A Note on Multiflows and Treewidth

Chandra Chekuri* Sanjeev Khanna† F. Bruce Shepherd‡

October 10, 2007

Abstract

We consider multicommodity flow problems in capacitated graphs where the treewidth of the underlying graph is bounded by r. The parameter r is allowed to be a function of the input size. An instance of the problem consists of a capacitated graph and a collection of terminal pairs. Each terminal pair has a non-negative demand that is to be routed between the nodes in the pair. A class of optimization problems is obtained when the goal is to route a maximum number of the pairs in the graph subject to the capacity constraints on the edges. Depending on whether routings are fractional, integral or unsplittable, three different versions are obtained; these are commonly referred to respectively as maximum MCF, EDP (the demands are further constrained to be one) and UFP. We obtain the following results in such graphs.

- An $O(r \log r \log n)$ approximation for EDP and UFP.
- The integrality gap of the multicommodity flow relaxation for EDP and UFP is $O(\min\{r \log n, \sqrt{n}\})$.

The integrality gap result above is essentially tight since there exist (planar) instances on which the gap is $\Omega(\min\{r, \sqrt{n}\})$. These results extend the rather limited number of graph classes that admit poly-logarithmic approximations for maximum EDP. Another related question is whether the cut-condition, a necessary condition for (fractionally) routing all pairs, is approximately sufficient. We show the following result in this context.

- The flow-cut gap for product multicommodity flow instances is $O(\log r)$. This was shown earlier by Rabinovich; we obtain a different proof.

1 Introduction

Let $G = (V, E)$ be an undirected graph with an integer valued *capacity* on the edges given by $c : E \rightarrow \mathcal{Z}^+$. A multicommodity flow instance in G is given by a non-negative symmetric demand matrix D. The demand for a node pair uv is given by $D(uv)$. Alternatively, D may be specified as the set of node pairs $s_1t_1, s_2t_2, \ldots, s_kt_k$ with positive demand values and in this setting we use d_i to denote the demand for pair s_it_i. A *product* multicommodity flow instance is a special case where the demand matrix D is specified by a node weight function $\pi : V \rightarrow \mathcal{R}^+$; the demand $D(uv)$ is set to be $\pi(u)\pi(v)$. We say that D is *routable with congestion* α in G if there exists a feasible multicommodity flow for D in G where the capacity of each edge is multiplied by a factor of α. For a set of nodes $S \subset V$, let $\delta_G(S)$ denote the set of edges with exactly one end point in S and let $c(S) = \sum_{e \in \delta(S)} c(e)$. Let $D(S) = \sum_{uv, u \in S, v \notin S} D(uv)$ denote the total demand between node pairs separated by S. A necessary condition for routability of D is the *cut condition*: for every set $S \subset V$, $c(S) \geq D(S)$. The *flow-cut gap* of G denoted by $\alpha(G)$ is the minimum $\alpha \geq 1$ such that every D that satisfies the cut condition is

*Dept. of Computer Science, University of Illinois, 201 N. Goodwin Ave, Urbana, IL 61801, USA. chekuri@cs.uiuc.edu.
†Dept. of CIS, Univ. of Pennsylvania, Philadelphia PA 19104, USA. sanjeev@cis.upenn.edu.
‡Dept. of Math. and Statistics, McGill University, Montreal, Quebec, H3A 2K6, Canada. bruce.shepherd@mcgill.ca.

routable in G with congestion α. The product flow-cut gap of G denoted by $\beta(G)$ is the minimum $\beta \geq 1$ such that every product multicommodity flow matrix that satisfies the cut condition is routable with congestion β. Leighton and Rao [28] showed in their seminal work that $\beta(G) = O(\log n)$ in general graphs. Subsequently it was shown in [29, 7] that $\alpha(G) = O(\log n)$. Both these bounds improve to $O(\log k)$ where k is the size of the support of D; we will utilize this result this in our work. It is also known that there are instances for which the above bounds are tight [28].

Given a multicommodity instance we are also interested in *integral* and *unsplittable* routings. A routing is integral for an integer valued matrix D if each flow path carries an integer amount of flow. A routing is unsplittable if for each pair the flow is sent along a single path. Note that if the values in D are in $\{0, 1\}$, then integral and unsplittable routings are the same; in this case the problem is the classical *edge disjoint paths* problem (EDP). We refer to the unsplittable flow problem by UFP. Checking the routability of a given demand matrix D is NP-hard if we require integrality or unsplittability. In addition to the feasibility question, the maximum EDP problem has also been widely studied. Here we seek algorithms that find (or approximate) the maximum *routable set*, that is, a set of terminal pairs that can be feasibly routed. We might also have weights on the pairs in which case we are interested in approximating the total weight of the pairs routed. The best approximation ratio for these problems in general graphs is $O(\sqrt{n})$ [12] obtained via the natural multicommodity flow based relaxation. It is also known that the integrality gap is at least this bad; in fact even in planar graphs the gap may be $\Omega(\sqrt{n})$ [17]. Henceforth, by EDP we refer only to the maximization version of the problem.

Multicommodity flow problems are of central importance in both theory and practice and there is a substantial amount of literature on these problems. One of the most interesting directions is whether the bounds on the flow-cut gap and/or the approximability of maximum EDP (and UFP) can be improved by exploiting properties of the underlying graph. An outstanding conjecture is that $\alpha(G) = O(1)$ for planar graphs and more generally [19] for minor-free graphs (which include graphs of bounded treewidth). It is known that $\beta(G) = O(1)$ in such graphs [21] and that $\alpha(G) = O(\sqrt{\log n})$ [33]. EDP was recently shown to be approximable to within $O(\log n)$ in planar graphs if congestion of 2 is allowed [10], and to within a constant if congestion 4 is allowed [13]. If all the degrees are even and the graph is planar, then an $O(\log^2 n)$ approximation is known [23].

1.1 Our Results

In this paper we consider graphs with treewidth bounded by some parameter r denote the treewidth of the given graph G. Informally speaking, treewidth of a graph G is r if for any subgraph G' of G, there is balanced node-separator of size r; that is, there is a set of r nodes whose removal results in connected components the largest of which has no more than two-thirds the number of nodes. We give a formal definition later. We remark that the capacities of the edges of G are not constrained by the treewidth bound r. It is known that many optimization problems can be solved exactly if the treewidth r is bounded by a fixed constant [1]. In contrast, EDP in capacitated trees is already APX-hard [17]. In this work we do not assume that r is a fixed constant and obtain approximation bounds that depend on r. The running time of our algorithms are polynomial in both n and r.

Our main results are the following.

- An $O(r \log r \log n)$ approximation for maximum EDP and UFP. We obtain a stronger bound of $O(r \log n)$ on the integrality gap.
- A proof that $\beta(G) = O(\log r)$ which differs from that of Rabinovich [31].

These results are obtained by decomposing into well-linked instances, a scheme previously explored in both solving EDP and flow-cut gap questions [10, 11]. Subsequent sections describe the relevant aspects of the scheme. We were inspired to apply these ideas to graphs with bounded treewidth while examining the work of Rabinovich [31] for such graphs. We mention that in some cases, the bounds derived can be strengthened by replacing the use of tree decompositions by that of branch decompositions for connectivity systems. The reader is referred to [18].

1.2 Related work and Discussion

The main tool employed today to understand flow-cut gaps is that of metric embeddings. This connection was first pointed out explicitly in [29, 7] and then explored in a number of papers. Earlier results for flow-cut gaps that have been obtained via network decomposition such as [28, 21] have been reinterpreted and strengthened using embeddings [29, 7, 33, 27, 31]. Further, new and improved approximation results for finding sparse cuts have been obtained [5, 6] by considering stronger relaxations for the cut problems. To prove bounds on the flow-cut gap, the embedding approach works with the metric induced by the dual of the concurrent flow relaxation. There is substantial earlier work surveyed in [16, 38] which explores conditions under which the flow-cut gap is 1, in other words when the cut condition implies feasibility of flow. Such results often require one to impose conditions on both the supply graph G and on the structure of the demands. A prototypical example of such a result is the well-known Okamura-Seymour theorem [30] that shows that the flow-cut gap is 1 if all terminals lie on a single face of a planar graph. Proofs of such results are typically shown via "primal" arguments: one starts with the cut condition and shows that a feasible flow can be constructed. This typically requires understanding the graph structure in more detail and this has several other benefits. In particular, this approach often yields additional properties on the structure of the flow such as integrality or half-integrality. On the other hand, the primal method seems difficult to adopt for cases where the flow-cut gap is larger than 1; here the embedding approach is flexible and allows one to use many powerful tools from geometry and combinatorics. We believe that it is useful and important to to explore the primal approach to prove flow-cut gap bounds. In [10], $\beta(G)$ is related to the congestion required to route flow for a matching on terminals in a well-linked set (see Section 2 for precise definitions). Using this equivalence and graph theoretic results on existence of suitably large minors [37], we have obtained primal proofs for the flow-cut gap $\beta(G)$ in planar graphs for both the edge capacitated case [10] as well as node-capacitated case [11]. In this paper we use similar ideas to study the relationship between $\beta(G)$ and graphs of bounded treewidth.

As we remarked, one of the motivations for the primal approach is to obtain algorithms for EDP and related routing problems. It is fair to say that, to date, the only general upper bound on an optimum solution comes from the multicommodity flow relaxation. Essentially all known algorithms are obtained by bounding the gap between the fractional and integer flows. The integrality gap of the flow relaxation is $\Omega(\sqrt{n})$ [17] and a matching upper bound was shown in [12]. However, allowing a constant congestion seems to dramatically improve the gap at least in planar graphs: for instance, allowing congestion 2 gives a bound of $O(\log n)$ [10] and congestion 4 gives a bound of $O(1)$ [13]. The classes of graphs for which a poly-logarithmic approximation is known for EDP with congestion 1 is limited. These include trees [17], graphs with large expansion (at least an inverse poly-logarithmic) [26], and even degree planar graphs [23]. Our bound of $O(r \log r \log n)$ sheds light on the role of treewidth on the integrality gap and also extends the class of graphs for which a poly-logarithmic approximation can be obtained. Our algorithm, when viewed appropriately, can be thought of extending the algorithm for trees. In terms of inapproximability of EDP, the following results are known. In general graphs, for any $\epsilon > 0$, there is no $\log^{1/2-\epsilon} n$-approximation unless $\mathrm{NP} \subseteq \mathrm{ZPTIME}(n^{\mathrm{polylog}(n)})$ [3, 4]. Moreover, under similar assumptions, there is no $\log^{O(1/c)} n$-approximation even if congestion c is allowed [4].

2 Preliminaries

We start with a brief review of some basic concepts and ideas that we will utilize in our algorithms.

2.1 Treewidth

We now formally define treewidth of a graph G, denoted by $tw(G)$. A *tree decomposition* of an undirected simple graph $G = (V, E)$ is a tree $T = (I, F)$, with a set $Z(i) \subseteq V$ associated with each node $i \in I$, such that the following conditions are satisfied.

- For each $v \in V$, there is an $i \in I$ with $v \in Z(i)$.
- For each edge $vw \in E$, there is an $i \in I$ with $v, w \in Z(i)$.
- For each $v \in V$, the set $\{i \in I \mid v \in Z(i)\}$ induces a (connected) subtree of T.

The *width* of the tree decomposition is $\max_{i \in I} |Z(i)| - 1$. The *treewidth* of G is the minimum width over all its tree decompositions. The -1 in the definition of width ensures that trees have treewidth 1. It is NP-hard to compute the treewidth of a graph, however for any fixed $k \geq 1$ there is a linear time algorithm to decide if the treewidth of a given graph is at most k [8]. The current best approximation algorithm [15] computes a tree decomposition of width $O(r\sqrt{\log r})$ where $r = tw(G)$. In this paper we do not need to explicitly compute the treewidth of a graph. We assume that we are given an upper bound r' on $tw(G)$ and the performance of our algorithms will be with respect to r'. If r' turns out to be an invalid upper bound, our algorithms might report an error in which case they provide evidence that $tw(G) > r'$. Thus, one can run the algorithms for each r in $1, 2, \ldots, |V| - 1$ and obtain a performance that is no worse than being given a true value of $tw(G)$; alternatively one can do a binary search for improved efficiency.

Given a graph $G = (V, E)$ and a node-subset $X \subseteq V$, we say that $A \subseteq V$ is a ρ-balanced node separator for X if $G[V \setminus A]$ has no connected component that has more than $\rho|X|$ nodes from X. The following is a standard observation on treewidth and separators.

Proposition 2.1 *Given a graph $G = (V, E)$ and any subset $X \subseteq V$, there is a $2/3$-balanced node separator for X containing at most $tw(G)$ nodes.*

2.2 Multicommodity Flow and LP Formulation for EDP

We restrict attention to multicommodity flow instances with 0-1 demands: an instance consists of an edge-capacitated graph $G = (V, E)$ with $c : E \rightarrow Z^+$ and k node pairs $s_1t_1, s_2t_2, \ldots, s_kt_k$ with $d_i = 1$ for $1 \leq i \leq k$. A terminal is a node that participates in one of the pairs and let X denote the terminal set. We will assume without loss of generality that each terminal occurs in exactly one pair thus the pairs are induced by a perfect matching M on X. In this *canonical* form an instance is given by the triple (G, X, M). We will also assume that each terminal has degree 1 in G.

An EDP instance consists of a triple (G, X, M) and the goal is to maximize the number of pairs that can be routed with edge-disjoint paths. For the given instance, we let $\mathcal{P}_i$ denote the set of 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 an optimal solution to the given instance. For each path $P \in \mathcal{P}$ we have a variable $f(P)$ which is the amount of flow sent on P. We let x_i denote the total flow sent on paths for pair i. We let $\bar{f}$ denote the flow vector with a component for each path P, and we denote by $|\bar{f}|$ the value $\sum_i x_i$. Then the LP relaxation is the following.

$$\begin{aligned}
\max \sum_{i=1}^{k} x_i \quad \text{s.t} & & & \\
x_i - \sum_{P \in \mathcal{P}_i} f(P) &= 0 & & 1 \le i \le k \\
\sum_{P : e \in P} f(P) &\le c(e) & & \forall e \in E \\
x_i, f(P) &\in [0,1] & & 1 \le i \le k, P \in \mathcal{P}
\end{aligned}$$

We work with the above exponential size path formulation for discussion sake, but there is an equivalent compact formulation that can be used for computational purposes. We use OPT denote the optimum value of the above relaxation on a given instance of EDP.

2.3 Well-linked Instances

A subset $X \subseteq V$ is *cut-well-linked* in G if for every $S \subset V$ we have that $c(S) \ge \min\{|S \cap X|, |(V \setminus S) \cap X|\}$. There is a stronger notion of flow-well-linkedness [11] that we do not need in this paper and hence we use the term well-linked when we mean cut-well-linked. An EDP instance (G, X, M) is a well-linked instance if X is well-linked in G. We rely on the following useful property of well-linked sets.

Lemma 2.2 *[11] Let X be a well-linked set of nodes in a graph G and let v be any node in G. Fix any subset $Y \subseteq X$. If v can be connected by edge-disjoint paths to every node in Y, then v can be connected by edge-disjoint paths to every node in any given subset $Z \subseteq X$ with $|Z| \le \lfloor |Y|/2 \rfloor$.*

2.4 Min-ratio Edge-Separators

Given a graph $G = (V, E)$, an *edge-separator* is a subset of edges of the form $\delta_G(S)$, where S is a nontrivial subset of V, and $\delta(S)$ are the edges with exactly one endpoint in S. We consider graphs with non-negative node-weights $w : V \to \mathcal{R}^+$ and non-negative edge-weights $c : E \to \mathcal{R}^+$. For an edge-separator $E' = \delta(S)$ we define the *ratio* of E' to be $c(E')/\min\{w(S), w(V \setminus S)\}$. A min-ratio edge-separator is an edge-separator with the minimum ratio. (If G is not connected, then there exists a separator whose ratio is zero.) Computing a min-ratio edge-separator is NP-hard. It is known that an $O(\beta(G))$ approximation can be obtained for min-ratio edge-separators by solving a multicommodity flow problem [28], however better approximation ratios can be obtained using stronger relaxations [5, 6].

2.5 Separators, Treewidth, and Well-linked Sets

Lemma 2.3 *Let X be a well-linked set of nodes in a graph G with $tw(G) \le r$, and $X \ge 6r$. Then there is a subset $A \subset V$ such that $|A| \le r$ and a set of edge disjoint paths $P_1, P_2, \ldots, P_h$ such that*

- *$h \ge |X|/3$,*
- *each path P_i intersects A and has as its end-points, two distinct nodes in X.*

- *each node in X is the end point of at most one path.*

Proof. From Proposition 2.1 there is a $2/3$-balanced node separator A for X of size at most r. That is, $|A| \leq r$ and there are two disjoint subsets $X_1, X_2 \subset X$ such that $|X|/3 \leq |X_1| \leq 2|X|/3$ and $|X|/3 \leq |X_2| \leq 2|X|/3$ and X_1 and X_2 are in separate components of $G[V \setminus A]$. Let $h = \min\{|X_1|, |X_2|\}$. From well-linkedness of X it follows that there are edge disjoint paths $P_1, P_2, \ldots, P_h$ that have one end point in X_1 and the other in X_2. Note that all these paths intersect A since A separates X_1 from X_2. These are the desired paths. □

Corollary 2.4 *Let X be a well-linked set of nodes in a graph G with $tw(G) \leq r$. Then there is a subset $A \subset V$ such that $|A| \leq r$ and there is a set of paths $\mathcal{P} = \{P_v \mid v \in X\}$ such that (i) P_v is a path from v to some node in A, and (ii) no edge e of G is in more than two paths of $\mathcal{P}$.*

Proof. Let A, X_1, X_2, and h be as in the Lemma 2.3. Assume without loss of generality that $h = |X_1|$. Since $|X_2| \leq 2h$ and X is well-linked, we can find two collections $\mathcal{P}_1$ and $\mathcal{P}_2$ of edge-disjoint paths from X_1 to X_2 such that each node in $X_1 \cup X_2$ is an end-point of at least one path in the collection. Note that an edge e may appear in both $\mathcal{P}_1$ and $\mathcal{P}_2$. Now any path $P \in \mathcal{P}_1 \cup \mathcal{P}_2$ can be partitioned into three segments such that (i) the first segment starts at a node in X_1 and ends when the path first meets a node $u_1 \in A$, (ii) the second segment starts at u_1 and ends at the last node from A that appears on the path, say u_2, and (iii) the third segment starts at u_2 and ends at a node in X_2. We add to $\mathcal{P}$ the first and the third segments as defined above for each path $P \in \mathcal{P}_1 \cup \mathcal{P}_2$. For each node $v \in A \cap X$, we add to $\mathcal{P}$ the trivial path consisting of node v alone. This gives us the desired collection of paths. If there are multiple paths in $\mathcal{P}$ representing a node v in X, we can discard them in an arbitrary manner until a single path P_v remains. □

3 An Overview of the Routing Framework

We now describe the underlying framework for our results. We follow the approach used in [9, 11, 10, 13] which has two steps. In the first step, using a fractional solution, the problem is decomposed into well-linked (in [13] they are not well-linked, but rather "well-behaved") instances without losing too much of the original fractional flow. In the second step, we route – integrally for EDP or fractionally for all-or-nothing flow – a poly-logarithmic (or constant in [13]) fraction of the remaining demands in each instance.

Specifically, we adopt the algorithm obtained in [11]. The algorithm in [11] takes an input instance (G, X, M) with an LP value OPT and obtains, in polynomial time, a collection of (node-disjoint) instances $(G_1, X_1, M_1), (G_2, X_2, M_2), \ldots, (G_h, X_h, M_h)$. The algorithm requires a sub-routine to compute approximate min-ratio edge separators. Let $\gamma(G)$ denote the approximation guarantee of the separator algorithm. Then the decomposition obtained has the following properties.

- For $1 \leq i \leq h$, $G_i = G[V_i]$ for some $V_i \subset V$, and $V_1, V_2, \ldots, V_h$ are disjoint.
- For $1 \leq i \leq h$, $M_i \subset M$.
- For $1 \leq i \leq h$, X_i is well-linked in G_i.
- $\sum_{i=1}^{h} |M_i| = \Omega(\frac{1}{\gamma(G) \log n}) \cdot \text{OPT}$.

Thus we obtain a reduction to well-linked instances while losing an $O(\gamma(G) \log n)$ factor in the approximation ratio. It follows from [31] that $\gamma(G) = O(\log r)$. If G satisfies additional properties, then we might have stronger bounds than $O(\log r)$. For example k-outerplanar graphs have treewidth $O(k)$ but by virtue of being planar, they have $\gamma(G) = O(1)$. We observe that the result above implies, by setting $\gamma(G) = 1$, that there *exists* a decomposition in which an $\Omega(1/\log n)$ factor of OPT is retained.

Following the decomposition, the goal is to route a large number of demands in a well-linked instance by exploiting properties of such instances. This is done via a *two-phase routing* as follows. First, in any well-linked instance we show the existence of a subgraph that behaves as a crossbar. In the case of fractional flows [9] this is a product multicommodity flow (based on Leighton-Rao [28]), and in the case of EDP (in planar graphs) this is a grid minor (based on Robertson-Seymour). In Phase 1 of the routing, we show that a large fraction of our terminals can route to the crossbar. In Phase 2, the appropriate terminals are matched up using the crossbar. The main contribution of the present paper is to show that the two-phase routing framework extends to EDP and MCF in graphs of bounded treewidth. The crossbar in this case is identified from within an appropriately chosen separator.

3.1 Two-Phase Routing

In this section, we formalize our notion of two-phase routing. Let D be a demand matrix in a graph G and let $f : V \to V$ be a mapping. We define a demand matrix D_f as follows:

$$D_f(xy) = \sum_{uv: f(u)=x, f(v)=y} D(uv).$$

In other words the demand $D(uv)$ for a pair of nodes uv is transferred in D_f to the pair $f(u)f(v)$. Thus the total demand transferred from u to $f(u)$ is $\sum_v D(uv)$. We define another demand matrix D'_f which essentially asks that each node u can transfer this amount of flow to $f(u)$.

$$D'_f(uf(u)) = \sum_v D(uv).$$

The matrix D'_f represents the first-phase of the routing whereby each node u distributes its demand to $f(u)$. This is to be interpreted as saying that the node u is routed to the node $f(u)$ of the crossbar. The matrix D_f, on the other hand, represents the demand that needs to be routed by the crossbar in the second-phase. The following elementary proposition shows a correspondence between the cut condition for the routing in the two phases and the cut condition for the original demand matrix.

Proposition 3.1 *If D'_f is routable in G with congestion a, and D_f is routable in G with congestion b, then D is routable with congestion $a + b$ in G.*

We need a cut condition given by the lemma below. For any $x \in \mathcal{R}^+$ and graph G, we denote by xG, the graph obtained by multiplying the capacity of each edge of G by x.

Lemma 3.2 *Let D be a demand matrix on a given graph G and let $f : V \to V$ be a mapping. If the cut condition is satisfied for D, and D'_f is routable in γG, then the cut condition is satisfied for D_f in $(\gamma + 1) \cdot G$.*

Proof. Consider a cut $\delta(S)$ in G for some $S \subset V$. Recall that $c(S)$ is the total capacity of the edges in $\delta(S)$. Since D satisfies the cut condition $c(S) \geq D(S)$ for all $S \subset V$. Also, since D'_f is routable in γG, $\gamma c(S) \geq D'_f(S)$ for all $S \subset V$.

To prove the lemma we need to show that $(\gamma+1)c(S) \geq D_f(S)$ for all $S \subset V$. From the above inequalities on $c(S)$, it suffices to show that $D_f(S) \leq D(S) + D'_f(S)$. Let X_S denote the set of all unordered pairs of nodes

uv such that u and v are separated by S, that is $\{s,t\} \cap S = 1$. We can write $D_f(S)$ as $\sum_{uv: f(u)f(v) \in X_S} D(uv)$. For each pair uv such that $f(u)f(v) \in X_S$, we charge $D(uv)$ to either $D(S)$ or $D'_f(S)$ such that there is no overcharge. This will complete the argument.

We consider two cases. If $uv \in X_S$ then we charge $D(uv)$ to $D(S)$. Note that $\sum_{uv \in X_S} D(uv) = D(S)$ and hence we do not over charge $D(S)$. If $uv \notin X_S$ then either $uf(u) \in X_S$ or $vf(v) \in X_S$ but not both. In $uf(u) \in X_S$ we charge $D(uv)$ to u, otherwise to v. We observe that the total charge to a node u is at most $D'_f(uf(u)$ and it is charged only if $uf(u) \in X_S$. Hence the total charge to $D'_f(S)$ is not exceeded either. □

4 Approximation algorithms for EDP and UFP

In this section we prove the $O(r \log r \log n)$ approximation for EDP and UFP. The approximation is based on the multicommodity flow relaxation described in Section 2. Using by now standard ideas [25] an algorithm for EDP based on the flow relaxation can be extended to obtain an algorithm with a comparable performance (to within constant factors) for UFP if $d_{\max} \leq c_{\min}$. Thus we focus on EDP.

Theorem 4.1 *Let (G, X, M) be a well-linked instance of EDP and let $tw(G) \leq r$. Then any subset of $|M|/(6r)$ pairs in M can be routed edge-disjointly in G. Moreover, given a well-linked instance (G, X, M) and an integer $r' \geq 1$, there is a polynomial time algorithm that either routes $|M|/(6r')$ pairs in M edge-disjointly or correctly claims that $tw(G) > r'$.*

Proof. We prove the first part of the theorem assuming knowledge of a valid upper bound r on $tw(G)$. If $|M| \leq 6r$, it suffices to route any pair in M (note that since X is well-linked, all vertices in X must belong to the same connected component in G). Let A be the separator as in Lemma 2.3, and consider the collection $\mathcal{P}$ of $|X|/3$ edge-disjoint paths that go through A such that end-points of each path are distinct nodes in X. By Pigeonhole principle, there exists a node v such that $|X|/(3r)$ paths in $\mathcal{P}$ go through v. In other words, there is a subset $Y \subseteq X$ of size $2|X|/(3r)$ that can reach v via edge-disjoint paths. By Lemma 2.2, it follows that any subset $Z \subseteq X$ of size $|X|/(3r)$ can reach v via edge-disjoint paths. In particular, we can choose an arbitrary set of $|M|/(6r)$ edges in M and route their end-points to v in an edge-disjoint manner. We thus get an edge-disjoint routing for the chosen $|M|/(6r)$ pairs in M, as claimed.

Now suppose we are given an integer parameter r' which might not be a valid upper bound on $tw(G)$. We pick an arbitrary set of $|M|/(6r')$ pairs from M and let Z be the set of end points of these pairs. For each node $u \in V(G)$ we solve single-source maximum flow problem from the set Z to u. If the flow value for any node is $|Z|$ we have found the desired paths. Otherwise, from the analysis above, we obtain a contradiction to the fact that $tw(G) \leq r'$. □

Theorem 4.2 *EDP in graphs with treewidth at most r can be approximated to within a ratio of $O(r \log r \log n)$. The integrality gap for the multicommodity flow relaxation is $O(r \log n)$.*

Proof. We solve the flow relaxation. Let OPT denote the value of the flow relaxation. Using the decomposition described in Section 3, we create a collection of well-linked instances, say (G_1, X_1, M_1), (G_2, X_2, M_2), …, (G_h, X_h, M_h), with $\sum_{i=1}^{h} |M_i| \geq \text{OPT}/(c \log r \log n)$ for some constant c. Moreover, $tw(G_i) \leq tw(G)$ for $1 \leq i \leq h$ since G_i is a subgraph of G. By Theorem 4.1, we can route in each G_i $|M_i|/(6r)$ pairs in M_i. Summing over all h instances, we get the desired bound. Note that we get the improved bound on the integrality gap by the same reasoning but using an optimum separator algorithm in the decomposition. □

Theorem 4.3 *The integrality gap of the multicommodity flow relaxation for EDP in graphs with treewidth at most r is $O(\min\{r \log n, \sqrt{n}\})$. Moreover, the integrality gap is $\Omega(\min\{r, \sqrt{n}\})$ even in planar graphs.*

Proof. The upper bound on the integrality gap follows by combining the bound of $O(r \log n)$ from Theorem 4.2 with the $O(\sqrt{n})$ shown in [12]. For the lower bound, we note that [17] establishes an $\Omega(\sqrt{n})$ lower bound on the integrality gap on a planar grid-like graph of treewidth $\Theta(\sqrt{n})$. For any $r \leq \sqrt{n}$, this family of instances can be scaled to a graph on $O(r^2)$ vertices with treewidth r and an integrality gap of $\Omega(r)$. □

5 Product Multicommodity Flow-Cut Gap

In [10] it was observed that proving an upper bound on $\beta(G)$ is equivalent to establishing an upper bound on the worst case congestion required to route an arbitrary matching M on a well-linked set X in G. Thus it suffices to prove the following theorem.

Theorem 5.1 *Let X be a well-linked set in G. If $tw(G) \leq r$ then any matching M on X can be routed fractionally in G with congestion $O(\log r)$.*

Proof. Let D be the demand matrix induced by the matching M on X, that is $D(uv) = 1$ if and only if uv is an edge of M. Since X is well-linked the cut condition is satisfied for D. Let $A \subset V$ be the set of at most r nodes with the properties guaranteed by Corollary 2.4; let P_v be the path from $v \in X$ to a node in A. Using these paths we define a mapping $f : X \rightarrow A$ where $f(v) = x$ where x is the end point of P_v in A. By construction we have that D'_f has a feasible routing in G with congestion 2. Thus by Lemma 3.2, the cut condition is satisfied for the demand matrix D_f in the graph $3 \cdot G$. We now observe that $D_f(uv) > 0$ only if $u, v \in A$. Therefore, the number of non-zero values in D_f is $O(r^2)$. It follows from the flow-cut gap results of [29, 7] that D_f is routable in G with congestion $O(\log r)$. Since D'_f is routable with congestion 2 and D_f is routable with congestion $O(\log r)$, D is routable with congestion $O(\log r)$. □

Flow-cut gap results have been motivated by and applied to find approximation algorithms for separator problems and their applications – see [28] for several examples. Gap results via embedding methods are typically shown via constructive algorithms that obtain a separator from a metric relaxation that is dual to the multicommodity flow problem. The above proof can be converted into an algorithm that finds in polynomial time an $O(\log r)$ approximation for finding min-ratio edge-separators in graphs of treewidth $O(r)$. However, as we mentioned earlier, such an algorithm can be derived from [31] and hence we omit the details of the algorithm.

6 Conclusions

We used the primal method and well-linked decomposition to obtain results on multiflow problems. The results are parametrized by the treewidth of the graph and are meaningful when it is not too large. For EDP we obtained an $O(r \log r \log n)$ approximation ratio. However, we prove a better bound of $O(r \log n)$ on the integrality gap of the flow relaxation — this bound is existential. It would be interesting to remove this discrepancy and obtain an approximation ratio that matches the integrality gap. Further, the known lower bound on the integrality gap is only $\Omega(r)$. Can a matching upper bound be shown?

As we mentioned already, it is conjectured [19] that the flow-cut gap for multicommodity flows in graphs of bounded treewidth is $O(1)$. The conjecture in fact states that the gap is $O(1)$ for any class of graphs that excludes a fixed minor. The conjecture is not known to be true even for graphs of bounded pathwidth [34]. In some recent work, Indyk and Sidiropoulos [20] showed that the flow-cut gap in graphs of bounded genus is $O(1)$ if the gap is $O(1)$ in planar graphs. Can this equivalence can be extended to minor-free graphs? If so, it suffices to prove the conjecture for planar graphs. One bottleneck [39] to showing such an equivalence are graphs of bounded pathwidth; path decompositions arise in Robertson and Seymour's structural characterization of minor-free graphs in terms of bounded genus graphs [35]. For these reasons, a fruitful intermediate question to pursue is whether the flow-cut gap is $O(1)$ in graphs of bounded pathwidth.

Acknowledgments. This work was done while Chandra Chekuri and F. Bruce Shepherd were at Lucent Bell Labs, Murray Hill, NJ 07974, USA; they acknowledge support from an ONR basic research grant N00014-05-1-0256 to Lucent Bell Labs. Sanjeev Khanna acknowledges support from an NSF Career Award CCR-0093117 and NSF Award CCF-0635084.

References

[1] K. Abrahamson and M. Fellows. Finite Automata, Bounded Treewidth and Well-Quasiordering, in: N. Robertson and P. Seymour (editors), Graph Structure Theory: Proceedings of the Joint Summer Research Conference on Graph Minors, Seattle, June, 1991, American Mathematical Society, Contemporary Mathematics 147 (1993), 539–564

[2] E. Amir. Efficient Approximation for Triangulation of Minimum Treewidth. *Proc. of the 17th Conference on Uncertainty in Artificial Intelligence (UAI '01)*, 2001.

[3] M. Andrews and L. Zhang. Logarithmic hardness of the undirected edge-disjoint paths problem. JACM, 53(5):745–761, 2006. Preliminary version in *Proc. of ACM STOC*, 276–283, 2005.

[4] M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang. Hardness of the Undirected Edge-Disjoint Paths Problem with Congestion. *Proc. of IEEE FOCS*, 226–244, 2005.

[5] S. Arora, S. Rao and U. Vazirani. Expander Flows, Geometric Embeddings, and Graph Partitionings. *Proc. of ACM STOC*, 222–231, 2004.

[6] S. Arora, J. R. Lee, and A. Naor. Euclidean distortion and the sparsest cut. *Proc. of ACM STOC*, 553–562, 2005.

[7] 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.

[8] H.L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. on Computing*, vol 25, 1305–1317, 1996. Preliminary version in *Proc. ACM STOC*, 1993.

[9] C. Chekuri, S. Khanna, and F. B. Shepherd. The All-or-Nothing Multicommodity Flow Problem. *Proc. of ACM STOC*, 156–165, 2004.

[10] C. Chekuri, S. Khanna, and F. B. Shepherd. Edge Disjoint Paths in Planar Graphs. *Proc. of IEEE FOCS*, 71–80, 2004.

[11] C. Chekuri, S. Khanna, and F. B. Shepherd. Multicommodity flow, well-linked terminals, and routing problems. *Proc. of ACM STOC*, 183–192, 2005.

[12] C. Chekuri, S. Khanna, and F. B. Shepherd. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory of Computing*, Vol 2, 137–146, 2006.

[13] C. Chekuri, S. Khanna, and F. B. Shepherd. Edge-Disjoint Paths in Planar Graphs with Constant Congestion. *Proc. of ACM STOC*, 757–766, 2006.

[14] C. Chekuri, M. Mydlarz, and F. B. Shepherd. Multicommodity Demand Flow in a Tree and Packing Integer Programs. *ACM Transactions on Algorithms*, Vol 3, 2007. Preliminary version in *Proc. of ICALP*, 2003.

[15] U. Feige, M. Hajiaghayi and J. Lee. Improved approximation algorithms for minimum-weight vertex separators. *SIAM J. on Computing*, to appear. Preliminary version in *Proc. of ACM STOC*, 563–572, 2005.

[16] 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.

[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] J. Geelen, B. Gerards and G. Whittle. Tangles, tree-decompositions, and grids in matroids. To appear.

[19] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Cuts, trees and ℓ_1-embeddings of graphs. *Combinatorica*, Vol 24, 233–269, 2004. Preliminary version in *Proc. of IEEE FOCS*, 1999..

[20] P. Indyk and A. Sidiropoulos. Probabilistic embeddings of bounded genus graphs into planar graphs. *Proc. of ACM SoCG*, 204–209, 2007.

[21] P. Klein, S. Plotkin and S. Rao. Planar graphs, multicommodity flow, and network decomposition. *Proc. of STOC*, 682–690, 1993.

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

[23] J. M. Kleinberg. An Approximation Algorithm for the Disjoint Paths Problem in Even-Degree Planar Graphs. *Proc. of FOCS*, 627–636, 2005.

[24] S. G. Kolliopoulos. Edge Disjoint Paths and Unsplittable Flow. Chapter in *Handbook on Approximation Algorithms and Metaheuristics*, Chapman Hall/CRC Press, 2007.

[25] S. G. Kolliopoulos and C. Stein. Approximating Disjoint-Path Problems Using Greedy Algorithms and Packing Integer Programs. *Math. Programming A*, (99):63–87, 2004. Preliminary version in *Proc. of IPCO*, 1998.

[26] P. Kolman and C. Scheideler. Improved bounds for the unsplittable flow problem. In *Proc. of ACM-SIAM SODA*, 184–193, 2002.

[27] R. Krauthgamer, J. Lee, M. Mendel and A. Naor. Measured descent: A new embedding method for finite metrics. *Geometric and Functional Analysis*, 15(4):839–858, 2005. Preliminary version in Proc. of FOCS 2004.

[28] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *JACM*, 46(6):787–832, 1999. Preliminary version in *Proc. of FOCS*, 1988.

[29] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. Preliminary version in *Proc. of FOCS*, 1994.

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

[31] Y. Rabinovich. On average distortion of embedding metrics into the line and into ℓ_1. *Proc. of ACM STOC*, 456–462, 2003.

[32] Y. Rabinovich. Personal communication, February 2006.

[33] S. Rao. Small distortion and volume preserving embeddings for planar and Euclidean metrics. *Proc. of ACM SoCG*, 300–306, 1999.

[34] N. Robertson and P. D. Seymour. Graph Minors. I. Excluding a forest. *J. of Combin. Theory Ser B*, 35(1):39–61, 1983.

[35] N. Robertson and P. D. Seymour. Graph Minors. XVI. Excluding a non-planar graph. *J. of Combin. Theory Ser B*, Vol 89, 43–76, 2003.

[36] 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.

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

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

[39] A. Sidiropoulos. Personal communication, November 2006.