Approximation Algorithms for the Edge-Disjoint Paths Problem via Räcke Decompositions

Matthew Andrews
Alcatel-Lucent Bell Labs
Murray Hill, NJ
andrews@research.bell-labs.com

***Abstract*—We study the Edge-Disjoint Paths with Congestion (EDPwC) problem in undirected networks in which we must integrally route a set of demands without causing large congestion on an edge. We present a $(polylog(n), poly(\log\log n))$-approximation, which means that if there exists a solution that routes X demands integrally on edge-disjoint paths (i.e. with congestion 1), then the approximation algorithm can route $X/polylog(n)$ demands with congestion $poly(\log\log n)$.**

The best previous result for this problem was a $(n^{1/\beta}, \beta)$-approximation for $\beta < \log n$.

***Keywords*-Edge-disjoint paths. Approximation algorithms.**

I. Introduction

In this paper we study the integrality gap of the Edge-Disjoint Paths with Congestion problem (EDPwC) in undirected graphs. We are given an undirected graph $G = (V, E)$ and a set of terminals $(s_1, t_1), \ldots, (s_k, t_k)$. The aim is to connect as many of these terminals as possible on edge-disjoint paths (i.e. paths that do not share any edges). In evaluating any solution we allow for the possibility that some edge may have a small number of paths routed through it.

This is one of the original NP-hard problems and has many applications to routing traffic in communication networks. Hence the approximability of the problem has generated a lot of interest. In general there are two metrics we can use regarding the approximability of the problem. The first is the number of terminals that are connected. The second metric is the congestion on any edge.

More generally, suppose that the instance has n nodes, m edges and k demands. We write the EDPwC problem as,

$$\max \quad \sum_i x_i$$

subject to (1)

$$\begin{aligned} \sum_i y_i^{uv} &\leq 1 \qquad \forall u, v \\ \sum_v (y_i^{uv} - y_i^{vu}) &= \begin{cases} x_i & u = s_i \\ -x_i & u = t_i \quad \forall i, u \\ 0 & \text{o.w.} \end{cases} \\ x_i, y_i^{uv} &\in \{0, 1\}, \end{aligned}$$

and denote the value of the optimal solution by OPT. Later on we shall also refer to the natural linear relaxation in which we replace the last constraint with $x_i, y_i^{uv} \in [0, 1]$. We say that a solution to the integer program (1) is an (α, β)-approximation if $\sum_i x_i \geq OPT/\alpha$ and $\sum_i y_i^{uv} \leq \beta \; \forall u, v$. In other words, we have an α-approximation for the amount of demand routed and each edge experiences congestion β. It is well-known since the celebrated result of Raghavan and Thompson [1] that if we *randomly round* any solution, (i.e. we perform a flow decomposition of the resulting flow paths and treat the resulting fractional weights as probabilities) then we can obtain a $(O(1), O(\log n/\log\log n))$-approximation where n is the size of the instance. The main focus of the current paper is to determine what values of α can be achieved when the congestion parameter β is much smaller than $\log n$, in particular when it is of the form $poly(\log\log n)$.

Due to the fundamental nature of the Edge-Disjoint Paths problem it has been the subject of a great deal of study. We now list a number of results that are known about the problem, many of which we shall use in our analysis. Some of these results refer to *expander graphs*. We say that a graph is a γ-expander if for all $S \subseteq V$ such that $|S| \leq |V|/2$, cap$(S, V - S) \geq \gamma|S|$, where cap$(A, B)$ denotes the number of edges between node set A and node set B.

A. Known Results on EDP

Suppose the instance has n nodes, m edges and k demands.

- Karp showed in [2] that it is NP-hard to decide whether or not all of the demands can be routed on edge-disjoint paths. However, for fixed k this decision problem is polynomially solvable in undirected graphs [3]. (For directed graphs the problem is NP-hard even for $k = 2$ [4].)
- For the case $\beta = 1$, there is an $(O(m^{1/2}), 1)$-approximation for both undirected and directed graphs [5], [6], [7]. For dense networks this was improved to $(O(n^{1/2}), 1)$ for the undirected case [8] and $(O(n^{2/3}, 1)$ for the directed case [9], [10].
- If β can be larger than 1, [1] showed that if we do a path decomposition of the solution to the linear relaxation and pick each path with a probability equal to its fractional weight, then we obtain an

0272-5428/10 $26.00 © 2010 IEEE
DOI 10.1109/FOCS.2010.33

$(O(1), O(\log n/\log\log n)$-approximation. For general values of β between 1 and $\log n$, [11], [12], [7] show how to obtain an $(O(m^{1/\beta}), \beta)$-approximation.

- If all the flow paths in the flow decomposition of the fractional solution have length at most d, a "folklore" result states that by using randomized rounding in combination with the Lovász Local Lemma, we can obtain a $(1, O(\log d))$ approximation.
- In terms of negative approximation results, [13] showed that unless P=NP there is no $(\Omega(m^{\frac{1}{2}-\varepsilon}), 1)$-approximation in directed graphs. For general β, the sequence of papers [14], [15], [16], [17], [18] showed that unless NP has randomized quasipolynomial time algorithms there is no $(n^{\Omega(\frac{1}{\beta})}, \beta)$-approximation in directed graphs for $\beta = o(\log n/\log\log n)$ and no $(\Omega(\log^{\frac{1}{\beta+1}-\varepsilon} n), \beta)$-approximation in undirected graphs for $\beta = o(\log\log n/\log\log\log n)$.[1]
- If the graph has good expansion properties then better results are possible. For example, in [19] it is shown that for any graph with expansion factor $\gamma \geq 1$ and a constant degree bound, there is some constant κ such that *any* set of $n/\log^{\kappa} n$ demands can be routed on edge-disjoint paths. (In particular this implies an $(O(\log^{\kappa} n), 1)$-approximation algorithm.) Other papers that study EDPwC in expanders include (e.g. [20], [21], [22]).
- Rao and Zhou showed in [23] then if the minimum cut in the graph is large then we do not need good expansion properties. In particular they showed that if the minimum cut in the graph has size $\Omega(\log^5 n)$ then we can always embed an expander into the graph and thereby obtain a $(polylog(n), 1)$-approximation algorithm.
- If the graph is planar then good approximation algorithms are also possible. In [24] it was shown that there is a $(polylog(n), O(1))$-approximation for this case.
- The paper [25] presented a $(O(\log^2 n), 1)$-approximation for a related problem known as the *All-or-Nothing Multicommodity Flow Problem*. In this problem we are allowed to route demands fractionally. However, we only get credit if we route all of a demand. In other words the problem is equivalent to (1) except that we relax the constraint $y_i^{uv} \in \{0, 1\}$ to $y_i^{uv} \in [0, 1]$.

Note that from the above results, if we wish for an approximation algorithm with performance bound $(polylog(n), \beta)$ then the best current result is the randomized rounding result of [1] which requires that $\beta = \Omega(\log n/\log\log n)$. In this paper we show that this bound can be improved significantly.

[1]For some applications it is convenient to have a reduction with *perfect completeness*. In our setting this translates into a requirement that in good solutions (i.e. "yes"-instances) all demands can be routed. In this case the hardness result for undirected graphs becomes $(\Omega(\log^{\frac{1}{\beta+2}-\varepsilon} n), \beta)$.

Theorem 1: There exists a randomized $(O(\log^{61}(n)), O((\log\log n)^6))$-approximation for the EDPwC problem.

We remark that due to the $(n^{\Omega(\frac{1}{\beta})}, \beta)$ hardness result of [18], Theorem 1 implies that EDPwC has a strictly better approximation factor in undirected graphs than it does in directed graphs.

We structure the paper as follows. In Section II we give a very high level overview of some previous work that we need in order to obtain our results. In Section III we state a few additional facts that we need to obtain our results. In Section IV we give a rough sketch of the proof of Theorem 1. In Section V we fill out the details for the case in which the network has a special structure. For the general case, due to space restrictions we defer some of the analysis to the full version of the paper.

II. Techniques from Previous Work

In this section we briefly describe a number of known results that we make use of in our analysis.

A. The Räcke decomposition

Räcke introduced his decomposition in [26] in the context of oblivious routing, i.e. the selection of a flow for each possible terminal-pair such that regardless of which terminal pairs actually need to be connected, if they can be feasibly routed with congestion 1, then this oblivious routing has congestion $O(\log^3 n)$. The main idea is to create a hierarchical clustering of the graph and then route any terminal pair up the hierarchy until they reach a common cluster.

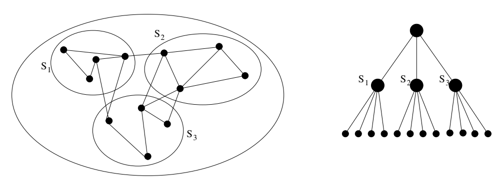

Figure 1. A Räcke decomposition and its associated tree.

As part of his analysis Räcke presented a method for decomposing graphs that has been useful in other routing contexts (e.g. [25]). We shall base our analysis on this decomposition. In particular, Räcke showed that we can decompose any undirected graph G into a laminar system such that each set in the system corresponds to a node in a tree. (See Figure 1.) At the top level there is a *level-0* root node v in the tree that corresponds to the whole graph. Two sets S_{u_t} and S_{v_t} in the laminar family corresponding to nodes u_t and v_t in the tree are connected in the tree if and only if one set is a subset of the other and there are no other sets in the laminar family that lie between them. We say that a set in the laminar family is at level ℓ if the corresponding node is at level ℓ in the tree. We shall use the following important properties of the Räcke decomposition.

Let S be a level ℓ cluster and let $S_1, S_2 \ldots$ be its level $\ell+1$ child clusters. Let U be a connected subset of S. We denote by $w_S(U)$ or $w_{\ell+1}(U)$ the number of edges in S that have at least one endpoint in U and which do not have *both* endpoints in the same child subcluster. Let $out(U)$ equal the number of edges with one endpoint in U. (We refer to these edges as *outedges* of U. We shall sometimes abuse notation and also use $out(U)$ to denote the set of such edges.)

Räcke showed that for some parameter $h = O(\log n)$, there exists a hierarchy of clusters with at most h levels such that if $|U| \leq |S|/2$ then

$$w_{\ell+1}(U) \leq \text{cap}(U, S-U)/h \quad (2)$$
$$out(U) \leq \text{cap}(U, S-U)/h. \quad (3)$$

The original presentation of Räcke was not constructive. (It required the optimal solutions to NP-hard problems as subroutines.) However, it was soon followed by constructive solutions using a similar approach (e.g. [27], [28]). We shall make use of the construction of Bienkowski et al. [28] since although it gives weaker bounds than the construction of [27] it is simpler to adapt for our purposes. In particular for any cluster S they defined a concurrent multicommodity flow (CMCF) problem such that for all $u, v \in S$ the demand between u and v is given by,

$$dem(u,v) := \frac{w_{\ell+1}(u) \cdot w_{\ell+1}(v)}{w_{\ell+1}(U)}.$$

Bienkowski et al. show that for some parameter $h = O(\log n)$ a hierarchy of clusters with h levels can be constructed in polynomial time such that if $|U| \leq |S|/2$,

$$out(U) \leq \text{cap}(U, S-U)/h^3, \quad (4)$$

and the CMCF can be solved with value $1/h^3$ within cluster S (i.e. we can support a $1/h^3$ fraction of each demand and all the flow routes *remain within cluster* S). Following Räcke we shall refer to Condition 4 as the *bandwidth property*.

Essentially what these results say is that each cluster has good expansion properties (although the notion of expander here is slightly different from the regular notion since the cut sizes are related to the number of edges leaving a subset rather than the number of nodes in the subset.) The main idea of our paper is to use these expansion properties to obtain an integral routing with small congestion.

For some parts of the proof it will be convenient to place a node in the middle of any edge that comes into a cluster. We refer to such a node as a *boundary node*. (See Figure 2.)

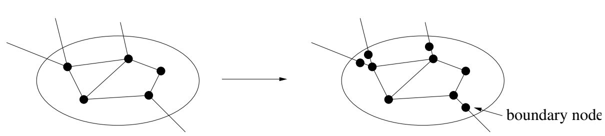

Figure 2. Adding boundary nodes at the edge of a cluster.

B. Well-linked instances

Our main result will be derived from fractional solutions to the program (1). In order to describe the result in detail we need the notion of a fractional solution being *well-linked*. For any set $A \subseteq V$ let $f(A)$ be the fractional flow that is routed to terminals in A. Following [29] we say that an instance is μ-well-linked if for any set $A \subseteq V$ such that $f(A) \leq f(V)/2$, cap$(A, V-A) \geq f(A)/\mu$. In [29] it is shown that up to constant factors in the approximation ratio we can assume that the problem instance is $O(\log^2 n)$-well-linked since we can decompose any instance into a set of disjoint instances such that each instance is $O(\log^2 n)$-well-linked and the total amount of flow that is "lost" is at most half the original flow. In other words we can restrict our attention to $O(\log^2 n)$-well-linked instances at the expense of a constant loss in the approximation ratio.

The paper [29] also showed that we can assume without loss of generality that (up to constant factors in the congestion) the fractional solution routes $\Omega(1/\log n)$ flow on any path with non-zero flow. (This in turn implies that in the fractional solution there are $O(\log n)$ flow paths through any edge.) Hence if A contains T_A terminals then $f(A) \geq T_A/\log n$ and so in an $O(\log^2 n)$ well-linked instance for any set $A \subseteq V$ such that $f(A) \leq f(V)/2$, cap$(A, V-A) = T_A/O(\log^3 n)$. Moreover, if we restrict k to be the number of terminal-pairs that have non-zero flow routed between them in the fractional solution then $k/\log n \leq OPT \leq k$. Hence in our quest for a $polylog(n)$-approximation in the number of demands routed, it is sufficient to compare the value of any candidate solution against k.

Another assumption we shall make without loss of generality is that each node is a terminal for at most 1 demand. If this condition does not hold we can easily enforce it by replacing each node that has multiple demands by a star topology.

C. Lovász Local Lemma

We shall refer to the Lovász Local Lemma on multiple occasions. Suppose we have a set of base random variables $r_1, r_2, \ldots$ and a set of events $E_1, E_2, \ldots$ that are defined by some subset of the random variables. Two events are independent if they depend on disjoint subsets of the base random variables. From the events we create an independence graph in which two events are independent if and only if they are not connected by an edge.

The original non-constructive form of the Lovász Local Lemma states that if there exist parameters $x_1, x_2 \ldots > 0$ such that $Pr[E_i] \leq x_i \Pi_{E_j \text{adj. to } E_i}(1-x_j)$ then $Pr[\text{no event } E_i \text{ occurs}] \geq \Pi_i(1-x_i) > 0$. In very recent work Moser and Tardos [30] and Haeupler et al. [31] have shown how to make this result constructive in the sense that if $Pr[E_i] \leq x_i \Pi_{E_j \text{adj. to } E_i}(1-x_j)$ for all i they obtain a randomized algorithm that with high probability sets the base random variables such that no bad event occurs. (We need the result of [31] because in our construction the number of bad events might be exponential in the problem instance. The running time of the algorithm of [31] is parameterized on the number of x_j values only.)

The Lovász Local Lemma is useful for EDPwC since it provides a way to get good congestion when rounding paths that are short. In particular suppose we have a fractional routing where all paths have length d. Then we can randomly round the solution at the expense of an $O(\log(d \log n))$ factor in the congestion. This is because the probability that any fixed edge has congestion $\Omega(\log(d \log n))$ is $O(1/poly(d \log n))$ (by a Chernoff bound). Moreover, each edge is only dependent on $poly(d \log n)$ other edges due to the short nature of the paths and the fact that the fractional solution routes $O(\log n)$ flow paths through any edge. Hence we can apply the Lovász Local Lemma to show there exists an integral routing where no edge has more than $\log(d \log n)$ congestion. In the case that $d = polylog(n)$ the congestion factor is $O(\log \log n)$.

D. Edge-Disjoint Paths in Expanders

One fact that we shall repeatedly make use of is that we can obtain good approximations for the EDPwC problem in expanders. There are many such results of this form. One early result was that of Broder, Frieze and Upfal [19] who showed how to obtain a polylogarithmic approximation in graphs with both constant degree and constant expansion. We shall use a later result due to Kolman and Scheideler [22]. They showed that for any fractional multicommodity flow in a graph, if we scale down the flow by a factor $1-\varepsilon$ for some constant ε, we can find a feasible multicommodity flow for the same demands that only uses paths of length $O((\Delta/\alpha) \log n)$ where Δ is the degree of the graph and α is its expansion parameter. This means that if Δ and α are both polylogarithmic we can use the Lovász Local Lemma to round the solution such that the congestion on any edge is $poly(\log \log n)$.

E. Expanders via Matchings

A large part of our analysis will be based on the Rao-Zhou result [23] for EDP in graphs with a large minimum cut. This result relies on another result of Khandekar, Rao, Vazirani [32] which shows how to build expanders in a robust way via a sequence of matchings. In particular, suppose we have a game where player 1 divides a set of nodes into two equal halves and player 2 finds an arbitrary perfect matching between them. Then player 1 divides the nodes again (based on player 2's response in round 1) and player 2 finds another arbitrary matching. Khandekar et al. show that there is a strategy for player 1 such that regardless of how player 2 creates the matchings, the resulting graph will have expansion at least $1/4$ after $O(\log^2 n)$ rounds. (Since player 2 creates matchings note that the degree of the resulting graph is equal to the number of rounds.)

F. Edge-Disjoint Paths in graphs with large minimum cuts

We now give a brief overview of the Rao-Zhou result which states that we can obtain a logarithmic approximation to EDP as long as each cut in the graph has size $\Omega(\log^5 n)$. Their analysis proceeds as follows. First they convert any instance with minimum cut $\Omega(\log^5 n)$ into a well-linked instance with minimum cut $\Omega(\log^3 n)$. The terminals are then grouped in order to increase the "linkedness" of the instance. They then apply a result of Karger [33] to show that if all the edges are randomly partitioned into $O(\log^2 n)$ groups, then with high probability, in *every* partition *every* cut decreases in size by a factor not much more than $O(\log^2 n)$. (This is the part of the proof where they require the instance to have a large minimum cut.) This in turn implies that if we divide the terminal groups into two distinct halves, we can create a matching between the two halves using disjoint paths. Since the partition of the edges has $O(\log^2 n)$ components, we can create these matchings $O(\log^2 n)$ times. By Khandekar et al.'s result this means that we have effectively created an expander on the terminal groups. Rao and Zhou then use a standard expander routing algorithm to connect up at least $OPT/O(\log^5 n)$ original terminal pairs using disjoint paths only.

III. Preliminaries

In this section we describe two notions that we shall use repeatedly in our analysis.

A. Expansion of Räcke clusters

We first show that each Räcke cluster may be treated as an expander in the following sense. Consider a Räcke cluster S and suppose that all its subclusters are contracted to a single node. Then, for any subset $U \subseteq S$, the value of $w_S(U)$ is simply the number of edges with an endpoint in U. If U is a connected subset such that $|U| \leq |S|/2$ then this implies that $w_S(U) \geq |U|$. (The worst case is when U is a tree with one edge joining it to the rest of S.) However, the CMCF property of S established by the construction of Bienkowski et al. implies that $cap(U, S \backslash U) \geq \frac{1}{2h^3} \min\{w_S(U), w_S(S \backslash U)\}$ since the total demand between U and $S \backslash U$ in the CMCF problem is at least $\min\{w_S(U), w_S(S \backslash U)\}/2$. This in turn implies that $cap(U, S \backslash U) \geq \frac{1}{2h^3} \min\{|U|, |S \backslash U|\}$ which implies that the cluster has expansion factor $1/2h^3$. (Note that in this section the $|\cdot|$ notation was used to denote size with respect to the number of nodes after the subclusters were contracted.)

B. The Grouping Technique

We now describe a technique that will be useful in a number of parts of the proof. At a high level the grouping technique allows us to convert a multicommodity flow solution where we have flow of size $f < 1$ leaving each terminal to a new flow where we have a flow of size 1 leaving a smaller number of terminals. Similar ideas were used by both Chekuri, Khanna, Shepherd [25] and Rao-Zhou.

The idea of the grouping technique is very simple. Suppose that we duplicate all the edges of the graph to make it Eulerian and then find an Eulerian tour. We then go around the tour and create a new group of terminals whenever we have found between $1/f$ and $2/f$ of them. If we end up with only one group of terminals then we pick an arbitrary flow path and route one unit of demand along that flow path. Otherwise, each group will contain between $1/f$ and $2/f$ terminals that are connected by a tree. Moreover, the trees for each group are edge-disjoint. Suppose that we now pick an arbitrary terminal in each group. Since the terminals in a group are connected by a tree we can push a flow of value f from the selected terminal to all the other terminals in a group such that the congestion on any edge in the tree is at most 4. (We get a factor 2 from the Eulerian tour and a factor 2 from the fact that a group could have $2/f$ terminals.) Suppose that we now wish to fractionally connect up two of the chosen terminals (call them s and t). Then we simply have to show that we can connect up $1/f$ flows of size f between pairs of terminals in the two associated groups. By concatenating this flow with the flow from the chosen terminal to the other terminals in the group, we can obtain a flow of size one between the two chosen terminals.

In addition, suppose that we can integrally route any pair of terminals s' and t', where s' is in the same group as s and t' is in the same group as t. Then, since each terminal group is connected by a tree we can use the route between s' and t' to integrally connect up the chosen terminals s and t.

IV. Overview of proof

In this section we give a basic outline of the proof of Theorem 1. We say that a cluster S is *big* (resp. *small*) if and only if $out(S)$ is at least (resp. less than) $\log^p n$, for some large constant p to be chosen later. We say that a big cluster is *critical* if all its ancestor clusters are big and all its child clusters are small. (See Figure 3.)

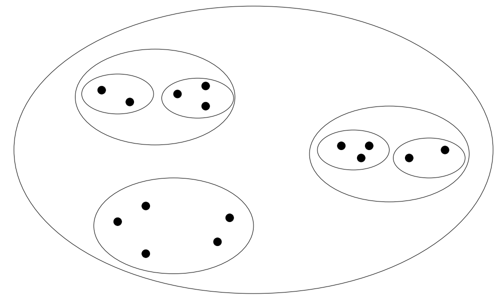

Figure 3. Classifying the clusters. The dark circles represent small clusters. All other clusters are big. The critical clusters are big clusters whose children are small.

The proof contains three main stages. First, we show how to route across small clusters. Next, we show how to route across critical clusters. Finally, we show how to route in the full graph. For this final part we divide the argument into two stages. We first look at the special case in which all nodes belong to a critical cluster. This case is much easier to handle and will contain many of the important ideas of the proof. We then move to the general case in which some nodes may not be a member of any critical cluster.

Small clusters: We route in small clusters (which have a polylogarithmic number of outedges) using an adaptation of the argument of Rao-Zhou for routing in graphs with a large minimum cut. (See Figure 4.) As already mentioned, Rao-Zhou makes use of the technique of Khandekar-Rao-Vazirani for embedding an expander in a graph via a sequence of matchings. The observation that we need for small clusters is that the number of matchings we need is $O(\log^2 n')$ where n' is the number of nodes on which we are trying to build the expander. For a small cluster we only have to connect up the at most $\log^p n$ outedges of the cluster plus whatever original terminals are contained in the cluster. (However, we shall ensure later on that we only ever have to route one original terminal in a small cluster.) This allows us to satisfy the minimum cut condition by using congestion $O((\log\log n)^2)$ on each edge.

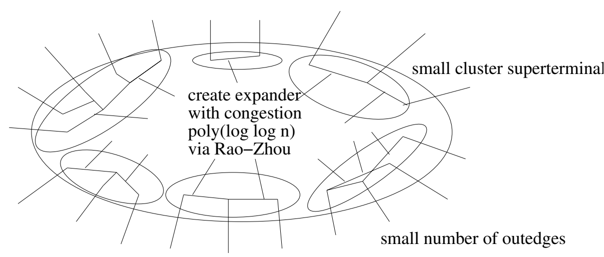

Figure 4. A small cluster showing the critical-cluster-superterminals. We route across small clusters by embedding an expander with congestion $poly(\log\log n)$ using the techniques of Rao-Zhou.

Unfortunately, the expander is not good enough to allow for arbitrary routings across the small cluster. This happens for three reasons. First, the inherent expansion of the small cluster could be as bad as $\Omega(1/\log^3 n)$. Second, the expander is built on groups of terminals, not individual terminals. Third, expander routing algorithms are not able to route arbitrary permutations. They typically can only route permutations on a subset of the nodes.

In order to deal with these difficulties we form groups of potential terminals of the small cluster using the grouping technique. (We refer to these groups as *small-cluster-superterminals*). Then when we route across the small cluster in subsequent parts of the proof we enforce the condition that only a small number of demands are routed through each group.

Critical clusters: Once we have established how to route in the small clusters we then turn our attention to the critical clusters. (See Figure 5.) Our strategy here is to use the fact that if we shrink all the subclusters of a cluster to a single node, the cluster becomes a good expander (using the argument of Section III-A.) This allows us to route across the cluster using an existing expander routing algorithm. We apply results of Kolman-Scheideler that show how to route along short fractional paths together with a Lovász Local Lemma argument that shows how to round short fractional paths so that the congestion on an edge or a small-cluster-superterminal is at most $O((\log\log n)^6)$. However, once again we cannot route arbitrary demands between the terminals of the critical cluster. We instead have to create groupings of the critical cluster terminals. We refer to these groups of terminals as *critical-cluster-superterminals*.

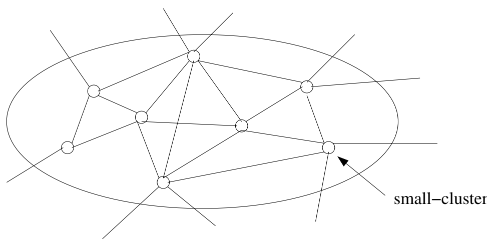

Figure 5. When routing across a critical cluster we treat it as an expander whose constituent nodes are the small clusters.

Routing in the full graph - uniform case: The final part of our analysis concerns routing in the full graph. To do this we build a new graph H by collapsing each of the critical clusters down to a single node. (All of the original terminals that are contained in the critical cluster are assigned to that node.) We start by presenting the argument for the case in which every node in the graph belongs to some critical cluster. We refer to this as the *uniform* case.

For uniform graphs we can show (using the expansion properties of the Räcke decomposition and the fact that each critical cluster is big) that the minimum cut in this graph has size $\Omega(\log^{p-3} n)$. For sufficiently large p we can then apply the analysis of Rao-Zhou to obtain a $(polylog(n), 1)$-approximation in H.

When we convert this into an algorithm for routing in the original instance G, the main issue we have to overcome is that our algorithm for routing across critical clusters assumes that we have at most one demand per critical-cluster-superterminal. However, we show that we can sample the edges in such a way that the number of demands that need to go through any critical-cluster-superterminal is $O(1)$. Hence we can simply run the algorithm for routing across the critical clusters $O(1)$ times (which leads to an eventual congestion of $O((\log\log n)^6)$.

Routing in the full graph - nonuniform case: We now describe how the above analysis can be adapted in the case

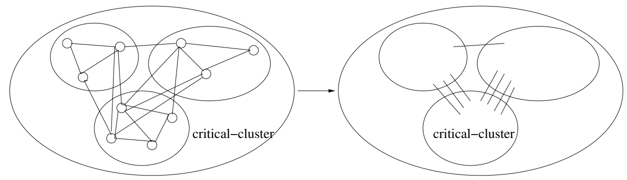

Figure 6. Suppose that all nodes belong to a critical cluster. Then, when routing in the full graph we treat it as a graph with a large minimum cut whose constituent nodes are the critical clusters.

that not all nodes are in a critical cluster. This will happen for example in the case that many clusters contain both large and small subclusters. If a cluster S contains a large subcluster S' as well as a small subcluster S'', then S' will contain a critical cluster but S'' will not be contained in any critical cluster. The problem with this situation is that if we applied Rao-Zhou directly then the sampling of the edges would mean that some small clusters could be disconnected completely. This in turn means that some cuts would not be preserved.

We deal with this situation as follows. At a high level, we try to apply surgery to the parts of the graph that are not in critical clusters so that we can essentially ignore them. We grow out paths from each cluster and then join up the endpoints of these paths. We refer to the paths emanating from these critical clusters as *tendrils*. We can then think of these joined tendrils as edges between the critical clusters. Hence we can consider a new graph whose nodes correspond to critical clusters and whose edges correspond to pairs of tendrils. We refer to this graph as the *joined-tendril graph* (see Figure 7) and we are able to show the following facts about it.

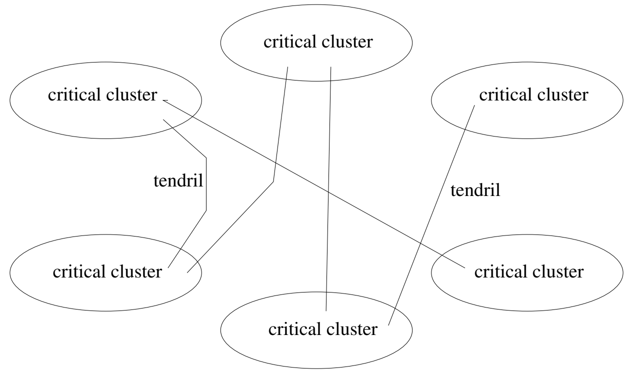

Figure 7. The joined-tendril graph.

- All terminals in the original graph can be routed on the tendrils to a critical cluster.
- Every cut in the joined-tendril graph can be converted into a cut in the original graph in which each critical cluster is mapped to the same side of the cut in both graphs. Moreover, the number of edges in these two cuts differ by at most a factor $O(\log^4 n)$.

- The congestion due to the tendrils in the original graph is $O(\log^4 n)$. However, we will use the Rao-Zhou technique to sample the tendrils in such a way that the eventual congestion on any edge and any small-cluster-superterminal is $O(\log\log n)$.
- The length of each tendril is $O(\log n)$.
- The minimum cut size in the joined-tendril graph is $\Omega(\log^{p-6} n)$. Hence we can apply the Rao-Zhou sampling technique to the joined-tendril graph in the same way that we did in the previous section.

We now briefly describe how we create the tendrils. Let H be the induced subgraph of G consisting of nodes that do not belong to a critical cluster. Suppose that H satisfies the bandwidth property (Räcke condition 4). In this case we can assume that H has outdegree less than $\log^p n$ since otherwise we can find a new critical cluster and then start the algorithm again. Since H has outdegree less than $\log^p n$ we can use small cluster routing to route across it.

For the case that H does *not* satisfy the bandwidth property, we can use a lemma of Räcke to show that H can be partitioned into a set of subgraphs that do satisfy the bandwidth property. Moreover, the total outdegree of these edges is only slightly more than $out(H)$. Therefore we can find many such clusters that have more outedges in common with $out(H)$ than outedges that are not in common with $out(H)$. Hence we can use these small clusters to extend edges from the outside of H into the interior of H. Moreover, every edge going into the interior of H can be associated with one edge from $out(H)$. This allows us to create a tendril from the exterior of H to its interior. We repeat this process and in doing so we continually shrink the interior of H. After $O(\log n)$ steps we reach a point where we have a cluster in the middle of H that satisfies the bandwidth property. At this point the tendrils are terminated.

We have now reached a situation where we can view the graph as consisting of a set of critical clusters joined by pairs of terminals. This is sufficiently similar to the uniform case in which all nodes are in critical clusters that we are able to apply similar techniques and thereby prove Theorem 1.

V. Detailed Analysis

Our algorithm is divided into three parts. We first show how to route across small clusters. We then show how to route in critical clusters. We conclude by showing how to route in the full graph.

A. Routing across the small clusters

In this section we will present an algorithm for routing in small clusters. We first show that small clusters do not contain too many original demands.

Lemma 2: Without loss of generality the number of original terminals in any small cluster is at most $\log^{p+3} n$.

Proof: For any small cluster that contains at most half the total demand, this follows immediately from the well-linked assumption. If we ever find a small cluster that contains more than half the total demand, we simply treat this small cluster as the complete instance and restart the algorithm. This affects the optimum fractional solution by a factor of 2 together with an additive term of at most $\log^p n$ since at most $\log^p n$ fractional demand is routed outside the cluster. Hence our eventual approximation ratio will not be affected by more than a constant factor as long as it is larger than $\log^p n$ (which will indeed be the case). ■

We can hence use a standard application of König's theorem[2] to group demands into $O(\log^{p+3} n)$ groups such that each group has at most 1 original terminal in each small cluster. From now on we consider one such set of demands only (at the cost of an $O(\log^{p+3} n)$ factor in the approximation ratio).

Consider a fixed small cluster S. We first use the Grouping Technique to group together the boundary nodes into groups of size $2h^3$ to overcome the fact that the expansion guarantee given by Räcke is less than 1. We shall refer to a set of boundary nodes grouped together in this way as a small-cluster-superterminal. Note that there are $O(\log^{p-3} n)$ such superterminals in each small cluster.

Lemma 3: For any cut in the cluster with x superterminals on one side and $y \geq x$ superterminals on the other side, the size of the cut must be at least x.

Proof: Denote the cut by $(X, S - X)$ and let the number of boundary nodes in X and $S - X$ respectively be $b(X)$ and $b(S - X)$. Since X contains x superterminals and $S - X$ contains $y \geq x$ superterminals we know that $x \leq \min\{b(X), b(S - X)\}/h^3$. We also have that $b(X) \leq out(X)$ and $b(S - X) \leq out(S - X)$. Hence $x \leq \min\{out(X), out(S-X)\}/h^3$. The bandwidth property of the Räcke cluster S implies that $\text{cap}(X, S - X) \geq \min\{out(X), out(S - X)\}/h^3$. Therefore $x \leq \text{cap}(X, S - X)$. ■

In other words the cut condition is satisfied for any flow between superterminals (i.e. the superterminals are 1-linked). We can now apply the analysis of Rao-Zhou which implies that in any 1-well-linked graph with n' nodes and minimum cut $\log^3 n'$ we can create an $O(\log^2 n')$-degree graph with expansion factor 1/4. We can satisfy the minimum cut condition by blowing up the congestion on each edge by a factor $O(\log^3 n')$. (In particular we will use one copy of each edge for each of the matchings required by Khandekar-Rao-Vazirani.) For the case of a small cluster we have $n' \leq \log^p n$ and so the congestion factor is $O((\log\log n)^3)$.

We now use the fact that in any n'-node graph with constant expansion we can integrally route any permutation

[2]König's theorem states that any bipartite graph with degree d can be edge-colored with d colors.

of the nodes with congestion $\log^2 n'$. (One factor $\log n'$ comes from the fact that the flow-cut gap for concurrent multicommodity flow is at most $\log n'$. (Note that due to the expansion of the graph, all cuts are satisfied up to constant factors.) Hence we can route any permutation fractionally with congestion $\log n'$. The second $\log n'$ factor comes from a randomized rounding of the fractional solution.

From this we immediately have:

Lemma 4: Suppose that we have an EDPwC instance in a small cluster such that there are at most Y demands per small-cluster-superterminal. Then we can route *all* these demands in the small cluster with congestion $O(Y(\log\log n)^5)$.

B. Routing within the critical clusters

We now show how to solve any routing problem that arises in a critical cluster S. We shall assume that the boundary nodes of the cluster together with the original demands of the cluster are grouped by the Grouping Technique to form critical-cluster-superterminals of size $\log^9 n$.

We assume that we have Z demands that need to be routed through each critical-cluster-superterminal. We can also assume that all demands are spread out among the nodes that make up a group according to the Grouping Technique. (Recall that the Grouping Technique showed how to group together flows when we have a flow of size f emanating from each terminal.) We now define a Concurrent Multicommodity Flow (CMCF) problem based on the fractional demand that is assigned to each terminal by the Grouping Technique. Note that the total number of internal demands coming from a node v within the critical cluster is at most 1 since we assumed in the previous section that each small subcluster has at most 1 original demand. Therefore, for any subset $U \subseteq S$, the total demand emanating from U is at most $2Zw_S(U)/\log^9 n$. The properties of the Räcke decomposition imply that the sparsity of the associated Concurrent Multicommodity Flow problem is at least $\log^9 n/2h^3 Z$. Hence by the worst case flow/cut gap for the CMCF problem we can fractionally route a $\log^8 n/2h^3$ fraction of the demands such that the congestion on each edge is at most Z. This in turn implies that we can route a $\log^5 n/4h^3$ fraction of the demands such that the total demand that needs to be routed through any small-cluster-superterminal is at most Z (since the number of edges in any small-cluster-superterminal is at most $2\log^3 n$.) Since $\log^5 n/4h^3 \geq 1$ this implies that all of the original demand can be routed.

We now recall that the expansion of any cluster in the Räcke decomposition is at least $1/2h^3$. In addition, the degree of any node is at most $\log^p n$. Hence, by the "short paths" result of Kolman and Scheideler [22] that we described earlier we can scale down the flow by a factor of 2 and obtain a new solution in which all flow paths have length at most $O(\log^3 n \log^p n \log n)$. Hence in order to connect up the terminal groups, we simply have to round the solution to a fractional multicommodity flow problem. However, we are now in the situation where all path lengths are $polylog(n)$ since the expansion factor of S, (namely $1/2h^3$) and $\log^p n$ are all polylogarithmic. Hence by the Lovász Local Lemma we can solve this part of the problem with congestion $O(Z\log\log n)$ on any edge and any small-cluster superterminal. Hence by Lemma 4 our final congestion is $Z(\log\log n)^6$. (Note that this rounding procedure will only connect up arbitrary members of the terminal groups. However, recall from the definition of the Grouping Technique that this is sufficient for connecting up the original demands since each original demand is connected to all the members of its terminal group by a tree and these trees are disjoint.)

C. Routing on the full graph - uniform case

We now show how we can solve the EDPwC problem on the full instance. We focus on the uniform case in which all nodes in the graph are members of a critical cluster. We essentially construct a routing on the critical clusters which only routes a small number of demands through each critical-cluster-superterminal.

We first perform a grouping of the terminals into groups of size $\log^{16} n$ such that after the Rao-Zhou style sampling has been performed we can still create flow of size 1 coming out of each terminal group in each version of the sampled graph. We then create a new graph where all critical clusters are shrunk to a single node.

Lemma 5: The size of the minimum cut in this graph before the sampling occurs is at least $\log^p n/h^3 = \Omega(\log^{p-3} n)$.

Proof: Let $(X, G-X)$ be the minimum cut and let S be some big cluster that is *not* completely contained on one side of the cut but all its child clusters are contained on one side of the cut. Then $S \cap X$ must be a union of big clusters and $S - X$ must also be a union of big clusters. Each big cluster has at least $\log^p n$ outedges. By the expansion properties of any Räcke cluster we must have cap$(S\cap X, S-X) \geq \log^p n/h^3$. ■

In the following we let $F = \Omega(\log^{p-3} n)$ be the size of the minimum cut.

We now look at each critical-cluster-superterminal and choose exactly one out-edge for this cluster. This edge is chosen at random. We then give this edge a color chosen at random from $1, \ldots, \log^2 n$. We also sample at most one original terminal from each critical-cluster-superterminal. We can now follow Rao-Zhou and apply a result of Karger [33] which states that the number of cuts of size δF is at most $n^{2\delta}$.

Recall that we have grouped the terminals into groups of size $\log^{16} n$. Suppose that we have a cut that has x of these groups on one side and at least x of these groups on the other side. By the linkedness of the example we know that the size of this cut is at least $\log^{16} nx/\log^3 n$.

Lemma 6: For each such cut, after we have sampled and colored the edges the size of the cut is at least x for each color.

Proof: Suppose not. The number of cuts of size between $2^i F$ and $2^{i+1} F$ before the sampling is at most $n^{2^{i+2}}$. Each edge appears in the sampling with probability equal to $1/\log^{11} n$ (since the size of the critical-cluster-superterminals is $\log^9 n$). The probability that a specific one of these cuts has size less than $2^i F/\log^{12} n$ after the sampling is at most $\exp(-2^{i-1} F/\log^{11} n)$ by a Chernoff bound. For sufficiently large p we have that $\exp(2^{i-1} F/\log^{11} n) \geq n^{2^{i+3}}$. (We shall choose an exact value of p in the next section when we consider the general case.) Hence by a union bound we have that for *all cuts* and *all colors* the size of the cut after the sampling and the coloring decreases by at most a factor $\log^{12} n$. Hence the cut under consideration (which originally had size $\Omega(x \log^{13} n)$) now has size at least $\Omega(x \log n)$. ■

We now apply the Rao-Zhou argument to show that we can create a $\log^2 n$-degree expander on the terminal groups of size $\log^{16} n$. The "edges" of this expander correspond to edge-disjoint paths through the network. If there are K such terminal groups then for an arbitrary set of demands between groups, Rao-Zhou show how to route $K/O(\log^5 n)$ pairs of demands on paths that are edge-disjoint *and node-disjoint* within the expander.

It remains to show how to map these routes back into our original instance. Due to the node-disjointness and edge-disjointness of the routes in the expander, each outedge of a critical-cluster-superterminal can be used once. Moreover, each critical-cluster-superterminal contains at most one original terminal that is routed. Therefore, if the routing uses a critical cluster C then the "demands" that need to be routed across the critical cluster have the property that there are at most 2 demands per critical-cluster-superterminal, i.e. the value of Z in the critical-cluster routing of Section V-B is 2. From the analysis of Section V-B these demands can be routed in the critical cluster with congestion $O((\log\log n)^6)$.

We now analyze what fraction of the demands remain. When we do the sampling of original terminals in the critical-cluster-superterminals then each terminal pair survives with probability $\frac{1}{\log^9 n} \times \frac{1}{\log^9 n}$. Moreover, a standard application of Chernoff bounds and union bounds implies that no final terminal group has more than $\log^8 n$ terminals. Hence of the remaining demands we can use König's theorem to select a $1/\log^8 n$ fraction of the remaining pairs such that we have at most one terminal in each final terminal group. Using the argument of Rao-Zhou for routing in the eventual expander that they construct, we can route an $\Omega(1/\log^5 n)$ fraction of these pairs. Moreover, recall that when we ensured that there is at most one original terminal in each small-cluster-superterminal, we had to reduce the approximation ratio by another factor of $\log^{p+3} n$. Putting all of this together, the number of demands routed is at least $k/\log^{p+34} n$ with high probability.

D. Routing on the full graph - nonuniform case

For the nonuniform case in which not all nodes are in a critical cluster, the analysis is more complex and so due to space limitations we defer the details to the full version of the paper. As already described in Section IV the main idea is to make the graph resemble a uniform instance by connecting up the critical clusters with paths that we call tendrils. We are able to show that for $p \geq 19$ and congestion $O((\log\log n)^6)$ we can route $k/\log^{p+42} n$ demands with high probability. This implies Theorem 1:

Theorem 1: There is a randomized $(O(\log^{61} n), O((\log\log n)^6))$-approximation algorithm for the Edge-Disjoint-Paths-with-Congestion problem.

Acknowledgments

This work has benefited from many discussions with Chandra Chekuri, Julia Chuzhoy, Anupam Gupta, Venkatesan Guruswami, Sanjeev Khanna, Ankur Moitra, Harald Räcke, Kunal Talwar and Lisa Zhang. The author would particularly like to thank Julia Chuzhoy for insightful comments on an earlier draft.

This research was supported by NSF contract CCF-0728980.

References

[1] P. Raghavan and C. Thompson, "Randomized rounding: a technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, pp. 365 – 374, 1991.

[2] R. Karp, "Reducibility among combinatorial problems," *In R.E. Miller and J.W. Thatcher, editors, Complexity of Computer Computations*, pp. 85 – 103, 1972.

[3] 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*. Berlin: Springer-Verlag, 1990.

[4] S. Fortune, J. Hopcroft, and J. Wyllie, "The directed subgraph homeomorphism problem," *Theoretical Computer Science*, vol. 10, no. 2, pp. 111–121, 1980.

[5] J. Kleinberg, "Approximation algorithms for disjoint paths problems," Ph.D. dissertation, MIT, Cambridge, MA, May 1996.

[6] A. Srinivasan, "Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems," in *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, Miami Beach, FL, October 1997, pp. 416 – 425.

[7] S. Kolliopoulos and C. Stein, "Approximating disjoint-path problems using greedy algorithms and packet integer programs," in *Integer Programming and Combinatorial Optimization*, 1998.

[8] C. Chekuri, S. Khanna, and B. Shepherd, "An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and UFP," *Theory of Computing*, vol. 2, pp. 137–146, 2006.

[9] C. Chekuri and S. Khanna, "Edge-disjoint paths revisited," in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[10] K. Varadarajan and G. Venkataraman, "Graph decomposition and a greedy algorithm for edge-disjoint paths," in *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004.

[11] Y. Azar and O. Regev, "Strongly polynomial algorithms for the unsplittable flow problem," in *Integer Programming and Combinatorial Optimization*, 2001, pp. 15–29.

[12] 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.

[13] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis, "Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems," in *Proceedings of the 31st Annual ACM Symposium of Theory of Computing*, Atlanta, GA, May 1999, pp. 19–28.

[14] M. Andrews and L. Zhang, "Hardness of the undirected edge-disjoint path problem," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, Baltimore, MD, May 2005.

[15] ——, "Hardness of the undirected congestion minimization problem," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, Baltimore, MD, May 2005.

[16] M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang, "Hardness of edge-disjoint paths with congestion on undirected graphs," in *Proceedings of the 46th Annual Symposium on Foundations of Computer Science*, Pittsburgh, PA, October 2005.

[17] M. Andrews and L. Zhang, "Logarithmic hardness of the directed congestion minimization problem," in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006, pp. 517–526.

[18] J. Chuzhoy, V. Guruswami, S. Khanna, and K. Talwar, "Hardness of routing with congestion in directed graphs," in *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, San Diego, CA, 2007.

[19] A. Broder, A. Frieze, and E. Upfal, "Existence and construction of edge-disjoint paths on expander graphs," *SIAM Journal Computing*, vol. 23, no. 5, pp. 976–989, 1994.

[20] ——, "Static and dynamic path selection on expander graphs: A random walk approach," *Random Structure and Algorithms*, vol. 14, no. 1, pp. 87–109, 1999.

[21] J. Kleinberg and R. Rubinfeld, "Short paths in expander graphs," in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, 1996, pp. 86–95.

[22] P. Kolman and C. Scheideler, "Improved bounds for the unsplittable flow problem," *Journal of Algorithms*, vol. 61, no. 1, pp. 20–44, 2006.

[23] S. Rao and S. Zhou, "Edge disjoint paths in moderately connected graphs," in *Proceedings of the 33rd International Colloqium on Automata, Languages, and Programming (ICALP '06)*, 2006, pp. 202–213.

[24] C. Chekuri, S. Khanna, and B. Shepherd, "Edge-disjoint paths in planar graphs with constant congestion," in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, Seattle, WA, 2006.

[25] C. Chekuri, S. Khanna, and F. Shepherd, "The all-or-nothing multi-commodity flow problem," in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 2004.

[26] H. Räcke, "Minimizing congestion in general networks," in *Proceedings of the 43nd Annual Symposium on Foundations of Computer Science*, 2002, pp. 43–52.

[27] C. Harrelson, K. Hildrum, and S. Rao, "A polynomial-time tree decomposition to minimize congestion," in *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms*, 2003, pp. 34–43.

[28] M. Bienkowski, M. Korzeniowski, and H. Räcke, "A practical algorithm for constructing oblivious routing schemes," in *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms*, 2003, pp. 24–33.

[29] C. Chekuri, S. Khanna, and B. Shepherd, "Multicommodity flow, well-linked terminals, and routing problems," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, 2005.

[30] R. Moser and G. Tardos, "A constructive proof of the general Lovász Local Lemma," *Journal of the ACM*, vol. 57, no. 2, 2010.

[31] B. Haeupler, B. Saha, and A. Srinivasan, "New constructive aspects of the Lovász Local Lemma," in *Proceedings of the 46th Annual Symposium on Foundations of Computer Science*, Las Vegas, NV, October 2005.

[32] R. Khandekar, S. Rao, and U. Vazirani, "Graph partitioning using single commodity flows," in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006, pp. 385–390.

[33] D. Karger, "Random sampling in cut, flow, and network design problems," in *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, 1994.