Edge-Disjoint Paths in Planar Graphs

C. Chekuri[*] S. Khanna[†] F. B. Shepherd[‡]

Abstract

We study the maximum edge-disjoint paths *problem (MEDP). We are given a graph $G = (V, E)$ and a set $\mathcal{T} = \{s_1t_1, s_2t_2, \ldots, s_kt_k\}$ of pairs of vertices: the objective is to find the maximum number of pairs in $\mathcal{T}$ that can be connected via edge-disjoint paths. Our main result is a* poly-logarithmic approximation *for MEDP on* undirected planar graphs *if a congestion of* 2 *is allowed, that is, we allow up to* 2 *paths to share an edge. Prior to our work, for any constant congestion, only a polynomial-factor approximation was known for planar graphs although much stronger results are known for some special cases such as grids and grid-like graphs. We note that the natural multicommodity flow relaxation of the problem has an integrality gap of $\Omega(\sqrt{|V|})$ even on planar graphs when no congestion is allowed. Our starting point is the same relaxation and our result implies that the integrality gap shrinks to a poly-logarithmic factor once* 2 *paths are allowed per edge. Our result also extends to the* unsplittable flow *problem and the* maximum integer multicommodity flow *problem.*

A set $X \subseteq V$ is well-linked *if for each $S \subset V$, $|\delta(S)| \geq \min\{|S \cap X|, |(V - S) \cap X|\}$. The heart of our approach is to show that in any undirected planar graph, given any matching M on a well-linked set X, we can route $\Omega(|M|)$ pairs in M with a congestion of* 2*. Moreover, all pairs in M can be routed with constant congestion for a sufficiently large constant. This results also yields a different proof of a theorem of Klein, Plotkin, and Rao that shows an $O(1)$ maxflow-mincut gap for uniform multicommodity flow instances in planar graphs.*

The framework developed in this paper applies to general graphs as well. If a certain graph theoretic conjecture is true, it will yield poly-logarithmic integrality gap for MEDP with constant congestion.

[*]Bell Labs, 600 Mountain Ave, Murray Hill, NJ 07974. Email: chekuri@research.bell-labs.com.

[†]Dept. of CIS, U. of Pennsylvania, Philadelphia PA. Email: sanjeev@cis.upenn.edu. Supported in part by an Alfred P. Sloan Research Fellowship and by an NSF Career Award CCR-0093117.

[‡]Bell Labs, 600 Mountain Ave, Murray Hill, NJ 07974. Email: bshep@research.bell-labs.com.

1 Introduction

In this paper we consider the classical *edge-disjoint path* problem (EDP). We are given a graph $G = (V, E)$ and a set $\mathcal{T} = \{s_1t_1, s_2t_2, \ldots, s_kt_k\}$ of pairs of vertices. The objective is to connect the pairs via edge-disjoint paths. In the maximum edge-disjoint path problem (MEDP), the objective is to find the maximum number of pairs in $\mathcal{T}$ that can be connected via edge-disjoint paths. Generalizations of EDP include the *unsplittable flow problem* (UFP) and *maximum integer multicommodity flow problem* (IMF). In UFP and IMF each pair s_it_i from $\mathcal{T}$ has an integer demand d_i and the graph has integer edge capacities, u_e, $e \in E$. In UFP the goal is to route the maximum number of pairs where a pair is routed if its full demand is sent along a single path. In IMF the goal is to route the maximum amount of integer flow between the pairs. In addition to being fundamental problems in combinatorial optimization, EDP, UFP, and IMF have a wide variety of applications in VLSI layout and virtual circuit routing in high-speed networks. Consequently these problems and their variants have been extensively studied. EDP was shown to be NP-hard even on very restricted instances including planar graphs [16]. Early work on this problem focused on characterizing classes of instances for which EDP and MEDP can be solved in polynomial time. See [14] for a survey. This includes the seminal work of Robertson and Seymour [35] that gave a polynomial time algorithm for EDP (and vertex disjoint paths) in undirected graphs when the number of pairs is a fixed constant independent of the input size.

The focus has recently shifted to finding approximation algorithms for these problems. Although constant factor and poly-logarithmic factor approximation algorithms are known for restricted classes of graphs such as trees, meshes, and highly connected graphs such as expanders, the approximability of these problems in general graphs is not well understood. The best approximation ratio for MEDP in undirected graphs is $O(\min(n^{2/3}, \sqrt{m}))$ where n and m are the number of vertices and edges respectively. On the other hand we only know APX-hardness [17]. For directed graphs Guruswami et al. [18] established a hardness of $\Omega(n^{1/2-\epsilon})$. The natural LP relaxation for the problem has an $\Omega(\sqrt{n})$ integrality gap even on planar graphs [17].

On the positive side, it is known that MEDP and UFP become easier if the capacity of each edge is *large* compared to the maximum demand. Raghavan and Thompson [34] showed via randomized rounding that a constant factor approximation is achievable for MEDP even in directed graphs if $\Omega(\log n)$ paths are allowed on each edge. Even with a modest relaxation on the number of paths allowed per edge, the problems exhibit a qualitative and quantitative improvement in tractability. In particular the *half-disjoint paths* problem in which two paths are allowed on each edge has received attention. This is partly motivated by a number of *half-integrality results* known for flow problems. A well known example is the theorem of Okamura and Seymour [32] that states that if all the terminals are on the outer face of a planar graph, then there is a half-integral flow for all pairs if and only if the natural cut condition is satisfied. For a strengthening of this result see Frank [13]. For the half-disjoint paths problem, Kleinberg [25] showed that the framework of Robertson and Seymour [35] can be simplified to gave a decision algorithm that works for a super-constant number of pairs (up to $\Omega((\log\log n)^{2/15})$ pairs). We note that no super-constant integrality gap is known for the LP relaxation for the half disjoint path problem even in directed graphs. Although allowing more than one path per edge seems to improve the tractability of the problem, the best approximation ratio known for general graphs improves only to $O(n^{1/B})$ where B is the number of paths allowed per edge [40]. This bound also applies to general packing problems [3] and it has been shown in [6] that no better ratio is possible for packing problems unless $NP = ZPP$.

In this paper we design a poly-logarithmic approximation algorithm for MEDP on undirected planar graphs if 2 paths per edge are allowed[1] a high level our approach also applies to general graphs and would yield a poly-logarithmic approximation provided certain graph theoretic properties hold. Our main result is the following.

Theorem 1.1 *Given an instance of MEDP for a planar graph G, there is a polynomial time algorithm that routes* $\Omega(\text{OPT}/\log^2 n \log\log n)$ *pairs such that the number of paths using an edge is at most* 2*.*

Corollary 1.2 *There is an* $O(\log^2 n \log\log n)$*-approximation algorithm for instances of UFP on a planar graph G which satisfy the condition* $d_{\max} \le u_{\min}/2$*. For IMF we obtain a similar approximation if* $u_{\min} \ge 2$*.*

To obtain the above theorem we prove the following theorem which is of independent interest. We say that a set $X \subseteq V$ in G is *well-linked*[2] if the following cut condition is satisfied: for any set $S \subset V$ with $|S \cap X| \le |X|/2$, $|\delta(S)| \ge |S \cap X|$.

Theorem 1.3 *There exists a universal constant* C *such that given any planar graph* G*, a well-linked set* X *in* G*, and any matching* M *on* X*,* $|X|/C$ *pairs in* M *can be routed with at most* 2 *paths using an edge. Moreover, if we allow up to* 3 *paths per edge, then* any *subset of* $|X|/C$ *pairs in* M *can be routed. It follows that any matching* M *on* X *can be routed with* $O(C)$ *congestion.*

A corollary of the above is the following.

Corollary 1.4 *The maxflow-mincut gap for product multicommodity flow instances in planar graphs is* $O(1)$*. In particular the gap for uniform multicommodity flow instances in planar graphs is* $O(1)$*.*

Klein, Plotkin and Rao [23] showed that the maxflow-mincut gap in uniform multicommodity flow problems in planar graphs is $O(1)$. In graphs excluding a K_r minor, they show a gap of $O(r^3)$ which was improved to $O(r^2)$ in [12]. Theorem 1.3 and Corollary 1.4 show that we can obtain a constant gap while ensuring that the flow for each pair is along a single path. Interestingly our proof of the above is very different from the dual based methods used to prove maxflow-mincut theorems [29, 23, 30, 2]. It is closer in spirit to the Okamura-Seymour theorem for planar graphs [32] that works directly with the cut condition.

1.1 Overview of the Algorithm

A central concept used by our algorithm is that of a *crossbar*. Given an instance of MEDP on a graph we say that the pairs in $\mathcal{T}$ are *routable* if we can find edge-disjoint paths for them. A graph $H = (V, E)$ is a *crossbar* with respect to $Y \subseteq V$ (or a Y*-crossbar*) if each matching on the graph $(Y, \{uv : u, v \in Y\})$ is routable in H. We may also call H a Y*-crossbar*. The set Y is called the *interface* of the crossbar. We say H is a k*-crossbar* if it is a Y-crossbar for some $|Y| \ge k$. A weaker notion is that of a well-linked graph. A subset Y is *well-linked* in G if for each subset S of $V(G)$, we have that $|\delta(S)| \ge \min\{|S \cap Y|, |(V - S) \cap Y|\}$. We say that G is k*-well-linked* if it is well-linked for some subset of size k. Clearly, if there is a Y-crossbar, then Y is well-linked.

In the following, our set of pairs is: $\mathcal{T} = \{s_1t_1, \ldots, s_kt_k\}$ and let $X = \{s_i, t_i : i = 1, 2 \ldots, k\}$ be the set of *terminals*. We can assume without loss of generality that each terminal occurs in exactly one pair in $\mathcal{T}$. Throughout, we let OPT denote the maximum cardinality of a routable subset of $\mathcal{T}$. The high level outline of the algorithm is as follows.

[1] We abuse usage and refer to all problems as edge-disjoint path problems even though we sometimes allow 2 paths per edge.

[2] In the work of Robertson and Seymour and others, the term well-linked is used for vertex connectivity, here we use it for edge connectivity.

The Algorithm

1. Given G and $\mathcal{T}$, we first decompose G into node-disjoint induced subgraphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_\ell = (V_\ell, E_\ell)$ and find subsets $\mathcal{T}_i \subset \mathcal{T}$, $1 \le i \le \ell$ with the following properties.

   - For each pair $s_j t_j \in \mathcal{T}_i$, s_j and t_j are in V_i.
   - For $1 \le i \le \ell$, if X_i is the set of terminals in $\mathcal{T}_i$ then X_i is well-linked in G_i.
   - $\sum_i |\mathcal{T}_i| = \Omega(\text{OPT}/\log^2 n \log\log n)$.

   We now solve the MEDP instances $(G_i, \mathcal{T}_i)$, $1 \le i \le \ell$ separately.

2. This step is based on the following result: if a graph is a k-well-linked, then it has an integral crossbar that is almost as large. This step amounts to computing such an integral crossbar.

3. We either find a way to satisfy much of $\mathcal{T}_i$ by routing to the crossbar, or show that we may reduce the size of the graph G_i while maintaining well-linkedness of X_i. In the latter case, we go back to Step 2.

The first step is accomplished by the ideas of the authors in [4] which are based on the hierarchical decomposition developed by Räcke [33] for oblivious routing. The second and third steps employ the strategy of Robertson and Seymour: route the terminals of the demand to a crossbar, and connect them through it. We use planarity in Step 2 in establishing the existence of an integral crossbar. Finally, in Step 3 we argue the existence of an edge that can be removed without destroying the well-linkedness of X_i. We mention that we lose a poly-logarithmic factor only in Step 1. After that we only lose constant factors.

Finally, our approach will achieve a poly-logarithmic integrality gap for any class of graphs for which a "well-linked implies integral crossbar" result holds as in Step 2. In planar graphs, one could use (although it does not directly yield a polynomial time algorithm for a non-constant number of terminals) a result of Robertson, Seymour and Thomas that planar graphs with tree width h have a crossbar of size $h/25$. In general graphs, the tree width decomposition results of Robertson and Seymour are too weak to obtain good approximations by this method. Their results are of the form: if G has tree width 25^{35h^2}, then it has a crossbar of size h. We on the other hand seek results of the form: if G has tree width h, then there is a crossbar of size $h/\text{poly-log}(n)$. This is the key result needed to make our scheme yield a poly-logarithmic approximation for MEDP in general graphs.

1.2 Related Work

As mentioned earlier, there is extensive literature on EDP and related problems. Here we only mention the work that is directly relevant to this paper. The vertex disjoint path problem was shown to be NP-hard by Karp [20] and Lynch [31] showed the same for planar graphs - these results imply NP-hardness for EDP in directed graphs. EDP in undirected graphs was shown to be NP-hard by Even, Itai, and Shamir [11] and for planar graphs by Kramer and van Leeuwen [28]. For directed graphs Fortune, Hopcroft and Wylie [15] showed that EDP is NP-hard even for two pairs. The decision version for fixed number of pairs was open for undirected graphs until Robertson and Seymour [35] obtained a polynomial time algorithm via their seminal work on graph minors. We borrow some key ideas from their work. See [14, 39] for surveys on polynomial time solvable cases and other efficient characterizations for both the decision and the maximization versions.

Work on disjoint paths is closely connected to the work on flows. For a single pair (s, t), Menger's theorem and Ford-Fulkerson's integral maxflow-mincut theorem provide efficient ways to compute the maximum number of edge (vertex) disjoint paths between s and t. For multicommodity flow instances there is no such nice characterization. Leighton and Rao [29] obtained the first maxflow-mincut gap theorem: they showed an $O(\log n)$ bound on the gap for uniform and product multicommodity flow instances. Subsequently this was generalized to arbitrary multicommodity flow instances by a series of papers and finally an $O(\log k)$ bound was obtained for all instances [30, 2] where k is the number of commodities. Klein, Plotkin, and Rao [23] established an $O(1)$ gap for uniform multicommodity flow instances in planar graphs. Unlike the single commodity maxflow-mincut theorem of Ford-Fulkerson, the multicommodity gap theorems do not establish integrality of the flows. In fact a gap of $\Omega(\sqrt{n})$ exists between max-integral-flow and mincut as shown in [17]. However it has been an important open problem if the gap between the max-half-integral-flow and mincut can be bounded by a poly-logarithmic factor. Our result establishes a poly-logarithmic factor upper bound on this gap in planar graphs.

For MEDP, only polynomial approximation ratios are known for general graphs. The best approximation ratio for general undirected and directed graphs is $\tilde{O}(\min(n^{2/3}, \sqrt{m}))$ [24, 40, 26, 7, 41]. Constant factor and poly-logarithmic approximation ratios are known for various special classes of graphs such as trees [17], meshes [1, 21], grid-like graphs [22], and expanders and other highly connected graphs [27]. See [24] for an overview of generalizations to UFP and related problems. We mention one variant, that of minimizing *congestion*. In this problem, we are given an instance of EDP and the objective is to find the smallest α such that multiplying the edge capacities by α admits a routing for *all* the pairs in $\mathcal{T}$. Raghavan and Thompson [34] showed that the minimum congestion can be approximated to an $O(\log n/\log\log n)$ factor. For directed graphs, Chuzhoy and Naor [9] showed an $\Omega(\log\log n)$ factor hardness of approximation for minimizing congestion. No better hardness factor larger than 2 is known for undirected graphs.

Recently, the authors considered a relaxation of the MEDP called the all-or-nothing multicommodity flow problem [4]. An instance for this problem is the same as that for MEDP, however the objective is to maximize the number of pairs for which a multicommodity flow of one unit each can be sent in the graph. Using Räcke's [33] hierarchical decomposition of undirected graphs, [4] obtained a poly-logarithmic approximation. This paper builds on some insights obtained in [4].

Organization: The rest of the paper is organized as follows. In Section 2 we make some observations to simplify the problem. Then we follow the outline of the algorithm. In Section 3 we show that if G is k-well-linked, then there is an $\Omega(k)$ sized grid minor. In Section 4 we prove that if a well linked set X cannot route to the interface of a large grid minor then we can remove an edge. In Section 5 we put together the details and prove our main results.

2 Preliminaries

We simplify the problem via two reductions that we describe next.

2.1 Unit Capacity Well-Linked Graphs

We work with a given capacitated graph $G = (V, E, u)$ and we assume that each capacity $u(e)$ is an integer. We let $n = |V|$ and $m = |E|$. Throughout, for any graph G and proper node subset $S \subseteq V$, we denote by $\delta_G(S)$, or simply $\delta(S)$ if G is clear from the context, the set of edges of G with exactly one endpoint in S.

For the given MEDP instance with $\mathcal{T} = \{s_1t_1, s_2t_2, \ldots, s_kt_k\}$, we let $\mathcal{P}_i$ denote the paths joining s_i and t_i in G and let $\mathcal{P} = \cup_i \mathcal{P}_i$. The following multicommodity flow relaxation is used to obtain an upper bound on the number of pairs from $\mathcal{T}$ that can be routed in G. With each path $P \in \mathcal{P}$ we have a variable $x(P)$ which is the amount of flow sent on P. We let $x_i = \sum_{P\in\mathcal{P}_i} x(P)$ denote the total flow sent for pair i. Then the LP relaxation is the following.

$$\max \sum_{i=1}^{k} x_i \quad \text{s.t}$$

$$\begin{aligned} \sum_{P:e\in P} x(P) &\leq u_e \quad \forall e \in E \\ x_i, x(P) &\in [0,1] \end{aligned}$$

We let OPT denote the optimum solution value to the above relaxation. Call a path P fractionally routed if $x(P) \in (0, 1)$, otherwise $x(P) \in \{0, 1\}$ and P is integrally routed. If the total flow routed on integrally routed paths is more than $\text{OPT}/2$ then we already obtain a 2-approximation. Thus the interesting and difficult case is when the fractionally routed paths contribute almost all the value of OPT. From standard polyhedral theory the number of fractionally routed paths in a basic solution to the above LP is at most m. Therefore we can assume that $u_e \leq m$ for all edges. By making parallel copies of edges, in the following, we assume that G has only unit capacity edges. We also assume without loss of generality that each terminal participates in exactly one pair: if a vertex v participates in multiple pairs we can create additional nodes, one for each pair that v participates in, and attach it by a single edge to v. Let X denote the set of terminals in $\mathcal{T}$. We now state a theorem that is implicit in [4]. It is based on the hierarchical decomposition and oblivious routing scheme of Räcke [33]. We use the improved bounds on the decomposition that were obtained in [19].

Theorem 2.1 *Let OPT be the value of an optimal solution to the LP for a given instance $(G, \mathcal{T})$ of MEDP in G. Then G can be partitioned in polynomial time into node-disjoint induced subgraphs $G_1, G_2, \ldots, G_\ell$ and we can find subsets $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_\ell$ of $\mathcal{T}$ such that: (i) if $s_jt_j \in \mathcal{T}_i$ then s_j and t_j are in G_i, (ii) $\sum_{i=1}^{\ell} |\mathcal{T}_i| = \Omega(\text{OPT}/\log^3 n \log\log n)$, and (iii) for $1 \leq i \leq \ell$, the set X_i of terminals in $\mathcal{T}_i$ are well-linked in G_i. If G is planar, $\sum_{i=1}^{\ell} |\mathcal{T}_i| = \Omega(\text{OPT}/\log^2 n \log\log n)$.*

We can thus restrict ourselves to working with unit capacity graphs with a well-linked set of terminals X.

2.2 Bounded Degree Graphs

An $r \times c$ *grid* is a graph $G_{r,c}$ with rc nodes $\{(i, j) : i = 1, 2, \ldots r, j = 1, 2, \ldots c\}$ and with edge set $(\cup_{i=1}^{r} R_i) \cup (\cup_{j=1}^{c} C_j)$, where R_i and C_j denote the *row i edges* and *column j edges* respectively. These latter sets are defined as follows: $R_i = \{(i, j)(i, j + 1) : j = 1, 2, \ldots, c - 1\}$ and $C_j = \{(i, j)(i + 1, j) : i = 1, 2, \ldots, r - 1\}$: see Figure 1. We similarly define a node to be in *column j* if it is of the form (i, j) for some j, and similarly for nodes in row i. We call the nodes in row 1 the *interface* of the grid. If $r = c$ then we also call $G_{r,c}$ an *r-grid* or grid of *order* r. It can be

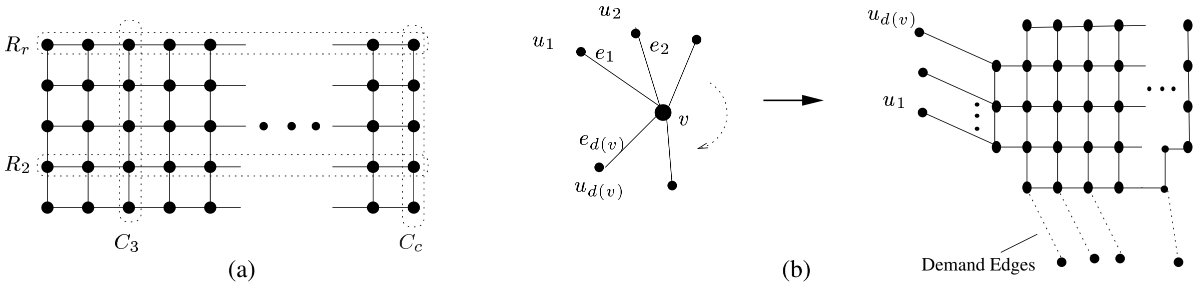

Figure 1. (a) An $r \times c$ Grid $G_{r,c}$ (b) Grid expansion at a node v

verified that a h-grid yields a h-crossbar with row 1 as the interface.

We now define an operation on a graph G that can be used to eliminate a high degree node. For each $v \in V$ we let G^{+v} denote the *grid expansion of G at v* to be the graph obtained by replacing the node v by a grid of order $h := d(v) + 2$ as follows. First, we delete the bottom left and right nodes, as well as the upper left nodes. Next, let $e_1, e_2, \dots, e_{d(v)}$ denote the edges incident to v and for each i, let $e_i = u_i v$. If G is planar, we also choose the edges to be in the clockwise rotation (for some embedding) order at v. We now delete v and introduce a grid G' of order h. We then add a new edge for each e_i. These edges are joined respectively from the u_i's to $(1, 2), (1, 3), \dots, (1, h-1)$ respectively – see Figure 1.

We also modify any demand graph $H = (V, F)$ associated with G as follows. If the node v was an endpoint of $k \leq d(v)$ demand edges, then we choose any k distinct nodes from the first row of the grid to be the new endpoints of the demand edges emanating from v. Let H^{+v} denote the resulting demand graph. One may define a mapping of fractional demands similarly. We also refer to the grid G' as *v's grid* in G^{+v}.

We let the *grid expansion* of G, denoted G^+, to be the graph obtained by applying this operation to every node in the graph. Clearly if G is planar, then so is G^+. For any demand graph $H = (V, F)$ associated with G, we also let H^+ denote the resulting demand graph in G^+. Also if Y was a set of terminals in G, we let Y^+ denote the terminals in H^+. Note that the demand edges of H^+ now form a matching in G^+. Clearly the order of applying the expansions is irrelevant, but we abuse notation in the sense that we also do not care how the assignments of terminals to demand edges was done. Clearly we may compute G^+ and H^+ in polytime.

An edge of G^+ is a *grid edge* if it is contained in the grid of some $v \in V(G)$. Any other edge is called a *real edge* and note that the real edges are in $1-1$ correspondence with $E(G)$. In the remainder we do not distinguish between real edges and their corresponding edge in G. We have the following.

Lemma 2.2 *If G satisfies the cut condition for some demand graph $R = (V, E(R))$, then G^+ satisfies the cut condition for R^+. In particular, if a set Y is well-linked in G, then Y^+ is well-linked in G^+. Moreover, any subset Q of edges of H is routable in G if Q^+ is routable in G^+. In addition, given a routing for Q^+ one may compute a routing for Q in polytime.*

2.3 Grid Minors

A minor in a graph G is a triple (H, Φ, E') where H is a graph, $E' \subseteq E(G)$ (the deleted edges) and $\Phi : V(H) \to \mathcal{P}(V(G))$ is a mapping such that (i) $\Phi(u) \cap \Phi(v) = \emptyset$ if $u \neq v$, (ii) $\Phi(u)$ induces a connected subgraph of $G - E'$ for each u, (iii) $E(H)$ is in $1-1$ correspondence with edges of G that remain after deleting E' and contracting each $\Phi(u)$ to a single node (and removing loop edges). Due to this correspondence we may speak of edges in H as being edges of G as well; if the context is clear, we denote by H_v the set of nodes $\Phi(v)$. Grid graphs are also significant for us due to the following fact.

Lemma 2.3 *If G has a grid minor of order h, then doubling the edges of G makes it an h-crossbar.*

Let H be a g-grid minor of a graph G. The *width* of a subset S of nodes of H is the smallest induced square subgrid of H that contains all nodes of S.

Lemma 2.4 *If S is a subset of H of width w and contains at most half the boundary of H, then $|\delta_H(S)| \geq w$.*

3 Finding a Large Integral Crossbar

In this section we prove the following.

Theorem 3.1 *Every k-well-linked planar graph G of maximum degree 4 contains a $\lfloor k/64 \rfloor - 2$-grid minor. Moreover, we may compute such a minor in polynomial time.*

We remark that one approach to finding such a large grid is to first deduce that a k-well-linked graph has large treewidth and to then apply the following result of Robertson, Seymour and Thomas [36]: every planar graph of branchwidth (which is within a constant factor of treewidth) at least $4g$ has a grid minor of size g. Although the branchwidth of planar graphs can be computed in polynomial time [38], there seems to be no polynomial time algorithms reported to actually obtain the grid minor. We give a direct algorithmic proof below for the special case we consider. Our proof is inspired by the concept of antipodality introduced in [38] and uses the same scheme for producing a grid, as found in [36].

Proof of Theorem 3.1. We assume that we have some fixed embedding of G on the sphere and that X is well-linked in G with $k = |X|$. We also fix some point ι of the sphere. A *G-contour* is a simple, closed curve in the plane, whose image only intersects the embedding of G at nodes. The *length* of such contour is the number of nodes whose embedding is contained in the image of the contour. We are interested in 'short' contours, namely those of length at most $\beta := \lfloor k/16 \rfloor$, whose images do not contain ι. For any such contour C, removing its image from the sphere produces two open regions (disks). We let $ins(C)$ (respectively $ext(C)$) denote the the region not containing ι (respectively containing ι), and we call these two regions *mates*. Without loss of generality, there is such a curve C whose length is 0 and whose interior contains the embedding of G. We call a curve *short* if its length is at most $\beta := \lfloor k/16 \rfloor$. We seek a short curve whose interior contains at least $k/2$ nodes of X and subject to this minimizes the number of edges and nodes of G that lie in its interior. Given the existence of C, there clearly exists such a curve, and moreover we may find it in polynomial time by greedily applying two operations to C. The first operation shifts the curve over an edge, and the second nudges the curve into a new node, thus increasing its length – see Figure 2.

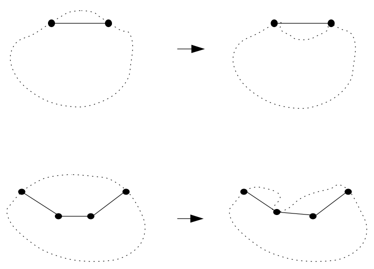

Figure 2. Shifting the curve.

Let C' denote the resulting curve and let S be the set of nodes that lie in the closure of the interior of C'. Since the length of C' is at most $k/16$ and since G is of maximum degree 4, we have that $|\delta(S)| < k/4$ (since at least one node on the boundary of C' is adjacent to some node in the interior of C'). It follows that $|S \cap X| > 3k/4$ by well-linkedness. From this we also deduce that the length of C' is exactly β. Otherwise, applying one of the shift operations increases the length of the curve by at most one, and clearly maintains at least $k/2$ terminals inside the curve.

We call a simple non-closed curve J a *cut* of C' if its image is contained in the closed interior of C' and only intersects the embedding of G at nodes, and also whose endpoints are distinct nodes a, b of G that lie on C'. Let U_1, U_2 denote the two subcurves of C' joining a, b. We call the two sets of nodes of G that lie on U_1, U_2 respectively, the *components* of the cut J. We claim that the length of J is at least the minimum of the lengths of the U_1 and U_2. If this is not the case, then the two curves C_1 and C_2 formed from U_1, U_2 and J, each has length at most β. The interior of at least one of these two curves, say C_1, contains more than $3k/8 - k/16 > k/4$ nodes from X. On the other hand, by minimality of C', the interior of C_1 does not contain $k/2$ such nodes, and hence the proper exterior of C_1 contains at least $k/2 - k/16 > k/4$ nodes. But then if S' is the set of nodes of G in the closed interior of C_1, we have that both S' and $V - S'$ contain more than $k/4$ nodes from X, yet $|\delta(S')| < k/4$, a contradiction.

Let G' denote the subgraph of G induced by S. We now follow the same argument as used in [36] for finding a grid minor. We have argued that for any cut J of C', its length is at least as large as the size of each of its components. In particular, this shows that for any two equal-sized disjoint subsets of nodes B_1, B_2 on the boundary of C', G' contains $|B_1|$ node-disjoint paths each joining a node of B_1 to a node of B_2. We now produce a γ-grid minor for $\gamma = \lfloor \beta/4 \rfloor$ as follows. Order the nodes on C' as $v_1, v_2, \ldots, v_\beta$ in say clockwise order. Partition these into 4 chunks of size γ: $B_i = \{v_j : i\gamma + 1 \leq j \leq (i+1)\gamma\}$ for $i = 0, 1, 2, 3$. We now find γ paths connecting the nodes of B_1 to the nodes of B_3. By planarity we actually have that node v_i is connected to to node v_{s+i} for each i (where $s = 2\gamma$). Then we find γ paths joining the nodes of B_2 to the nodes of B_4. Clearly the combined collection of paths yields the desired grid minor of size $\gamma \geq \lfloor k/64 \rfloor - 2$. □

4 Finding A Deletable Edge

In this section, we assume that our graph G has a grid minor H of order $k' \leq k$. The boundary of H refers to vertices in H that are in columns $1, k'$, and in rows $1, k'$. The interface I of the grid minor refers to the nodes in row 1. The goal of this section is to establish that if we cannot route a "large" number (at least $k'/17$) of the terminals to the interface of H, then there exists an edge inside the grid minor that can be deleted without altering the well-linked property of the set X. For clarity of exposition, we argue as though the vertices of the grid minor are singleton components. The general arguments follow along similar lines, but we defer the details to the longer version of the paper.

We begin by formalizing the notion of routing to the grid interface. Consider the following instance of the single source-single sink flow problem. Add 2 new nodes s, t to G such that s is adjacent to each node $v \in I$, and t is adjacent to each terminal in X. Let G^* denote this new network — all edge capacities are set to one. Consider any s-t mincut in G^*. If the cut size is at least $k'/17$, then we can route some subset of $k'/17$ terminals to the interface I of grid minor. In Section 5, we describe how this fact can be used to route $\Omega(|X|)$ pairs from any matching on X. In this section, we address the situation when such a flow does not exist:

Theorem 4.1 *If the maximum s-t flow in G^* is less than $k'/17$, then G contains an edge e such that X is well-linked in $G - e$.*

By standard uncrossing techniques, there is a unique minimal set C in G^* that induces a minimum s-t cut. That is, C contains s but not t and $c := |\delta_{G^*}(C)| < k'/17$. We now argue that there is the desired deletable edge of the grid minor H that is inside C. We call a set $S \subseteq V(G)$ *light* if $|S \cap X| \leq |X - S|$ and in addition $G[S]$ and $G[V - S]$ are connected. For our purposes, an edge e is *deletable*, if $|\delta_{G-e}(S)| \geq |S \cap X|$ for all light sets S. In the following, we sometimes abuse notation and say that a node v of a minor "contains" a node(s) of X. By this we mean that H_v (the nodes contracted to become v) contains a node(s) of X.

The general strategy is as follows. For the most part, it would be enough to find an edge deep inside the grid minor. One setback to this approach is that within the grid minor may be a small subgraph (or single node!) that contains most of the terminal nodes. Our first step is to identify such a set, so that later we can avoid it.

Lemma 4.2 *The graph induced by $\bar{C}$ contains a component L, that has at least $(k - c)$ terminals contained inside it.*

We now find a grid that avoids the subset L. An *i-ring* of a grid refers to the $(k - 2i) \times (k - 2i)$ subgrid obtained by discarding all nodes in rows and columns 1 through i, as well as $k - i + 1$ through k.

Lemma 4.3 *There is an induced subgrid H^* of H of order $k^* := (k' - c)/2$ that contains no node of L.*

Proof. Suppose first that some boundary node of H is contained in L (we may assume that each boundary node v, has $|H_v| = 1$). There could be at most c such nodes, since each such contributes an edge to the cut $\delta_{G^*}(C)$. It follows that the image of L in H does not intersect H's c-ring, for otherwise the grid itself contributes more than c edges to $\delta_{G^*}(C)$ – see Lemma 2.4. In this case, the c-ring of H is our desired subgrid. So now suppose that the image of L intersects H but not its boundary. Then since $G[L]$ is connected the image of L is contained inside H, where we use planarity. It follows that the width of this image is at most c, since otherwise there exists a set Q in the expansion of the grid with $|\delta_G(Q)| > c$ but $\delta_G(Q) \subseteq \delta_G(C)$, a contradiction. Thus there is a c by c subgrid of H that contains all nodes from L. Thus we can find a subgrid H^* of dimension at least $\frac{k'-c}{2}$ whose nodes are disjoint from L. □

From now on, we work with H^*. The following lemma is straightforward.

Lemma 4.4 *Let S be a subset of C, for which the images in H^* of both S and $\bar{S}$ intersect the boundary of H^*, and whose cut $\delta_G(S)$ contains an edge of the $2c$-ring of H^*. Then $\delta_G(S)$ contains more than $2c$ edges in the grid minor H^*.*

The following lemma shows that any edge in the $2c$-ring of H^* would be deletable but for the cuts induced by sets contained entirely in C.

Lemma 4.5 *Let $e = uv$ be an edge with both ends in C and both ends in the $2c$-ring of H^*. Then for any light set S with $S - C \neq \emptyset$, we have $|\delta_{G-e}(S)| \geq |S \cap X|$.*

Proof. Without loss of generality we may assume that $u \in S$. Let Z be the connected component of $G[S \cap C]$ that contains u. For ease of exposition we will assume that Z is the only connected component of $G[S \cap C]$, the proof extends easily to the case of multiple components. Let $x_1 = |Z \cap X|, x_2 = |S \cap X| - x_1$. We also let $\alpha := |[C - Z, Z]|, \beta := |[Z, S - Z]|, \gamma := |[V - (S \cup C), S - Z]|$ and $\delta_1 := |[S - Z, C - Z]|$ and $\delta_2 := |[Z, V - (S \cup C)]|$ – see Figure 4. (Here $[A, B]$ denotes the set of edges with one endpoint in A and one endpoint in B.) We include e in the count for α. Note that $S - Z$ satisfies the cut inequality and so $|\delta(S - Z)| = \beta + \delta_1 + \gamma \geq x_2$. Note also that $|\delta(S)| = |\delta(S - Z)| + \delta_2 + \alpha - \beta$ and hence S satisfies the cut inequality after deleting e as long as

$$\delta_2 + \alpha - \beta > x_1. \tag{1}$$

Suppose first that Z does not contain any boundary element from H: in particular it does not contain any element from the interface of H. Then $|\delta_{G^*}(C - Z)| = |\delta_{G^*}(C)| - x_1 - \beta - \delta_2 + \alpha$. So by minimality of C we must have that $-x_1 - \beta - \delta_2 + \alpha > 0$, and hence S is indeed fine.

So we may assume that Z contains some boundary nodes of H. Since e has its endpoints in the $2c$-ring of H^*, Z also contains some boundary nodes of H^* by planarity and since $G[Z]$ is connected. It follows by Lemma 4.4 that either $\delta_G(Z)$ contains more than $2c$ edges from H^*, or $H^* - Z$ identifies a subgraph of H^* disjoint from the

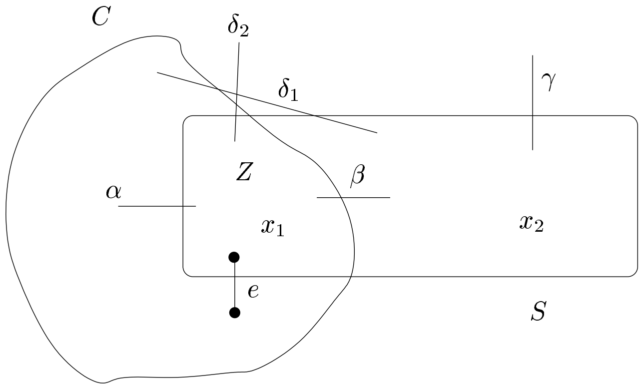

Figure 3. Illustration for Proof of Lemma 4.5.

boundary. In the first case, we may deduce that $\alpha > c$, (since $\delta_1 + \delta_2 + \beta < c$). Moreover, since C is a mincut, we have that $x_1 + \beta + \delta_1 + \delta_2 \le c$. In particular, $\alpha > \beta + x_1$ and so (1) again holds.

Thus $H^* - Z$ is a subgraph of H^* that does not include the boundary. But since $G[V - S]$ is connected and G is planar, we have that $V - S$ is itself a subset of $V(H^*)$. Since S is light we must have that $V - S$ contains at least half ($k/2$) of the nodes of X. Of these at most c are contained in C, and so $V(H^*)$ contains a node of L, a contradiction. □

This lemma tells us that as long as we remove an edge from the $2c$-ring of H^*, then the only cuts that could be ruined, are those induced by sets S contained in C. Thus we now focus on picking an edge in the $2c$-ring that does not affect any such cut.

Good Columns: We say a column i in H^* is *left-good* if for each $j \le i$, the number of terminals from X that are contained in columns j through i is at most $i - j$. Similarly, a column i is *right-good* if for each $j \ge i$, the number of terminals from X contained in columns i through j is at most $j - i$. A column i is *good* if (i) it is both left-good and right-good (ii) $i > 2c$, (iii) $i < k^* - 2c$, and (iv) it does not contain any nodes from $\bar{C}$.

The primary property of goodness that we use is the following. Let $H^*[S]$ be a connected subgraph containing a node of a good column, but no node of the boundary of H^*. Then if S has width w, it contains fewer than w terminals.

Lemma 4.6 *At least $k^* - 8c$ columns in the grid minor H^* are good.*

Proof. We will establish this via a simple marking scheme. Initially all columns are unmarked. We process columns from left to right. When a column i containing, say α, terminals from X is processed, we identify α unmarked columns that are closest to i on either side and mark them. More precisely, let $j_1 \le i$ be the largest integer such that the interval $[j_1, i]$ contains α unmarked columns ($j_1 = 1$ if no such integer exists) and let $j_2 > i$ be the smallest integer such that the interval $(i, j_2]$ contains α unmarked columns ($j_2 = k^*$ if no such integer exists). We mark all columns in these two intervals. Upon termination, the total number of marked columns is at most $2c$. Next, we mark all columns in the interval $[1, 2c]$ and $[k^* - 2c, k^*]$. Finally, consider the connected components of H^* induced by nodes containing elements of $\bar{C}$. For any component Q of this graph with width w, the grid contains at least w edges of $\delta(C)$. It follows that the total width of these subgraphs is at most $2c$. Thus marking any column that contains a node from $\bar{C}$ creates at most $2c$ more marked columns. Thus the total number of marked columns is bounded by $8c$.

We now claim that every unmarked column is good. Properties (ii) (iii) and (iv) are trivially satisfied. To see (i), fix an unmarked column i. To see that i is left-good, for any integer $j \le i$, consider the columns j through i. Let β denote the total number of terminals from X in these $(i - j + 1)$columns. It is easy to verify that these β terminals will generate at least β markings in the interval $[j, i]$. Thus if $\beta > (i - j)$, then column i must have been marked — a contradiction. We can establish that column i is right-good analogously. The lemma follows. □

We now find a deletable edge via the above result.

Lemma 4.7 *If $c < k'/17$, then there exists a deletable edge in the $2c$-ring of H^* whose endpoints lie in C.*

Proof. By Lemma 4.6 we have some good column, so we may choose an edge $e = uv$ in, say the middle of such a column. We claim that e is deletable. By Lemma 4.5, we need only consider a light set S that is contained in C. Without loss of generality $u \in S$. First, consider any subset S that does not contain a node from the boundary of H^*. Then since e is in a good column, we have that if the width of S (w.r.t H^*) is w, then $|S \cap X| < w$, and yet $|\delta_G(S)| \ge w$ by Lemma 2.4. Otherwise, S contains a boundary node, and so $|\delta_G(S)| \ge 2c$ edges by Lemma 4.4 and so S is fine since $|S \cap X| \le c$. □

5 Proofs of Theorems 1.1 and 1.3

Proof of Theorem 1.1. We follow the outline of the algorithm given in Section 1.1. Using Theorem 2.1 we can decompose the given instance $(G, \mathcal{T})$ into instances $(G_1, \mathcal{T}_1), (G_2, \mathcal{T}_2), \ldots, (G_\ell, \mathcal{T}_\ell)$ such that the terminal set X_i of $\mathcal{T}_i$ is well-linked in G_i. Further, $\sum_i |\mathcal{T}_i| = \Omega(\text{OPT}/\log^2 n \log\log n)$. From Theorem 1.3 we can route $|\mathcal{T}_i|/C$ pairs from $\mathcal{T}_i$ in G_i with congestion 2. Note that the graphs G_i are node and edge disjoint. Therefore it follows that we can route $\Omega(\text{OPT}/\log^2 n \log\log n)$ pairs with congestion 2 in G. □

Corollary 1.2 follows from Theorem 1.1 by using simple rounding and scaling ideas for packing problems. See [8] for details.

To prove Theorem 1.3 we need the following lemma on the routability between well-linked sets.

Lemma 5.1 *Let H_1 and H_2 be two well-linked sets in G. Suppose $A \subset H_1$ is routable to $B \subset H_2$ and $|A| \leq |H_1|/2$. Then given any $A' \subset H_1$ with $A' \leq |A|/2$, A' is routable to B. Moreover, given any $A' \subset H_1$ and $B' \subset H_2$ with $|A'| = |B'| \leq |A|/3$, A' is routable to B'.*

We note that Lemma 5.1 is tight.

Proof of Theorem 1.3. Let X be a well-linked set in G and let M be a matching on X. Let $k = |X|$. If $k < 150$, we may simply route a single demand, so suppose that $k \geq 150$. From Theorem 3.1 we can find an h-grid minor H in G where $h \geq \lfloor k/64 \rfloor - 2 \geq k/150$. From Theorem 4.1, if we cannot route $k/3000$ terminals from X to the interface of the grid, then we can find an edge e whose removal does not affect the well-linkedness of X in G. Hence we can repeat these steps until we find a grid such that we can route $k/3000$ terminals to its interface. Let X' be the set of terminals that can reach the interface of H. From Lemma 5.1 it follows that *any* set $X'' \subset X$ such that $|X''| \leq k/6000$ can route to the interface. Therefore, given any matching M on X of size $k/12000$, the end points of the M can be routed to the interface of H. Once the end points are routed to the interface, the pairs can be matched up using the grid minor with congestion 2, as shown in Lemma 2.3. The overall congestion is bounded by 3: one for the terminals to reach the interface and two for routing via the grid minor. Since M was an arbitrary matching of size $k/12000$, it follows that any matching on X can be routed with congestion 18000.

We now show that we can route $\Omega(|M|)$ pairs of M with congestion 2 instead of 3. We do this by routing the endpoints of the chosen edges to the interface of the grid without using any edges of the grid itself. We outline the idea below and omit full details in this version. We partition the grid into 4 subgrids of equal sizes H_1, H_2, H_3, H_4. This partitions the plane into 5 pieces - the interiors of the subgrids, and the rest. Simple counting shows that a set $M' \subset M$ with $|M'| = \Omega(|M|)$ has its endpoints in the exterior of one of the four subgrids, which we can without loss of generality assume to be H_1. Let X' be the set of end points of M'. We claim that the end points of $\Omega(M')$ pairs from M' can routed to the boundary of H_1. This follows from the fact that X' can be routed to the interface of H and the interface of H can be routed to the boundary of H_1 (note that both X' and the interface of H are well-linked and the boundary of H_1 is approximately well-linked). Since X' is in the exterior of H_1 we can route X' to the boundary of H_1 without using any edges that are in the interior of H_1. Although H_1 cannot route all the pairs that reach its boundary, it can route a constant fraction of them with congestion 2. The proof is similar to that of Lemma 2.3. Thus we can route $\Omega(|M|)$ pairs with congestion 2. □

6 Concluding Remarks

Our approach for planar graphs will extend to obtain a poly-logarithmic approximation for general graphs if we can constructively prove the following two claims: (i) given a graph G with a well-linked set of size k, G contains a cross-bar of size $\Omega(k/\text{polylog}(n))$, and (ii) if a large fraction of the well-linked set can not be routed to the interface of the crossbar, there exists a deletable edge in the crossbar. Recently, we have established property (ii) above [5]. However, our current proof of (ii) does not yield a polynomial-time algorithm to find such a deletable edge. We note that if (i) can be shown then our proof of (ii) would imply a poly-logarithmic integrality gap for the LP relaxation. Recently Demaine and Hajiaghayi [10] show that in graphs excluding a fixed minor, if the graph has treewidth w then it has a grid minor of size $\Omega(w)$. Thus our approach extends to graphs excluding a fixed minor.

For planar graphs we have an outline of an algorithm that improves the approximation ratios for Maximum EDP and related problems to $O(\log n)$ from the present $O(\log^2 n \log\log n)$. We obtain this by an improvement to Theorem 2.1 that also yields better ratios for the all-or-nothing multicommodity flow problem [4]. For planar graphs, we believe that we can eventually obtain an $O(1)$-approximation with $O(1)$ congestion.

Acknowledgments: This work is supported in part by an ONR basic research grant N00014-04-M-0042 to Bell Labs. We are grateful to Bruce Reed for sharing a preliminary draft of his book on Graph Minors.

References

[1] Y. Aumann and Y. Rabani. Improved Bounds for All Optical Routing. SODA 1995, pp. 567–576.

[2] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, February 1998.

[3] A. Baveja and A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Mathematics of Operations Research*, Vol. 25, pp. 255–280, 2000.

[4] C. Chekuri, S. Khanna, and F. B. Shepherd. The All-or-Nothing Multi-commodity Flow Problem. *Proc. of STOC*, June 2004.

[5] C. Chekuri, S. Khanna, and F. B. Shepherd. A deletable edge lemma for general graphs. *Manuscript*, 2004.

[6] C. Chekuri and S. Khanna. On Multidimensional Packing Problems. *SIAM J. on Computing*, 33(4):837–851, 2004.

[7] C. Chekuri and S. Khanna. Edge Disjoint Paths Revisited. *Proc. of SODA*, 2003.

[8] C. Chekuri, M. Mydlarz, and F. B. Shepherd. Multicommodity Demand Flow in a Tree and Packing Integer Programs. Submitted. Preliminary version in *Proc. of ICALP*, 2003.

[9] J. Chuzhoy and J. Naor. New inapproximability results for congestion minimization and machine scheduling. *Proc. of STOC*, 2004.

[10] E. Demaine and M. Hajiaghayi. Graphs Excluding a Fixed Minor have Grids as Large as Treewidth, with Combinatorial and Algorithmic Applications through Bidimensionality. To appear in *Proc. of SODA*, 2005.

[11] S. Even, A. Itai and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. on Computing*, Vol 5, 691-703, 1976.

[12] J. Fakcheroenphol and K. Talwar. An Improved Decomposition Theorem for Graphs Excluding a Fixed Minor. *Proc. of APPROX*, 2003.

[13] A. Frank. Edge-disjoint paths in planar graphs. *J. of Combinatorial Theory, Ser. B.*, No. 2 (1985), 164-178.

[14] A. Frank. Packing paths, cuts, and circuits - a survey. In B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, eds., *Paths, Flows and VLSI-Layout*, 49–100. Springer Verlag, Berlin, 1990.

[15] S. Fortune, J. Hopcroft and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111–121, 1980.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, 1979.

[17] N. Garg, V. Vazirani, M. Yannakakis. Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees. *Algorithmica*, 18(1):3-20, 1997.

[18] V. Guruswami, S. Khanna, R. Rajaraman, F. B. Shepherd, and M. Yannakakis. Near-Optimal Hardness Results and Approximation Algorithms for Edge-Disjoint Paths and Related Problems. *JCSS*, 67:473–496, 2003.

[19] C. Harrelson, K. Hildrum, and S. Rao. A polynomial-time tree decomposition to minimize congestion. *Proc. of SPAA*, 2003.

[20] R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, R. E. Miller, J. W. Thatcher, Eds., New York: Plenum Press, 1972, 85–103.

[21] J. M. Kleinberg and É. Tardos. Approximations for the disjoint paths problem in high-diameter planar networks. *Journal of Computer and System Sciences*, 57:61–73, 1998.

[22] J. M. Kleinberg and É. Tardos. Disjoint Paths in Densely Embedded Graphs. *Proc. of FOCS*, pp. 52–61, 1995.

[23] P. Klein, S. Plotkin and S. Rao. Planar graphs, multicommodity flow, and network decomposition. *Proc. of STOC*, 1993.

[24] J. M. Kleinberg. Approximation algorithms for disjoint paths problems. PhD thesis, MIT, Cambridge, MA, 1996.

[25] J. M. Kleinberg. Decision algorithms for unsplittable flow and the half-disjoint paths problem. *Proc. of STOC*, pp. 530–539, 1998.

[26] S. G. Kolliopoulos and C. Stein. Approximating Disjoint-Path Problems Using Greedy Algorithms and Packing Integer Programs. *Proc. of IPCO*, 1998.

[27] P. Kolman and S. Scheideler. Improved bounds for the unsplittable flow problem. In *Proc. of SODA*, 2002.

[28] M. R. Kramer and J. L. van Leeuwen. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. *VLSI-Theory* (F. P. Preparata ed.) [Advances in Computing Research, Volume 2], JAI Press, Greenwich, Connecticut, 1984, 129–146.

[29] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *JACM*, 46(6):787–832, 1999.

[30] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[31] J. F. Lynch. The equivalence of theorem proving and the interconnection problem. *ACM SIGDA Newsletter*, 5:3, 1975.

[32] H. Okamura and P.D. Seymour. Multicommodity flows in planar graphs. *Journal of Combinatorial Theory, Series B*, 31 (1981) 75–81.

[33] H. Räcke. Minimizing congestion in general networks. *Proc. of FOCS*, 2002.

[34] P. Raghavan and C. D. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[35] N. Robertson and P. D. Seymour. Outline of a disjoint paths algorithm. In B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, Eds., *Paths, Flows and VLSI-Layout*. Springer-Verlag, Berlin, 1990.

[36] N. Robertson, P. D. Seymour and R. Thomas. Quickly Excluding a Planar Graph. *Journal of Combinatorial Theory (B)*, 62: 323-348 (1994).

[37] N. Robertson and P. D. Seymour. Graph Minors X. Obstructions to Tree-Decomposition. *Journal of Combinatorial Theory (B)*, 52: 153-190 (1991).

[38] P. D. Seymour and R. Thomas. Call Routing and the Ratcatcher. *Combinatorica*, 14(2): 217-241 (1994).

[39] A. Schrijver. Combinatorial Optimization: Polyhedra and Efficieny. Springer-Verlag, Berlin Hiedelberg, 2003.

[40] A. Srinivasan. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. *Proc. of the FOCS*, pp. 416–425, 1997.

[41] K. Varadarajan and G. Venkataraman. Graph Decomposition and a Greedy Algorithm for Edge-disjoint Paths. *Proc. of SODA*, 2004.