta'} \cup B_{\beta''}$ and

$$B_{\beta'} \cap B_{\beta''} \subseteq W_{\beta'} \cap W_{\beta''} \subseteq X_{\beta'} \cap X_{\beta''}.$$

Thus by Lemma 5.4, we can compute $R_{G[B_\beta]}(X_{\beta'} \cup X_{\beta''}, \mathcal{D})$ in time $(k+\omega)^{O((k+\omega)k)}$. We now observe that $W_\beta \subseteq W_{\beta'} \cup W_{\beta''}$, and so $X_\beta \subseteq X_{\beta'} \cup X_{\beta''}$; thus by Lemma 5.3, we can compute $R_{G[B_\beta]}(X_\beta, \mathcal{D})$ from $R_{G[B_\beta]}(X_{\beta'} \cup X_{\beta''}, \mathcal{D})$ in time $(k+\omega)^{O((k+\omega)k)}$. Summing these times up the tree to the root α, we obtain

Theorem 5.5 *Given a branch-decomposition of G of width ω, we can decide the realizability of $\mathcal{T}$ in time $[|E| \cdot (k+\omega)^{O((k+\omega)k)}]$.*

6 Decision Algorithms for Realizability

We need the following combination of results of Robertson, Seymour, and Thomas [13] and Robertson and Seymour [12].

Theorem 6.1 *[12, 13]*

(i) There exists a algorithm that, given a graph G and an integer m, finds either a subgraph of G isomorphic to a hex of order m or a branch-decomposition of G of width $\omega \leq 20^{2m^5}$. The running time is $[n^2 \cdot (m^2+\omega)^{O(m^2+\omega)}]$.

(i) There exists a algorithm that, given a planar graph G and an integer m, finds either a subgraph of G isomorphic to a hex of order m or a branch-decomposition of G of width $\omega \leq 12m$. The running time is $[n^2 \cdot (m^2+\omega)^{O(m^2+\omega)}]$.

We note that the algorithms guaranteed by parts (i) and (ii) of this theorem are actually quite simple — involving only dynamic programming in graphs of bounded branch-width — and they do not involve large constants other than the (fairly large) bound in (i).

We can now give the basic decision algorithm that proves Theorem 1.1.

Proof of Theorem 1.1. We first call the algorithm of Theorem 6.1 on G with $m = 24k^{3/2}+2k+1$. Using the bound we have assumed on k, the algorithm runs in time polynomial in n. If the algorithm finds a hex of order m, then we may use Theorem 4.4 to locate an irrelevant vertex v; we then iterate on $G - \{v\}$. Otherwise, it produces a branch-decomposition of G whose width (again by our bound on k) is $O\left(\frac{\log n}{(\log\log n)^2}\right)$. By Theorem 5.5, we can then decide the realizability of $\mathcal{T}$ in G in polynomial time. ■

The case of planar graphs is strictly analogous.

Proof of Theorem 1.2. Again, we call the algorithm of Theorem 6.1 on G with $m = 24k^{3/2}+2k+1$; if a hex of order m is returned, we locate an irrelevant vertex and iterate. Otherwise, it produces a branch-decomposition of G whose width is $O\left(\left(\frac{\log n}{\log\log n}\right)^{1/2}\right)$, and we apply Theorem 5.5. ■

Polynomial-time algorithms for the three optimization problems defined in the introduction follow immediately from the corresponding decision algorithms in Theorems 1.1 and 1.2, since k is so small relative to n. To find a realizable subset of terminal pairs of maximum total demand, we simply test the realizability of all subsets of $\mathcal{T}$. To find a routing of minimum capacitated congestion, we simply determine the realizability of $\mathcal{T}$ when the vertex capacities are set to each of the possible values of $\rho_{i_1} + \cdots + \rho_{i_\ell}$ that are at least 1. Finally, to determine the minimum number of rounds needed for routing $\mathcal{T}$, we first enumerate all collections of at most k realizable subsets of $\mathcal{T}$, and find the smallest such collection that constitutes a set cover.

References

[1] S. Arnborg, A. Proskurowski, "Linear-time algorithms for NP-hard problems restricted to partial k-trees," *Disc. Appl. Math.*, 8(1987), pp. 277–284.

[2] B. Bollobás, A. Thomason, "Highly linked graphs," *Combinatorica* 16(1996), pp. 313–320.

[3] A. Frank, "Packing paths, cuts, and circuits — a survey," in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, A. Schrijver, Eds., Berlin: Springer-Verlag, 1990, pp. 49–100.

[4] N. Garg, V. Vazirani, M. Yannakakis, "Primal-dual approximation algorithms for integral flow and multicut in trees, with applications to matching and set cover," *Proc. ICALP*, 1993, pp. 64–75.

[5] R.M. Karp, "On the computational complexity of combinatorial problems," *Networks* 5(1975), pp. 45–68.

[6] J. Kleinberg, "Single-source unsplittable flow," *Proc. 37th IEEE FOCS*, 1996.

[7] J. Kleinberg, É. Tardos, "Disjoint Paths in Densely Embedded Graphs," *Proc. 36th IEEE FOCS*, 1995, pp. 52–61.

[8] S. Kolliopoulos, C. Stein, "Improved approximation algorithms for unsplittable flow problems," *Proc. 38th IEEE FOCS*, 1997.

[9] M. Middendorf, F. Pfeiffer, "On the complexity of the disjoint paths problem," *Combinatorica*, 13(1993). pp. 97–107.

[10] N. Robertson, P.D. Seymour, "An outline of a disjoint paths algorithm," in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, A. Schrijver, Eds., Berlin: Springer-Verlag, 1990, pp. 267–292.

[11] N. Robertson, P.D. Seymour, "Graph Minors V. Excluding a planar graph," *J. Combinatorial Theory Ser. B*, 41(1986), pp. 92–114.

[12] N. Robertson, P.D. Seymour, "Graph Minors XIII. The disjoint paths problem," *J. Combinatorial Theory Ser. B*, 63(1995), pp. 65–110.

[13] N. Robertson, P.D. Seymour, R. Thomas, "Quickly excluding a planar graph," *J. Combinatorial Theory Ser. B*, 62(1994), pp. 323–348.

[14] A. Schrijver, "Disjoint homotopic paths and trees in a planar graph," *Discrete and Computational Geometry*, 6(1991), pp. 527–574.

[15] A. Schrijver, "Homotopic routing methods," in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, A. Schrijver, Eds., Berlin: Springer-Verlag, 1990, pp. 329–372.

[16] A. Schrijver, P.D. Seymour, P. Winkler, "The ring loading problem," *SIAM J. Discrete Math.*, to appear.

[17] A. Srinivasan, "Improved approximations for edge–disjoint paths, unsplittable flow, and related routing problems," *Proc. 38th IEEE FOCS*, 1997.