Algorithms for 2-Route Cut Problems

Chandra Chekuri[*]   Sanjeev Khanna[†]

May 8, 2008

Abstract

In this paper we study approximation algorithms for *multi-route cut* problems in undirected graphs. In these problems the goal is to find a minimum cost set of edges to be removed from a given graph such that the edge-connectivity (or node-connectivity) between certain pairs of nodes is reduced below a given threshold K. In the usual cut problems the edge connectivity is required to be reduced below 1 (i.e. disconnected). We consider the case of $K = 2$ and obtain poly-logarithmic approximation algorithms for fundamental cut problems including single-source, multiway-cut, multicut, and sparsest cut. These cut problems are dual to multi-route flows that are of interest in fault-tolerant networks flows. Our results show that the flow-cut gap between 2-route cuts and 2-route flows is poly-logarithmic in undirected graphs with arbitrary capacities. 2-route cuts are also closely related to well-studied feedback problems and we obtain results on some new variants. Multi-route cuts pose interesting algorithmic challenges. The new techniques developed here are of independent technical interest, and may have applications to other cut and partitioning problems.

1 Introduction

We study *multi-route cut* problems in undirected graphs which generalize well-known and standard (1-route) cut problems. Consider a single pair of nodes s, t in an edge-weighted $G = (V, E)$ with c_e denoting the weight/cost of edge $e \in E$. Then a K-route cut for s, t is a subset $E' \subseteq E$ such that removing E' would leave at most $K - 1$ edge-disjoint paths between s and t. In other words s and t are not K-edge-connected in $G[E \setminus E']$; we say that s, t are K-*separated* by E' or that E' is a K-route cut. A regular s-t cut is a 1-route cut. We are interested in finding a minimum weight K-route cuts for $K > 1$. In this paper we focus on $K = 2$ as the first non-trivial case in this class of problems. We consider several natural and well-studied cut problems such as the single-source multiple-sink cut, multiway cut, and multicut in this new setting. We also consider two orthogonal generalizations in the definition of the cut: the first is to find a set of edges that reduces the *node*-connectivity, and the second is to find node-weighted cuts instead of edge-cuts. We start by motivating the study of the multi-route cut problems.

Our primary inspiration comes from the fact that multi-route cuts are the natural dual problems to multi-route *flows*. To describe multi-route flows it is useful to consider the standard (1-route) flow between s and t in a graph G as an assignment of non-negative numbers to the set of all the *paths* $\mathcal{P}$ from s to t. Let $\mathcal{P}_K$ denote the set of all tuples $(p_1, p_2, \ldots, p_K)$ where $p_i \in \mathcal{P}$ and the

paths $p_1, p_2, \ldots, p_K$ are *edge-disjoint*. A tuple $(p_1, p_2, \ldots, p_K)$ of this form is called an *elementary K-flow*. A K-route flow between s, t is simply an assignment of non-negative numbers to elementary K-flows. Multi-route flows arise in several applications where fault-tolerance to edge (or node) failures is relevant [20, 1, 5, 6, 2], and are implicitly used in LP-relaxations for connectivity problems such as the survivable network design problem (SNDP). Kishimoto [20], building on some earlier work, introduced multi-route flows. For any K, the maximum K-route flow between s and t can be computed via the ellipsoid method for linear programming. Kishimoto [20] gave an algorithm that requires solving at most K regular maximum flows. Aggarwal and Orlin [1] simplified the ideas and analysis in [20] (see also [3]), and showed several interesting applications of multi-route flows to combinatorial problems. Just as a regular s-t minimum-cut upper bounds the s-t maximum flow, a minimum K-route cut bounds the maximum K-route flow. However, for $K > 1$, the flow-cut equivalence no longer holds even for a single pair s, t. Thus it is of interest to study the two following questions. What is the gap between the maximum K-route flow and the minimum K-route cut for a single pair as well as more general multi-pair settings? What is the complexity of finding minimum K-route cuts? Some of these questions have been raised in the recent work Bruhn *et al.* [7] who studied multi-route flows and cuts in the single source setting for unit-capacity graphs and suggested several open problems, including some of which we study in this paper.

Another motivation for multi-route cuts, and in particular for 2-route-cuts, comes from feedback problems in undirected graphs. The input to a feedback problem is a graph with either edge or node weights, and a set of (simple) cycles $\mathcal{C}$ that is usually specified implicitly. The goal is to remove edges (or nodes) of minimum-cost such that the remaining graph has no cycle from $\mathcal{C}$. These problems have received considerable attention in the past and have several applications. In the subset feedback problem, a set of terminals $S \subseteq V$ defines $\mathcal{C}$; a cycle $C \in \mathcal{C}$ iff C contains a terminal. An $O(1)$-approximation is known for this problem even in the node-weighted setting [14]. We observe that a 2-route cut corresponds to removing edges (or nodes) such that the remaining graph does not have 2-edge-disjoint (or 2-node-disjoint) paths between specified terminal pairs. In undirected graphs, a 2-route cuts for node-disjoint paths give rise to new and interesting feedback problems, e.g., the 2-route multiway-cut problem corresponds to the feedback problem where $\mathcal{C}$ is the set of cycles that contain at least *two* terminals from a given terminal set $S \subseteq V$. Note that the edge-disjoint 2-route cuts require more than simple cycles to be removed (see Fig 2).

Finally, we remark that 2-route cuts, and more generally K-route cuts, are algorithmically challenging. Many of the techniques that have been developed for the regular cut problems cannot be applied directly. We develop some new techniques in this paper and we believe that a proper and full understanding of these problems would require new algorithmic ideas of broader applicability. We discuss the technical challenges in more detail after we describe our results.

2-route Cut Problems: In this paper we restrict our attention to $K = 2$ and study several natural 2-route cut problems in undirected graphs. In particular we consider the following problems in undirected graph $G = (V, E)$.

- Single-source multiple-sink. Given a source s and multiple sinks $t_1, t_2, \ldots, t_h$, find a minimum weight cut to 2-separate s from t_i for $1 \le i \le h$.

- Multiway-cut. Given h terminals $s_1, s_2, \ldots, s_h$, find a minimum weight cut that 2-separates s_i from s_j for each $i \ne j$.

- Multicut. Given h node pairs $s_1 t_1, s_2 t_2, \ldots, s_h t_h$, find a minimum weight cut that 2-separates s_i from t_i for $1 \le i \le h$.

- Sparsest cut. Given h node pairs $s_1t_1, s_2t_2, \ldots, s_ht_h$, find a cut that minimizes the ratio of the cut cost to the number of pairs that are 2-separated.

Results: We first describe our results for the case of edge-disjoint 2-route cut problems in edge-weighted graphs. We obtain poly-logarithmic approximations for the four 2-route cut problems that we mentioned above. In particular, we achieve approximation ratios of $O(\log n)$, $O(\log n \log h)$ and $O(\log^2 n \log h)$ for the single-source, multiway-cut and multicut problems respectively. The multicut result can be used to obtain an $O(\log^2 n \log^2 h)$ approximation for the sparsest cut problem using standard ideas. Our results are obtained via a natural LP relaxation whose dual corresponds to a maximum 2-route flow problem. Thus we obtain poly-logarithmic upper bounds on 2-route flow-cut gaps.

Our techniques can be adapted to get approximation guarantees similar to the corresponding edge-disjoint version for the single-source multiple-sink and the multiway cut problem in the *node-disjoint* setting; however, the multicut problem introduces some additional technical difficulties and we leave this as a direction for future work. All our results, including those for node-disjoint paths version, extend to *node-weighted* problems with identical performance guarantees.

It is easy to show that the 2-route cut problems considered here are at least as hard to approximate as their 1-route counterparts (modulo constant factors). Moreover, it is easy to show that the LP integrality gap is $\Omega(\log n)$ for the 2-route multiway-cut problem even when the terminals span the vertex set. In the rest of the paper, we focus on the 2-route edge-disjoint cut problems in edge-weighted graphs. Most proofs are omitted due to space limitations; please see the author web pages for a longer version of the paper.

Algorithmic ideas: We illustrate the new algorithmic ideas needed to address 2-route problems by considering the simplest setting, namely the s, t 2-route cut problem. An optimal solution to this problem may be obtained as follows: *guess* an edge e, and output the edges in a minimum s, t cut in $G - e$. We leave the proof of the optimality of this procedure as a useful exercise to the reader. For general K, one needs to guess $K - 1$ edges. Thus, the single pair problem for any fixed K can be solved optimally even in directed graphs. Now consider the single-source multiple-sink problem where we wish to 2-separate s from $t_1, t_2, \ldots, t_h$. When $K = 1$ this problem can be reduced to the single pair case by simply connecting $t_1, t_2, \ldots, t_k$ to a super-sink t with infinite cost edges. However, for $K \geq 2$ this reduction does not work. In fact, the 2-route single-source multiple-sink problem is at least as hard as the regular multiway-cut problem which is known to be APX-hard. We therefore resort to approximation algorithms and consider the natural LP relaxation for these problems. The relaxation assigns lengths to the edges such that for each pair s_it_i that needs to be separated, the minimum length of 2-edge disjoint paths between them is at least 1. The main challenge is to round a fractional solution to this relaxation.

One of the difficulties with rounding for K-route-cuts when $K \geq 2$ is that the cut does not disconnect the graph into connected components with the source and sink of a terminal pair in different components. In fact, for $K \geq 2$, if the original graph G is connected, then deleting the edges in any *minimal* solution, would yield a residual graph that is still connected. Thus the standard technique of using a "ball-growing" procedure to identify the set of edges to be deleted, is not directly applicable in this setting. To concretely illustrate the difficulty of adapting classical ball-growing techniques to our setting, consider the simple example shown in Figure 1. It shows a feasible fractional solution for a 2-route cut separating s from t. The shortest distance from s to t in this example is 0, while the two edge-disjoint path distance from s to t is 1. Thus any ball-growing procedure that uses shortest path distances, will place both s and t inside the same ball, no matter how small the radius of the ball. On

the other hand, in this example, any ball grown from s w.r.t. 2 edge-disjoint path distance contains only vertex s if the radius is less than 1, and the entire graph if the radius is 1. In case of former, the only edges leaving the ball have a fractional length of 0. Thus these edges cannot be deleted in any solution with a finite performance guarantee. In case of latter, we do not get to separate s from t. This simple example also highlights that the two-edge-disjoint-path distance measure behaves quite differently from the usual shortest path distance metric. It appears that the standard region growing algorithms [21, 15] and embedding methods are difficulty to adapt to the 2-route setting.
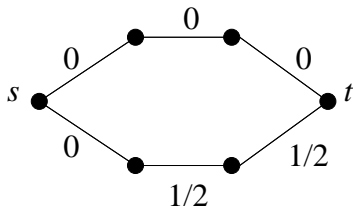

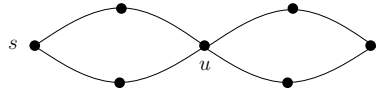Figure 1: Ball-growing w.r.t. 2 edge-disjoint paths distance

Figure 2: There is no simple cycle between s and t but an edge has to be removed to 2-separate s from t in the edge-disjoint case.

To overcome these difficulties, we introduce a novel randomized rounding technique that allows us to reduce 2-route cut problems to 1-route problems (not in a one-to-one correspondence). An interesting aspect of our reduction is that it maps feasible fractional solutions to 2-route cut problems to feasible fractional solutions to appropriate 1-route cut problems. We can then use standard rounding algorithms for the 1-route cut problems to output a feasible solution for the initial 2-route cut problem. The indirect nature of this rounding process creates technical difficulties in arguing feasibility of the final solution. We note that the random scaling we use is non-standard and there does not appear to be a natural deterministic analogue (yet) of the procedure. We believe these new techniques are of independent technical interest, and may have applications to other cut and partitioning problems.

Related Work: Cut and flow problems are ubiquitous in combinatorial optimization and hence we do not discuss this well known area. As mentioned earlier, multi-route flows have been of interest since the work of Kishimoto and others [20]. Recently, Bruhn *et al.* [7] considered the gap between a maximum K-route flow and a maximum 1-route flow for the single-source multiple-sink problem. They showed that for *unit capacity* undirected graphs, the maximum 1-route flow is not more than $2(1-1/K)$ times the maximum K-route flow. In particular this implies that for any K, in unit-capacity graphs one can obtain a simple $2(K-1)$ approximation for the single-source multi-sink K-route cut problem. We note that the unit-capacity problem is very different in nature from the problem with general capacities for $K \geq 2$. In a regular 1-route problem one can replace an edge with integer capacity c_e by c_e edges of unit capacity without changing the problem. However, for $K \geq 2$ this transformation does not preserve the flow! It is easy to construct examples where this transformation would increase the K-route flow by an unbounded amount. For instance, consider a network with two nodes s and t and two parallel edges, one with capacity 1 and other with capacity M for some integer $M \geq 1$. Then the maximum 2-route flow between s and t is 1. However, if we replace the capacity M edge by M parallel copies of unit capacity, the 2-route flow value increases to $\Omega(M)$.

2 Preliminaries

Given a (multi-)graph G, we will use n and m to denote the number of vertices and edges in G respectively. For convenience, we will assume that m is polynomially bounded in n. We let $c(e)$ denote the cost of an edge e. We say that a vertex s is K-*separated* from a vertex t if the maximum K-route flow from s to t is 0.

For any two nodes $s, t \in V$, let $\mathcal{P}_\ell^G(s, t)$ denote the set of all tuples $(p_1, p_2, \ldots, p_\ell)$ where each p_i is a path from s to t and p_i and p_j are edge-disjoint for $i \neq j$. Let $x : E \to \mathbb{R}^+$ be an assignment of non-negative weights to the edges and for a path p let $x(p) = \sum_{e \in p} x(e)$. For two nodes $s, t \in V(G)$ we let $d_1^G(s, t; x)$ denote the length of a shortest path between s and t with respect to edge weights x in the graph G. In other words $d_1^G(s, t; x) = \min_{p \in \mathcal{P}_1^G(s,t)} x(p)$. We let $d_2^G(s, t; x) = \min_{(p,q) \in \mathcal{P}_2^G(s,t)} x(p) + x(q)$ denote the minimum length of two edge disjoint paths from s to t with respect to edge weights x. We define $d_\ell^G(s, t; x)$ for any integer ℓ in a similar fashion. Finally, for any vertex s we define $B_\ell^G(s, \theta; x) = \{v | d_\ell^G(s, v; x) \leq \theta\}$. We drop the superscript G when the graph is clear from the context.

LP Relaxation: We now describe an LP relaxation for our problems - this is a natural generalization of the $K = 1$ case and has been considered earlier. Since the multicut problem captures the other two problems in our study as special cases, it suffices to only give a formulation for multicut. Recall that we restrict attention to the edge-weighted case of the 2-route edge-disjoint cut problems. For each $e \in E$ there is a variable $x(e)$ that in the binary case models whether e is in the cut ($x(e) = 1$) or not ($x(e) = 0$). In the LP relaxation, we let $x(e)$ be any non-negative number. If x is a feasible solution to the LP then $d_2(s_i, t_i; x) \geq 1$ for $1 \leq i \leq h$.

$$\min \sum_{e \in E} c(e)x(e)$$
$$x(p) + x(q) \geq 1 \qquad (p, q) \in \mathcal{P}_2^G(s_i, t_i), 1 \leq i \leq h$$
$$x_e \geq 0 \qquad e \in E.$$

One can solve the LP in polynomial time using the ellipsoid method: the separation oracle is a simple min-cost flow problem (can be solved by the successive shortest path algorithm). We remark that the LP can be generalized in straight forward fashion for K-route cuts where K is any integer. The dual of the above LP can be seen to be a maximum 2-route multicommodity flow LP; there is a scaling factor of K involved depending on whether one counts the total flow or the total elementary K-flow and we ignore this issue for now.

For the single-source multiple-sink problem the pairs to be separated are st_i, $1 \leq i \leq h$ and for the multiway-cut problem the pairs are $s_i s_j$, $i \neq j$. In the sequel we work with a feasible fractional solution to the LP and all our approximation bounds will be with respect to the lower bound provided by an optimum solution to the LP.

2.1 A Useful Lemma

The simple lemma below will be useful in our analysis.

Lemma 1 *Let $G = (V, E)$ be a graph and let $x : E \to [0, 1/3)$ be an edge weight function. Let $s \in V$ such that for all $u \in V \setminus \{s\}$, $d_2^G(s, u; x) \geq 1$. Then for any cycle C containing s there is a node v in C such that $d_1^G(s, v; x) > 1/3$.*

Proof: Let $a_1, a_2, \ldots, a_p$ be the neighbours of s. From the assumptions of the lemma, there is no a_i such that s has two parallel edges to a_i. Without loss of generality we assume that $x(s, a_i) = 0$ for $1 \le i \le p$ for otherwise we can split the edge (s, a_i) into two edges by inserting a dummy node and setting the edge closer to s have x value 0. Let $G' = G[V \setminus \{s\}]$ and A_i be the set of all nodes u in G' such that $d_1^{G'}(a_i, u; x) \le 1/3$. We observe that $A_i \cap A_j = \emptyset$ for $i \ne j$ for otherwise we would have a cycle in G containing s, a_i, a_j of total length $2/3$. Further, there is no edge between A_i and A_j since $x(e) < 1/3$ for all e.

Let C be any cycle containing s and say it contains a_i and a_j. Let P be the path between a_i and a_j obtained by removing the edges (s, a_i) and (s, a_j) from C. We need to show that there is some node v in P such that $d_1^G(s, v; x) \ge 1/3$. Let $V_i = V(P) \cap A_i$ and $V_j = V(P) \cap A_j$. As we argued before V_i and V_j are disjoint and further there are no edges between V_i and V_j. Since P connects a_i and a_j there must be a node $v \in V(P) - V_i - V_j$ such that v has an edge to V_i. This node v cannot belong to A_h for any $1 \le h \le p$ again because that would imply an edge between A_i and A_h. Therefore $d_1^{G'}(a_i, v; x) > 1/3$ for $1 \le i \le p$. Since any path from s to v has to use one of the nodes $a_1, a_2, \ldots, a_p$, we have that $d_1^G(s, v; x) > 1/3$. $\qquad\square$

Corollary 1 *Let $G = (V, E)$ be a graph and let $x : E \to [0, 1/3)$ be an edge weight function. Let $s \in V$ and $B = B_2(s, 0; x)$ such that for all $u \in V \setminus B$, $d_2^G(s, u; x) \ge 1$. Then for any cycle C containing s and a node $u \in V \setminus B$, there is a node $v \in C$ such that $d_1^G(s, v; x) > 1/3$.*

Proof: Obtain a new graph G' by shrinking $B_2^x(s, 0)$ into s. Then apply Lemma 1 to G' and s. $\square$

Application to Feedback Problems: The integrality gap of the standard LP for feedback vertex set is shown to be $O(\log n)$ in [4] and $O(\log k)$ for the more general subset feedback vertex set in [13] (hence also for the feedback edge problems). The above lemma combined with the $O(\log k)$ integrality gap for the 1-route multicut problem [15] yields simple alternate proofs of these results. Note that these problems now have a constant factor approximation but via more involved relaxations [14].

3 Single-Source 2-Route Cuts

We are given a graph G, a source s, and a set $T = \{t_1, t_2, \ldots, t_h\}$ of terminals that need to be 2-route separated from the source s. We first point out an easy reduction from multiway cut problem to this problem, establishing its APX-hardness. The reduction is essentially the same as the one in [13] for the subset feedback edge set problem.

Lemma 2 *The single-source 2-route cut problem is APX-hard.*

Proof: Suppose we are given an instance G of multiway cut with terminals $t_1, t_2, \ldots, t_k$. Obtain a new graph G' from G by adding a new vertex s and connecting s to the terminals t_1 through t_k by edges of cost ∞. Now any finite cost 2-route cut that separates s from $T = \{t_1, t_2, \ldots, t_k\}$ in G' is a multiway cut of the same cost in G for the given terminals, and vice versa. Note that 2-separating s from T in G' is equivalent to 2-separating s from $V(G') - \{s\}$. It follows that the single-source uniform 2-route cut problem is APX-hard. $\qquad\square$

Note that the reduction shows that even the restricted case of the single-source 2-route cut problem where $T = V \setminus \{s\}$ is APX-hard.

We now give a rounding algorithm for single-source multi-sink problem. Let x be a feasible fractional solution for the 2-route cut instance on G. We will assume that in the solution x, each variable $x(e)$ is assigned a value that is an integral multiple of $1/n$. This can be ensured by replacing each $x(e)$ with $\min\{1, (\lfloor 4nx(e) \rfloor)(\frac{1}{n})\}$. Since no minimal edge-disjoint collection of paths contains more than $2n$ edges, it is easy to see that the resulting solution is feasible; the new solution's cost is at most four times that of the original. For a number $\alpha \in [0, 1]$ we let $n(\alpha)$ denote the number $\lfloor n\alpha \rfloor \frac{1}{n}$.

Now pick a radius $\theta \in (0, 1)$ uniformly at random. Let $B = B_2(s, \theta; x) = \{v \mid d_2^G(s, v; x) \leq \theta\}$, and let $A = V \setminus B$. Note that all terminals lie in A and that the induced graph $G[B]$ is 2-edge-connected. Our goal now is to alter the original LP solution x into another solution x' so that $d_2^G(s, v; x') \geq 1$ for all $v \in A$ and $d_2^G(s, v; x') = 0$ for all $v \in B$.

For an edge $e = (u, v)$, let $r(e)$ denote the least radius r such that $d_2^G(s, u; x) \leq r$ and $d_2^G(s, v; x) \leq r$. Note that $r(e)$ is the same as the length of the shortest (not-necessarily simple) cycle containing s and e with edge lengths given by x. We set $x'(e)$ as follows. If $r(e) \leq \theta$ we set $x'(e) = 0$, otherwise we scale up $x(e)$ by a factor of $1/(r(e) - n(\theta))$, that is, we set $x'(e) = x(e)/(r(e) - n(\theta))$. We observe that $e \in G[B]$ implies $x'(e) = 0$. An equivalent scaling process is to pick i uniformly at random from $0, 1, \ldots, n - 1$ and set $\theta = i/n$.

Lemma 3 *In the solution* $\mathbf{x}'$, $d_2^G(s, u; x') \geq 1$ *for all* $u \in A$. *Moreover, for all* $v \in B, u \in A$, $d_2^G(v, u; x') \geq 1$.

Proof: For clarity of exposition, let $\alpha(v) = d_2^G(s, v; x)$ and $\beta(v) = d_2^G(s, v; x')$. Assume by way of contradiction that there is some $v \in A$ such that $\beta(v) < 1$. Among all such vertices choose w such that $\alpha(w)$ is largest. Let P and Q be two edge-disjoint paths from s to w such that $x'(P) + x'(Q) < 1$. By the choice of w, for any edge $e \in P \cup Q$, $r(e) \leq \alpha(w)$. Walk from w to s along P to find the first node a such that $a \in B$ (a exists since $s \in B$) and let P' be the sub-path of P from w to a. Using Q, define b and Q' as above. We claim that $x(P') + x(Q') \geq \alpha(w) - \theta$ for otherwise we can use P' and Q' to find two disjoint paths between s and w of total x length strictly less than $\alpha(w)$. We prove this claim after we use it to finish the proof of the lemma. Note that for any edge e in $P' \cup Q'$, $x'(e) = x(e)/(r(e) - n(\theta)) \geq x(e)/(r(e) - \theta) \geq x(e)/(\alpha(w) - \theta)$. Thus, after scaling $x'(P' \cup Q') \geq 1$ which implies that $x'(P \cup Q) \geq 1$ contradicting our assumption. For the second part of the lemma, we observe that $d_2^G(s, v; x') = 0$ for any $v \in B$. Further, by the triangle inequality $d_2^G(s, u; x') \leq d_2^G(s, v; x') + d_2^G(v, u; x')$ and hence $d_2^G(v, u; x') < 1$ implies $d_2^G(s, u; x') < 1$.

Now we prove the claim. Since $a \in B$, there are two edge disjoint paths P_a and Q_a from a to s such that $x(P_a) + x(Q_a) \leq \theta$. Further, P_a and Q_a have all their edges in $G[B]$. Similarly let P_b and Q_b be the paths for b. We claim that $P' \cup Q' \cup P_a \cup Q_a \cup P_b \cup Q_b$ contain two edge disjoint paths from w to s of total length at most

$$x(P') + x(Q') + (x(P_a) + x(Q_a))/2 + (x(P_b) + x(Q_b))/2 < \alpha(w) - \theta + \theta/2 + \theta/2 < \alpha(w).$$

It is easy to see that $P' \cup Q' \cup P_a \cup Q_a \cup P_b \cup Q_b$ contains two edge disjoint paths from w to s. To bound the total x-length of these paths, we create a *fractional* flow of two units from w to s of the desired length such that no edge has more than one unit of flow. Then the claim follows by using the fact that there exists an integer flow of no higher cost than the fractional flow. Send one unit of flow from w along P' to a which then splits the flow into a half unit along P_a and another half along Q_a. The other unit of flow is sent along Q' and split at b for P_b and Q_b. It can be checked that no edge has more than one unit of flow and that the x-length of this flow is equal to $x(P') + x(Q') + (x(P_a) + x(Q_a))/2 + (x(P_b) + x(Q_b))/2$. $\square$

We next show that the expected cost of the resulting solution is only $O(\log n)$ times larger than the cost of the original solution.

Lemma 4 $E_\theta[x'(e)] = O(\log n) \cdot x(e)$.

Proof: Let $r(e) = i/n$ for some integer i. Note that $x'(e) = 0$ if $\theta \geq r(e)$ and otherwise $x'(e) = x(e)/(r(e) - n(\theta))$. Therefore

$$E_\theta[x'(e)] = \int_0^{r(e)} \frac{x(e)}{r(e) - n(\theta)} \cdot d\theta = x(e) \sum_{j<i} \frac{1}{n} \cdot \frac{1}{i/n - j/n} = O(\log n)x(e).$$

$\square$

Remark 3.1 *Lemmas 3 and 4 hold for the following modified scaling procedure as well: $x'(e) = x(e)$ if $\theta \geq r(e)$ and $x'(e) = x(e)/(r(e) - n(\theta))$ otherwise.*

The rounding procedures for 2-route multiway cut and 2-route multicut implicitly need the modifed analysis mentioned in the above remark. The analysis given in Lemma 4 is tight even when there is a single terminal to be separated from the source, and all edge weights are 1. Consider a graph G with vertices v_0 through v_n. Let $s = v_0$ and $t = v_n$. For each $0 \leq i < n$, there are two parallel edges between v_i and v_{i+1}. Consider an LP solution that for each pair of parallel edges assigns $x_e = 1/n$ on one of the edges and 0 on the other. Then $d_2(s, v_i) = i/n$. Now consider the edge e between $v_{n/2-1}$ and $v_{n/2}$. With probability $1/n$ the initial radius r is between i/n and $(i + 1)/n$. When $i < n/2$, we scale e by a factor roughly $1/(1/2 - i/n)$. Thus the expected scaling factor is $\Omega(\log n)$; a similar argument shows that this holds for $\Omega(n)$ edges.

We now complete the description of the algorithm by showing how we can use the solution x' to find a feasible cut. We remove any edges e such that $x'(e) \geq 1/3$. In the remaining graph let $T' = \{u \mid d_1^G(s, u; x') \geq 1/3\}$. We solve a single source min-cut problem to disconnect s from T'. Note that $3x'$ is sufficient to pay for both the above steps since the single source min-cut problem has an integrality gap of 1. By Corollary 1, any cycle involving s and a node from A contains a node from T'. Therefore, separating T' from s ensures that there is no cycle involving s and a node from A. Since all terminals are in A, the solution is feasible. The expected cost is $3 \sum_e c(e)x'(e)$ which by Lemma 4 is $O(\log) \sum_e c(e)x(e)$. We can easily derandomize the procedure by using standard ideas; the proof of Lemma 4 shows the existence of a $\theta \in [0, 1)$ such that $\sum_e c(e)x'(e) = O(\log n) \sum_e c(e)x(e)$. We observe that there are only n distinct values in $\{d_2(s, v; x) \mid v \in V\}$ that are relevant in choosing θ, hence we can try all these values and pick the one which results in the least cost. This gives the following theorem.

Theorem 1 *Single-source 2-route cut problem has an $O(\log n)$-approximation.*

4 2-Route Multiway Cut

Let $S = \{s_1, s_2, \ldots, s_h\} \subseteq V$ be a set of terminals. In the 2-route Multiway Cut problem the goal is to find a minimum cost set of edges whose removal 2-separates s_i and s_j for all $1 \leq i < j \leq h$. The isolating cut heuristic [11] and a greedy splitting algorithm [22] give a $2(1 - 1/h)$-approximation for the standard (1-route) multiway cut problem. The generalization of the isolating cut heuristic to the 2-route problem is the following. For each i, find $E_i \subseteq E$ to 2-separate s_i from all nodes in $S \setminus \{s_i\}$

using either an exact or an approximate algorithm. Output $\cup_{1 \le i < h} E_i$ where we assume without loss of generality that E_h is the set of largest weight amongst $E_1, \ldots, E_h$. The following example shows that this can give a solution of value $\simeq (h-1)\mathsf{OPT}$. Consider a cycle on h nodes $v_1, v_2, \ldots, v_h$ with each edge of the cycle of weight 1. Connect s_i to v_i using two parallel edges of weight $1 - \epsilon$. An optimum isolating cut for s_i consists of one of the edges that connects s_i to v_i. Thus the isolating cut heuristic can output a solution of value $(1 - \epsilon)(h - 1)$. In this example $\mathsf{OPT} = 1$; simply remove any edge of the cycle. Note that this example is similar to the one that demonstrates that the greedy algorithm gives a tight $2 - 2/k$ ratio for the multiway-cut problem [22].

Lemma 5 *The integrality gap of the LP for 2-route multiway-cut is $\Omega(\log n)$ even when $S = V$.*

Note that when $S = V$ the problem is equivalent to the feedback edge set problem and the LP solution for the 2-route problem is equivalent to the LP solution for the feedback edge problem for which the $\Omega(\log n)$ gap was observed in [13] using high-girth expanders.

We give an LP rounding approach that gives an $O(\log n \log h)$ approximation. Let x be a feasible solution to the LP. As before we will assume that $x(e)$ is an integer multiple of $1/n$. Let $e = (u, v)$. We let $r_i(e)$ to be the smallest r such that $B_2(s_i, r; x)$ contains both end points of e. We set $r(e) = \min_i r_i(e)$. The algorithm consists of the following steps.

1. Pick θ uniformly at random from $[0, 1/4)$.

2. For each e, set $x'(e) = \max\{2x(e), x(e)/(r(e) - n(\theta))\}$.

3. Remove edges e such that $x'(e) \ge 1/3$.

4. Separate all pairs (s_i, v) with $d_1^G(s_i, v; x') \ge 1/3$ by solving a multi-cut prob.

5. Output edges removed in Steps 3 and 4.

We analyze the algorithm to prove the theorem below.

Theorem 2 *The 2-route multiway cut problem has an $O(\log n \log h)$-approximation.*

The analysis is based on the following lemmas.

Lemma 6 *For $1 \le i \le h$ define x'_i by $x'_i(e) = x'(e)$ if $r_i(e) > \theta$ and $x'_i(e) = 0$ otherwise. Then for any $u \in V \setminus B_2(s_i, \theta; x)$, $d_2^G(s_i, u, x'_i) \ge 1$.*

Proof: Let $\theta \in [0, 1/4)$ and $i \in \{1, \ldots, h\}$. Define x''_i by $x''_i(e) = 0$ if $r_i(e) \le \theta$ and $x''_i(e) = x(e)/(r_i(e) - \theta)$ otherwise. Then by essentially the same argument as in Lemma 3, $d_2^G(s_i, u, x''_i) \ge 1$ for any $u \in V \setminus B_2(s_i, \theta; x)$. We claim that $x'_i(e) \ge x''_i(e)$ for all e. To see this we consider two cases. If $\theta < r(e)$ then $x'(e) = x(e)/(r(e) - \theta)$ and $r(e) - \theta < r_i(e) - \theta$ and hence $x'_i(e) \ge x''_i(e)$. If $\theta > r(e)$ it implies that $r(e) < 1/4$. There are two sub-cases. If $r_i(e) = r(e)$ then $x'_i(e) = x''_i(e) = 0$. Otherwise $r(e) = r_j(e)$ for $j \ne i$. We claim that that $r_i(e) \ge 3/4$ for otherwise the end points of e are in both $B_2(s_j, r(e); x)$ and $B_2(s_i, 3/4; x)$ and since $r(e) < 1/4$, we have $d_2(s_i, s_j; x) < 1$ which contradicts the feasibility of x. If $r_i(e) \ge 3/4$ then $x''_i(e) \le 2x(e) \le x'(e) = x'_i(e)$. $\square$

Lemma 7 *Let F be the set of edges removed by the multi-cut algorithm in Steps 3 and 4. Then F is a feasible solution to the given instance.*

9

Proof: After removing edges in Step 3, consider the remaining graph. Let $x_i'(e)$ be defined as in Lemma 6. Let T_i be the set of nodes u such that $d_1^G(s_i, u, x_i') \geq 1/3$. Since $x' \geq x_i'$, $d_1^G(s_i, u; x') \geq 1/3$ for each node $u \in T_i$. By Corollary 1, separating s_i from nodes in T_i removes all cycles that contain s_i and any node in $V \setminus B_2(s_i, \theta; x)$. Note that every terminal s_j, $j \neq i$ is in $V \setminus B_2(s_i, \theta)$. Since the multicut instance separates s_i from T_i, after removing F, there are no cycles containing two or more terminals. $\square$

Lemma 8 *The total weight of edges removed is in expectation $O(\log n \log h) \sum_{e \in E(G} c(e) x(e)$.*

Proof: The expected value of $x'(e)$, as in the proof of Lemma 4, is $O(\log n) x(e)$. In Step 4 we solve a multicut problem. Notice that $3x'$ gives a feasible fractional solution to the multicut instance that we define. Using [15, 18] there is an integral multicut of value $O(\log h)$ times the fractional solution value. Thus the expected weight of the edges removed is $O(\log n \log h) \sum_e c(e) x(e)$. $\square$

5   2-Route Multicut

We now consider the 2-route multicut problem. We are given G and h pairs $s_1 t_1, s_2 t_2, \ldots, s_h t_h$ and the goal is to 2-separate s_i from t_i for $1 \leq i \leq h$.

We give an LP rounding algorithm that essentially reduces it to the standard multicut problem. Our algorithm is inspired by the algorithms of Calinescu, Karloff and Rabani for multiway-cut [8] and 0-extension [9]; the underlying idea has seen several applications subsequently. Let x be a feasible solution to the LP. For an edge $e = uv$, we let $r_i(e) = \max\{d_2(s_i, u; x), d_2(s_i, v; x)\}$.

1. For each $u \in V(G)$, set $\rho(v) = 0$.

2. Pick θ uniformly at random from $[0, 1/2)$ and pick a random permutation σ of $\{1, 2, \ldots, h\}$.

3. For $i = 1$ to h do

   - For $v \in B_2(s_{\sigma(i)}, \theta; x)$, if $\rho(v) = 0$ then $\rho(v) = i$.

4. For each edge e:

   - Find the least index j such that $r_{\sigma(j)}(e) \leq \theta$; if no such j exists then set $j = h + 1$.
   - If $j = 1$, set $x'(e) = x(e)$, else set $x'(e) = \max_{i < j} x(e) / (r_{\sigma(i)}(e) - n(\theta))$.

5. Remove edges e such that $x'(e) \geq 1/10$.

6. In G separate all pairs (p, q) with $d_1^G(p, q; x') \geq 1/6$ via a multicut algorithm.

7. Output edges removed in Steps 5 and 6.

We observe that the first three steps of the above algorithm are similar to the adaptation of the CKR procedure and analysis from [9] for the (1-route) multicut problem (see lecture notes [17, 10] for details of this). The only difference is that step 3 is performed with respect to 1-route distance. At the end of step 3, the 1-route multicut algorithm outputs as solution all edges (u, v) such that $\rho(u) \neq \rho(v)$. The feasibility of this solution is immediate since the radius of each ball is less than $1/2$, and hence no ball can contain both a source and its corresponding sink. An elegant argument from [9] can then

be used to show that the expected cost of this solution is within an $O(\log n)$ factor of the optimal. We note that the classical region growing algorithm of [15] may be viewed as a deterministic version of this randomized ball-growing process.

In contrast, for 2-route multicut, a critical step is the randomized scaling (step 4) which allows us in effect to reduce our problem to an instance of 1-route multicut. The cost analysis of the resulting solution combines the scaling analysis from Lemma 4 with the argument from [9] followed by the integrality gap for the standard 1-route multicut [15]; this is not too difficult. The main difficulty, however, is in proving the feasibility of the resulting solution. In the setting of 2-route distance, the sets $\{v \mid \rho(v) = i\}$ are difficult to visualize, and the intuitive distance based arguments are no longer applicable. We rely on a careful inductive proof to argue for the feasibility of the cut produced by the algorithm.

5.1 Feasibility

We will show that the solution obtained in the step 7 above is indeed a feasible solution. For clarity of exposition, assume without loss of generality that the permutation σ is an identity permutation. For $i \in [1..h]$ let $V_i = \{w \mid \rho(w) = i\}$.

Lemma 9 *For any node $w \in V_i$ and $u \in V \setminus V_i$, we have $d_2(w, u; x') \geq 1$.*

Proof: We will prove this by induction on i. The base case of the hypothesis $(i = 1)$ is true by Lemma 3. Assume the hypothesis is true for $1 \leq \ell < i$. Now consider any vertex $w \in V_i$ and $u \in V \setminus V_i$.

Consider any pair of edge-disjoint paths P, Q from vertex w to u. Let z be a vertex on $V(P \cup Q)$ such that $\rho(z)$ is minimum over all vertices in $V(P \cup Q)$. Suppose $\rho(z) = q$. If $q \geq i$, we say that the pair of paths P, Q is *clean*. Consider first the case $q < i$. We claim that $x'(P) \cup x'(Q) \geq 1$ by applying induction hypothesis to V_q, and noting that $w \in V \setminus V_q$.

Now consider all nodes u such that there is an *infeasible* clean pair of paths P, Q connecting w to u, that is, $x'(P) + x'(Q) < 1$. Now we derive a contradiction in a manner very similar to that in the proof of Lemma 3. The reason for this is the fact that P and Q are clean and hence the scaling process is essentially similar to the single source case where the source is now s_i.   □

The main technical lemma needed to establish feasibility is the one below.

Lemma 10 *For any $i \in [1..h]$, let C be any cycle (possibly non-simple) that involves a node $w \in V_i$ and a node $u \in V \setminus V_i$. Then after scaling in the step 4 of the algorithm, either the cycle C has a pair of nodes p, q such that $d_1(p, q; x') \geq 1/6$ or there is an edge e on C such that $x'(e) \geq 1/10$.*

Proof: We focus on cycles that do not contain any edge e with $x'(e) \geq 1/10$. We will prove this also by induction on i. The base case with $i = 1$ follows from Lemma 1. Consider the inductive step. Let z be a vertex on C such that $\rho(z)$ is minimized. Suppose $\rho(z) = q$. If $q \geq i$, we say that the cycle C is *clean*. Consider first the case when $q < i$. In this case, the assertion follows by applying induction hypothesis to V_q, and noting $w \in V \setminus V_q$.

Now we consider the case when C is a clean cycle. We can assume without loss of generality that C is a simple cycle. If not, consider an Eulerian traversal of C that decomposes it into a collection of simple cycles. Since C involves a vertex in V_i and a vertex in $V \setminus V_i$, at least one of the cycles in the decomposition, say C', satisfies this property as well. Thus we can set $C = C'$.

11

For clean simple cycles, we prove a stronger statement. Let C be a clean (simple) cycle containing a node u with $\rho(u) > i$. Let $a, b \in V_i \cap V(C)$ (not necessarily distinct) be such that the two paths $P_{a,b}^1$ and $P_{a,b}^2$ induced by C between a and b satisfy the following property: $P_{a,b}^1$ contains only nodes in V_i and $P_{a,b}^2 = av_1 \ldots v_2 b$ where $v_1, v_2 \notin V_i$. Then there is a node t on C such that $d_1(a, t; x') \geq 1/6$ (by symmetry, same holds for b). Note that for any clean cycle there exist a, b satisfying the above properties. To see this, say C connects u to $w \in V_i$. Traverse the cycle starting at u and let a be the first node encountered in V_i. Continue the traversal and let b be the last node after a before the traversal encounters a node outside V_i. It can be easily checked that a, b satisfy the desired properties.
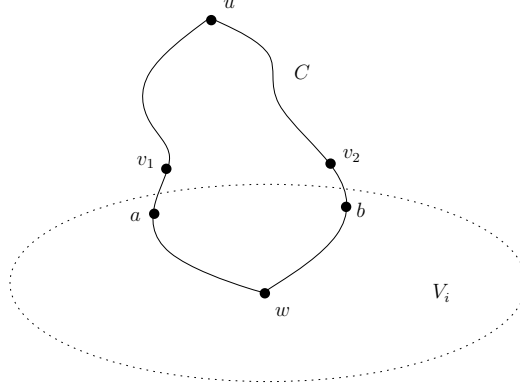

Figure 3: Illustration for proof of Lemma 10.

Now we prove the claim by way of contradiction. Consider the smallest length (w.r.t. x') clean cycle that does not satisfy the claim. Also, we focus on the case where $a \neq b$. Let α be the shortest path distance from a to b in the graph $G[V_i]$ with respect to x'. We consider two cases based on α.

Case 1: $\alpha \geq 1/2$. Consider the segment $P_{a,b}^1$ as defined above. Recall that $P_{a,b}^1$ contains only vertices from V_i. Traverse $P_{a,b}^1$ from a to b, and let t be the first node such that the distance traversed along $P_{a,b}^1$ is at least $1/4$. If $d_1(a, t; x') \geq 1/6$, the node t is the desired node. Otherwise, there is a path $P_{a,t}$ of length $< 1/6$ from a to t. If $P_{a,t}$ leaves V_i then it either touches a node z such that $\rho(z) < i$ or $\rho(z) > i$. In either case, the path will induce a cycle of length < 1 that contains some node on $P_{a,b}^1$ and the node z. This is a contradiction to Lemma 9.

If $P_{a,t}$ is completely contained in V_i then we obtain a strictly shorter cycle C' by shortcutting $P_{a,b}^1$ using the new path. If C' is a simple cycle then it satisfies the properties with respect to u, a, b; therefore there is a node t' on C' such that $d_1(a, t'; x') \geq 1/6$. Note that t' cannot be on $P_{a,t}$ and hence must be on C. If C' is not a simple cycle then again we get a contradiction to the choice of C as the smallest cycle that does not satisfy the claim.

Case 2: $\alpha < 1/2$. Consider the segment $P_{a,b}^2$ as defined above. Note that the length of $P_{a,b}^2$ must be at least $1/2$. Otherwise, we obtain a clean cycle containing u, a, b that has length < 1: concatenate $P_{a,b}^2$ with the shortest path from a to b inside V_i. By Lemma 9, this is a contradiction.

Let v_1 be the neighbour of a on $P_{a,b}^2$ and let v_2 be the neighbour of b on $P_{a,b}^2$. Note that $v_1, v_2 \notin V_i$. Let t be the first node on $P_{a,b}^2$ at distance $> 1/4$ *along* the path from a, and let $P_{a,t}^2$ denote this segment of $P_{a,b}^2$. Since $x'(e) < 1/10$ for each $e \in C$, we have $t \neq v_1, v_2$. If $d_1(a, t; x') \geq 1/6$, then t is the desired node. Otherwise, there is a path $P_{a,t}$ of length $< 1/6$ from a to t. Let v be the first node on the path $P_{a,t}^2$ that $P_{a,t}$ touches. Since $P_{a,t}$ ends at t, v exists. We consider two sub-cases based on v. Let $P_{a,v}$ denote the segment of the path $P_{a,t}$ from a to v.

If $v \neq v_1$ then the path $P_{a,v}$ and $P_{a,v}^2$ form a cycle that contains v_1 and a but has length < 1: a contradiction to Lemma 9.

If $v = v_1$, then the first edge on $P_{a,v}$ must be the edge (a, v_1). Otherwise, we get a cycle of length < 1 that contains node $a \in V_i$ and node $v_1 \in V \setminus V_i$: once again, a contradiction to Lemma 9. Moreover, by the same reason, the path $P_{a,t}$ must be vertex-disjoint from $P_{a,b}^1$. Also, we can assume without loss of generality that $P_{a,t}$ is a simple path.

Suppose the path $P_{a,t}$ contains a node z with $\rho(z) < i$. Then consider the graph induced by $E(P_{a,t}) \oplus E(P_{a,t}^2)$ (the symmetric difference). The induced graph is Eulerian and hence can be decomposed into a collection of *simple* cycles, say $C_1, C_2, ..., C_\gamma$. There is at least one cycle since z is in $P_{a,t}$ and not in $P_{a,t}^2$. The total length of these cycles is no more than $x'(P_{a,t}) + x'(P_{a,t}^2)$ which is at most $1/6 + 1/4 + 1/10 < 1$ and hence each cycle is of length < 1. Moreover, since the $\rho()$ value of every vertex on $P_{a,b}^2$ is at least i, at least one of these γ cycles contains a pair of vertices z_1, z_2 such that $\rho(z_1) \geq i$ and $\rho(z_2) < i$. By Lemma 9, this is a contradiction.

Thus for every vertex z on $P_{a,t}$, we have $\rho(z) \geq i$. Recall that $P_{a,t}$ is vertex disjoint from $P_{a,b}^1$. Consider the shortest path Q' between v_1 and b in the graph induced by $E(P_{v1,t}) \cup E(P_{v1,b}^2)$; the length of Q' is strictly less than the length of $P_{v1,b}^2$ since $P_{a,t}$ and $P_{a,t}^2$ both share the edge av_1 but $x'(P_{a,t}) < 1/6$ and $x'(P_{a,t}^2) \geq 1/4$. Now consider the cycle C' obtained by concatenating the edge av_1 with Q' and $P_{a,b}^1$. We thus have $x(C') < x(C)$ and C is a clean simple cycle that contains a and v_1. And hence there is a node t' on C such that $d_1(a, t'; x') \geq 1/6$. Moreover, $t' \notin V(P_{a,t})$ and hence $t' \in C$. $\qquad \square$

We now finish the proof of the feasibility of the solution output by the algorithm. For any pair $s_i t_i$, $1 \leq i \leq h$, we claim that $\rho(s_i) \neq \rho(t_i)$. Suppose not. Let $\rho(s_i) = \rho(t_i) = q$. Then $s_i \in B_2(s_{\sigma(q)}, \theta; x)$ and $t_i \in B_2(s_{\sigma(q)}, \theta; x)$ which implies that $d_2(s_i, t_i; x) \leq 2\theta$. Since $\theta < 1/2$ this would imply that $d_2(s_i, t_i; x) < 1$ which contradicts the feasibility of x.

From above and Lemma 10, for any cycle C that contains both s_i and t_i, either there is any edge e in C such that $x'(e) \geq 1/10$ or there are nodes p, q in C such that $d_1(p, q; x') \geq 1/6$. Since the algorithm removes all edges f with $x'(f) \geq 1/10$ (in Step 5) and ensures that there is no path between nodes p, q with $d_1(p, q; x') \geq 1/6$ (in Step 6), every cycle C between s_i and t_i is removed.

5.2 Cost Analysis

We will first analyze the cost of the solution x'. To do so, it suffices to consider the expected scaling factor for any edge in G. Fix $\theta \in (0, 1/2)$ and an edge $e = (u, v)$. Recall that $r_i(e) = \max\{d_2(s_i, u; x), d_2(s_i, v; x)\}$. By renumbering pairs, assume that $r_1(e) \leq r_2(e) \leq ... \leq r_h(e)$. We will denote by $f(e)$ the scaling factor for edge e. Define $f_i(e, \theta) = 1$ if $r_i(e) \leq \theta$ and $\frac{1}{(r_i(e) - n(\theta))}$ otherwise. The scaling factor $f(e)$ for edge e is determined to be $f_i(e, \theta)$ only if in the random permutation σ, the source s_i occurs before each one of $s_1, s_2, ..., s_{i-1}$. The probability of this event is at most $1/i$. Thus for a fixed choice of θ, the expected scaling factor for an edge e can be bounded by $E_\sigma[f(e)] \leq \sum_{i=1}^h \frac{1}{i} f_i(e, \theta)$.

Taking the expectation over θ, which is independent of σ, we get the expected scaling factor for the edge e is at most

$$
\begin{aligned}
E_{\theta,\sigma}[f(e)] &\leq \int_0^{1/2} \sum_{i=1}^h \frac{1}{i} f_i(e, \theta) \cdot d\theta = \sum_{i=1}^h \frac{1}{i} \left(\int_0^{r_i(e)} \frac{1}{r_i(e) - n(\theta)} \cdot d\theta + \int_{r_i(e)}^{1/2} 1 \cdot d\theta \right) \\
&= O(\log h \log n).
\end{aligned}
$$

Thus the expected cost of the solution x' is $O(\log h \log n)$ times the cost of the solution x. Finally, we lose another factor of $O(\log n)$ in solving the multicut instance on x'. We thus get the following theorem.

Theorem 3 *There is an $O(\log h \log^2 n)$-approximation algorithm for the 2-route multicut problem.*

6 Conclusions

We obtained the first poly-logarithmic approximation guarantees and flow-cut gap results for 2-route cut problems in graphs with arbitrary edge or node capacities. In the process, we developed some new rounding techniques. Many problems remain open. The bounds we establish on the integrality gap are not tight. It seems particularly challenging to establish strong lower bounds on the gaps. In particular, does the single-source multiple-terminal problem admit a constant factor approximation via the LP? The multicut problem in the 2-node-disjoint setting generalizes the subset feedback problem, and it would be interesting to see if one can obtain a poly-logarithmic approximation for this problem. Finally, K-route cut problems for $K \geq 3$ require substantially new ideas. It would also be interesting to explore the hardness of approximation of many of these problems, even when K is large.

References

[1] C. Aggarwal and J. Orlin. On Multi-route Maximum Flows in Networks. *Networks*, 39:43–52, 2002.

[2] A. Bagchi, A. Chaudhary and P. Kolman. Short length Menger's theorem and reliable optical networking. *Theoretical Computer Science*, 339:315–332, 2005.

[3] A. Bagchi, A. Chaudhary, P. Kolman and J. Sgall. A simple combinatorial proof for the duality of multiroute flows and cuts. TR 2004-662, Charles Univ., 2004.

[4] R. Bar-Yehuda, D. Geiger, J. Naor, R. M. Roth. Approximation Algorithms for the Feedback Vertex Set Problem with Applications to Constraint Satisfaction and Bayesian Inference. SIAM J. Comput. 27(4): 942-959 (1998).

[5] G. Brightwell, G. Oriolo and F. B. Shepherd. Some strategies for reserving resilient capacity. *SIAM J. on Discrete Math.*, 14(4):524–539, 2001.

[6] G. Brightwell, G. Oriolo and F. B. Shepherd. Reserving Resilient Capacity for a Single Commodity with Upper Bound Constraints. *Networks*, 41(2): 87–96, 2003.

[7] H. Bruhn, J. Cerny, A. Hall and P. Kolman. Single Source Multiroute Flows and Cuts on Uniform Capacity Networks. *Proc. of ACM-SIAM SODA*, 2007.

[8] G. Călinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. *Journal of Computer and System Sciences*, 60:564–574, 2000.

[9] G. Călinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. *SIAM J. on Computing*, 34(2): 358–372, 2004.

[10] C. Chekuri. Lecture notes on "Multicut rounding via CKR method". http://www.cs.uiuc.edu/homes/chekuri/teaching/fall2006/lect15.pdf

[11] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM J. on Computing*, 23:864–894, 1994.

[12] G. Even, J. Naor, S. Rao and B. Schieber, Divide-and-conquer approximation algorithms via spreading metrics. *JACM*, 47:585–616, 2000.

[13] G. Even, J. Naor, B. Schieber, L. Zosin. Approximating minimum subset feedback sets in undirected graphs with applications. *SIAM J. Disc. Math*, 13(2):255–267, 2000.

[14] G. Even, J. Naor and L. Zosin. An 8-approximation for the subset feedback vertex set problem. *SIAM J. on Computing*, 30(4):1231–1252, 2000.

[15] N. Garg, V. Vazirani, and M. Yannakakis. Approximate Max-Flow Min-(Multi)Cut Theorems and Their Applications. *SIAM J. Comput.*, 25(2): 235-251, 1996.

[16] M. Grötschel, L. Lovász and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. Springer-Verlag, 1987.

[17] A. Gupta and R. Ravi. Lecture notes on "LP solutions as Metrics: MultiCut, and Region Growing". http://www.cs.cmu.edu/afs/cs/academic/class/15854-f05/www/scribe/lec20.pdf

[18] O. Günlük. A new min-cut max-flow ratio for multicommodity flows. *SIAM J. on Discrete Math.*, 21(1):1–15, 2007. Preliminary version in *Proc. of IPCO*, 2002.

[19] D. Karger, P. Klein, C. Stein, M. Thorup, and N. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, 668–678, 1999.

[20] W. Kishimoto. A method for obtaining the maximum multi-route flow in a network. *Networks*, 27(4):279–291, 1996.

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

[22] V. Vazirani. Approximation Algorithms. Springer, 2001.