Towards Duality of Multicommodity Multiroute Cuts and Flows: Multilevel Ball-Growing*

Petr Kolman[1] and Christian Scheideler[2]

1 Faculty of Mathematics and Physics, Charles University in Prague
Malostranské nám. 25, 118 00 Prague, Czech republic
kolman@kam.mff.cuni.cz

2 Dept. of Computer Science, University of Paderborn
Fürstenallee 11, 33102 Paderborn, Germany
scheideler@upb.de

Abstract

An *elementary h-route flow*, for an integer $h \geq 1$, is a set of h edge-disjoint paths between a source and a sink, each path carrying a unit of flow, and an *h-route flow* is a non-negative linear combination of elementary h-route flows. An *h-route cut* is a set of edges whose removal decreases the maximum h-route flow between a given source-sink pair (or between every source-sink pair in the multicommodity setting) to zero. The main result of this paper is an approximate duality theorem for multicommodity h-route cuts and flows, for $h \leq 3$: The size of a minimum h-route cut is at least f/h and at most $O(\log^3 k \cdot f)$ where f is the size of the maximum h-route flow and k is the number of commodities. The main step towards the proof of this duality is the design and analysis of a polynomial-time approximation algorithm for the minimum h-route cut problem for $h = 3$ that has an approximation ratio of $O(\log^3 k)$. Previously, polylogarithmic approximation was known only for h-route cuts for $h \leq 2$. A key ingredient of our algorithm is a novel rounding technique that we call multilevel ball-growing. Though the proof of the duality relies on this algorithm, it is not a straightforward corollary of it as in the case of classical multicommodity flows and cuts. Similar results are shown also for the sparsest multiroute cut problem.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases Multicommodity flow; Multiroute flow; Cuts; Duality

Digital Object Identifier 10.4230/LIPIcs.STACS.2011.129

1 Introduction

The celebrated maximum-flow minimum-cut theorem of Ford and Fulkerson [5] is among the most important results in combinatorial optimization. Its importance has influenced the search for various generalizations. In the *maximum multicommodity flow problem* the goal is to maximize the sum of flows between given source-sink pairs subject to capacity constraints. In the dual problem, namely in the *minimum multicut problem*, the objective is to find a subset of edges of minimum total capacity whose removal disconnects each of the given source-sink pairs. Though an exact duality theorem does not apply to these two problems, Garg et al. [6], building on an earlier work of Leighton and Rao [10] and of others, proved an approximate max-flow min-cut theorem; the approximation factor is logarithmic

* This research was partially supported by the grants GA ČR 201/09/0197, 1M0021620808 (ITI), and DFG SCHE 1592/1-1.

in the number of commodities and is asymptotically optimal. The results are proved using the ball-growing (also known as region-growing) technique that was introduced in the paper of Leighton and Rao.

Multi-route flows and multi-route cuts generalize in a natural way the concept of classical flows and cuts in graphs. An *elementary h-route flow*, for an integer $h \geq 1$, is a set of h edge-disjoint paths between a source and a sink, each path carrying a unit of flow, and an *h-route flow* [8, 1] is a non-negative linear combination of elementary h-route flows. An *h-route cut* is a set of edges whose removal disconnects a given source-sink pair with respect to h-route flows (in the multicommodity setting, it disconnects every source-sink pair). In other words, an h-route cut is a set of edges whose removal decreases the edge-connectivity of a given source-sink pair (or of every given source-sink pair) below h. Note that for $h = 1$, h-route flows and h-route cuts correspond to the classical flows and cuts.

1.1 Our results and techniques

The main result of this paper is an approximate duality theorem for multicommodity h-route cuts and flows for $h \leq 3$. In particular, we prove an upper bound of $O(\log^3 k \cdot f)$ on the size of a minimum h-route cut where f is the size of a maximum h-route flow and k is the number of source-sink pairs (or *commodities*); trivially, f/h is a lower bound.

A major step towards the proof of the duality in this paper is the design and analysis of an approximation algorithm for the minimum 3-route cut problem. The approximation ratio of our algorithm is $O(\log^3 k)$. This provides a partial answer to open problems of several papers (Bruhn et al. [3], Chekuri and Khanna [4] and Barman and Chawla [2]). The 3-route cut problem is more complicated than the 1-route and 2-route cut problems: while 1-route and 2-route cuts separate the graph into independent parts, h-route cuts do not have this property for $h > 2$. For example, when providing a 2-route cut C for the commodity (s_1, t_1) that partitions the graph into the node sets S_1 and T_1 with at most one remaining edge between them, then the commodities that have both nodes in S_1 or both in T_1 can be treated independently because no simple path can connect two nodes in S_1 (resp. T_1) via a path through T_1 (resp. S_1). This is not the case for 3-route cuts where a simple path between two nodes in S_1 may very well pass through T_1. A key ingredient to handle this problem in our paper is a novel rounding technique, called multilevel ball-growing, a generalization of the well-known ball growing argument that makes it possible to control the dependencies between parts of the graph that are separated by 3-route cuts.

Though the proof of the duality relies on the approximation algorithm, it is not a straightforward corollary of it as is the case for classical multicommodity flows and cuts. For the duality proof we show a tight relationship between two different linear relaxations [4, 2] of the h-route cut problem.

1.2 Other related results

The concept of multi-route flows was introduced by Kishimoto and Takeuchi [8]. As far as we know, the problem of a minimum h-route cut, for $h > 1$, was first considered by Bruhn et al. [3] in a paper dealing primarily with single source multi-route flows on graphs with uniform capacities. In this particular setting they established an approximate max-flow min-cut theorem and, as a corollary, described a $(2h-2)$-approximation algorithm for the minimum h-route cut problem, for any $h > 1$.

For graphs with non-uniform capacities, the first non-trivial approximation for multi-route cuts was given by Chekuri and Khanna [4]. They dealt with the special case of $h = 2$ and provided an $O(\log^2 n \log k)$-approximation for the 2-route cut problem where n is the number of vertices in G. As their algorithm is based on an LP relaxation that is dual to the LP for the maximum 2-route flow problem, an implicit corollary of their result is an approximate duality of 2-route flows and 2-route cuts. The approximation factor for 2-route cuts was recently improved by Barman and Chawla [2] who described an $O(\log^2 k)$-approximation for the 2-route cut problem. Their algorithm is based on a different linear programming relaxation that allows them to extend the classical (discrete) ball-growing (or region-growing) technique (cf. [10, 6, 11]) to 2-route cuts. In a subsequent work [9], using a combination of the multilevel ball-growing technique and other arguments, we proved an approximate duality theorem for multicommodity h-route cuts and flows for *any* h, on uniform capacity networks. A challenging open problem is to prove an analogous result for networks with general capacities.

2 Minimum h-Route Cut Problem

Suppose that we are given a minimum h-route cut problem for the graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}_+$ and with commodities $(s_1, t_1), \ldots, (s_k, t_k)$. If $F \subseteq E$ is an h-route cut for the instance, then for every commodity there exists a set F_i of at most $h - 1$ edges such that $F \cup F_i$ is a classical cut for the commodity i. With this observation, the integer LP for the minimum h-route cut problem can be stated as follows (by $\mathcal{P}_i$ we denote the set of all edge-simple paths in G between s_i and t_i):

$$\min \sum_{e \in E} c(e)x(e) \tag{1}$$

$$\begin{array}{rcll} \sum_{e \in p}(x(e) + x_i(e)) & \geq & 1 & \forall i \in [k], p \in \mathcal{P}_i \\ \sum_{e \in E} x_i(e) & \leq & h-1 & \forall i \in [k] \\ x(e) & \in & \{0,1\} & \forall e \in E \\ x_i(e) & \in & \{0,1\} & \forall i \in [k], \forall e \in E \end{array}$$

In order to find a good approximate solution for this ILP, we will look at its LP relaxation where $x(e) \in \{0, 1\}$ is replaced by $x(e) \geq 0$ and $x_i(e) \in \{0, 1\}$ is replaced by $x_i(e) \geq 0$. In the following, let the x- and x_i-values represent an optimal solution of this LP relaxation and let $\phi = \sum_{e \in E} c(e)x(e)$. Our goal is to round these values to an integral solution with cost at most $O(\phi \log^3 k)$ for $h = 3$. For this we will use a novel rounding technique that we call *multilevel ball-growing*. At the heart of this (as well as the classical ball growing) technique is the following lemma from elementary calculus.

▶ **Lemma 1.** *Let $[l_1, r_1], [l_2, r_2], \ldots, [l_z, r_z]$ be internally disjoint intervals of real numbers such that $l_1 < l_2 < \cdots < l_z$ and let $\mathcal{R} = \bigcup_{i=1}^{z} [l_i, r_i]$. Assume that the following holds:*
- *f is a nondecreasing function on $\mathcal{R}$ and $f(l_1) > 0$,*
- *f is differentiable on $\mathcal{R}$, except for finitely many points,*
- *g is a function on $\mathcal{R}$ such that $\forall r \in \mathcal{R}$, $g(r) \leq f'(r)$, except for finitely many points.*

Let $\gamma = f(r_z)/f(l_1)$. Then there exists $r \in \mathcal{R}$ such that $g(r) \leq \frac{1}{|\mathcal{R}|} \log \gamma \cdot f(r)$.

Proof. Assume, by contradiction, that for every $r \in \mathcal{R}$ we have $g(r) > \frac{1}{|\mathcal{R}|} \log \gamma \cdot f(r)$. Then

$$\log \gamma \leq \int_{r \in \mathcal{R}} \frac{1}{|\mathcal{R}|} \log \gamma \,\mathrm{d}r < \int_{r \in \mathcal{R}} \frac{g(r)}{f(r)} \,\mathrm{d}r \leq \int_{r \in \mathcal{R}} \frac{f'(r)}{f(r)} \,\mathrm{d}r \leq \log \frac{f(r_z)}{f(l_1)} = \log \gamma \;,$$

a contradiction. ◀

3 Single Source

Our algorithm for 3-route cuts works in iterations, and in iteration i some 3-route cut is found for some commodity i that does not yet have a 3-route cut. These 3-route cuts are added up to some final cut $F \subseteq E$. Our goal is to make sure that $c(F) = O(\log^3 k \sum_e c(e) \cdot x(e)) = O(\phi \log^3 k)$. We start with some basic notation for iteration i.

We define $d_y(u)$ as the length of the shortest path from t_i to the node u with respect to the length function $y : E \to \mathbb{R}_{\geq 0}$. For the definitions of the δ-sets in iteration i (see below) we view every edge $uv \in E$ as a segment consisting of two parts: an x-part of length $x(uv)$ followed (on the way from t_i) by an x_i-part of length $x_i(uv)$. Certainly, $x(uv) + x_i(uv) \geq |d_{x+x_i}(v) - d_{x+x_i}(u)|$ for every edge $uv \in E$ but for the definition of the δ-sets below it will be convenient to assume equality between the two quantities. To ensure the equality, we perform a minor temporary modification of the x and x_i values: if $x(uv) \leq |d_{x+x_i}(v) - d_{x+x_i}(u)|$ then we reduce $x_i(uv)$ to $|d_{x+x_i}(v) - d_{x+x_i}(u)| - x(uv)$, otherwise we reduce $x(uv)$ to $|d_{x+x_i}(v) - d_{x+x_i}(u)|$ and set $x_i(uv) = 0$. These adjustments are *only* valid for the following definitions.

In iteration i, for any $r \in [0,1]$ we define

$$\begin{aligned}
B(r) &= \{u \in V \mid d_{x+x_i}(u) \leq r\} \\
\delta(r) &= \{uv \in E \mid d_{x+x_i}(u) \leq r < d_{x+x_i}(v)\} \\
\delta_x(r) &= \{uv \in \delta(r) \mid d_{x+x_i}(u) \leq r \leq d_{x+x_i}(u) + x(uv)\} \\
\delta_{x_i}(r) &= \{uv \in \delta(r) \mid d_{x+x_i}(v) - x_i(uv) < r \leq d_{x+x_i}(v)\}
\end{aligned}$$

In words, the set $B(r)$, called a *ball* (or region) with center at t_i and radius r, is the set of nodes at distance at most r from t_i (with respect to $x + x_i$); $\delta(r)$ is the set of edges in the cut between $B(r)$ and $V \setminus B(r)$, $\delta_x(r)$ is the subset of edges from the cut $\delta(r)$ that are cut in their x-part, and $\delta_{x_i}(r)$ are those from $\delta(r)$ that are cut in their x_i-part. Clearly, $\delta(r) = \delta_x(r) \cup \delta_{x_i}(r)$. We denote by $\delta_1(r)$ the set $\delta(r)$ without the most expensive edge (i.e., $\delta_1(r) = \delta(r) \setminus \{\operatorname{argmax}_{e \in \delta(r)} c(e)\}$), and for $l > 1$ we denote by $\delta_l(r)$ the set $\delta_{l-1}(r)$ without the most expensive edge (i.e., $\delta_l(r) = \delta_{l-1}(r) \setminus \{\operatorname{argmax}_{e \in \delta_{l-1}(r)} c(e)\}$). Note that for every $r \in [0,1]$, the set $\delta_{h-1}(r)$ is an h-route cut between t_i and s. For a set $E' \subseteq E$ of edges we define $c(E') = \sum_{e \in E'} c(e)$. For a graph (resp. node set) H, let $V(H)$ be the set of nodes and $E(H)$ be the set of edges in H (resp. the set of edges in E that have both endpoints in H).

3.1 2-Route Cuts

To outline our general approach in a simple setting, we sketch in this subsection an alternative proof of the known result for 2-route single-source cuts.

In iteration i we define $\mathcal{R} = \{r \in [0,1] \mid |\delta_{x_i}(r)| \leq 1\}$ and observe that the measure of this set is at least $1/2$. For $r \in [0,1]$, let $f(r) = \phi/k + \int_{\rho \in \mathcal{R} \cap [0,r]} c(\delta_x(\rho))\, \mathrm{d}\rho$ and $g(r) = c(\delta_1(r))$ where ϕ denotes the optimal objective value of the LP relaxation. The functions f (volume) and g (cut size) satisfy the assumptions of Lemma 1 and thus, there exists $r \in \mathcal{R}$ such that $c(\delta_1(r)) = O(\log k) f(r)$. This is the key observation of Barman and Chawla [2] (proved in a different way). We add the edges from $\delta_1(r)$ to the 2-route cut that we construct, remove the ball $B(r)$ from the graph (observe that after the removal of $\delta_1(r)$, no terminal t_j in $B(r)$ is 2-connected with s) and proceed with the next iteration. The relationship between $c(\delta_1(r))$ and $f(r)$ makes it possible to charge the cost of the edges in $\delta_1(r)$ to the volume $f(r)$ of the ball $B(r)$ (cf. the analysis of the classical 1-route cut algorithm [11]). This immediately yields the $O(\log k)$-approximation for the 2-route single-source cut problem and, with some effort, also the $O(\log^2 k)$-approximation for the general 2-route cut problem.

3.2 3-Route Cuts

In contrast to the cases $h \leq 2$, for $h = 3$ we will need to charge more than one cut to some edges. In order to keep track of how many cuts were already charged to which edge, we maintain for every edge e a counter called *a level of an edge*, denoted $\ell(e)$, which represents (an upper bound on) how many cuts were already charged to the edge e. The edges with positive level are called *restricted edges* and are maintained in a set D. Initially, the level of every edge is zero and $D = \emptyset$. Throughout the run of the algorithm, every edge $e \in D$ satisfies $x(e) \leq 1/(2h \log k)$ and $\ell(e) \leq L$, for $L = \log k$.

Recall that $F \subseteq E$ is the edge set in which we collect the edges for the final 3-route cut output by the algorithm. Whenever we have a statement holding for any h-route cut, we use the h instead of 3 so that it becomes clear which techniques only apply to $h = 3$ and which techniques could also be applied to larger h-values.

Consider the iteration of the algorithm in which we deal with the terminal t_i. For any edge $uv \in E$ let the *distance* of uv from t_i be defined as $d(uv) = \min\{d_{x+x_i}(u), d_{x+x_i}(v)\}$. We partition the edges from D into two subsets, according to their levels and their distance d from t_i:

$$D_1 = \{e \in D \mid d(e) \text{ is minimal among all } f \in D \text{ with } \ell(f) = \ell(e)\} \quad \text{and} \quad D_2 = D \setminus D_1$$

Ties are broken arbitrarily to ensure that there is at most one edge per level in D_1. Observe that for every edge $f \in D_2$ there exists an edge $e \in D_1$ with $\ell(e) = \ell(f)$.

A radius $r \in [0, 1]$ is *forbidden* if $|\delta_{x_i}(r)| > h - 1$ or if there exists an edge $e \in D_1$ such that $e \in \delta_x(r)$. A radius $r \in [0, 1]$ that is not forbidden is *good*. Let $\mathcal{R}$ denote the set of good radii for the current iteration, that is, $\mathcal{R} = \{r \in [0, 1] \mid \delta_{x_i}(r) \leq h - 1 \text{ and } \delta_x(r) \cap D_1 = \emptyset\}$.

▶ **Lemma 2.** *The measure of the set $\mathcal{R}$ of good radii is at least $1/(2h)$.*

Proof. Let μ be the measure of the set $\{r \in [0, 1] \mid |\delta_{x_i}(r)| \geq h\}$. Considering the constraint $\sum_{e \in E} x_i(e) \leq h - 1$ we obtain an upper bound on μ: $h\mu \leq \sum_{e \in E} x_i(e) \leq h - 1$, and thus, $\mu \leq 1 - 1/h$. Therefore the measure of the set $\{r \in [0, 1] \mid |\delta_{x_i}(r)| \leq h - 1\}$ is at least $1/h$. Since the number of edges in $\mathcal{D}_1$ is at most $\log k$ and since $x(e) \leq 1/(2h \log k)$ for every $e \in \mathcal{D}_1$, the measure of the set $\{r \in [0, 1] \mid \delta_x(r) \cap D_1 \neq \emptyset\}$ is at most $1/(2h)$. Hence, $|\mathcal{R}| \geq 1/(2h)$. ◀

Recall that ϕ is the optimal value of the objective function. For $r \in [0, 1]$, we define

$$V(r) = \frac{\phi}{k} + \int_{\rho \in \mathcal{R} \cap [0,r]} c(\delta_x(\rho)) d\rho \ .$$

The value $V(r)$ is called the *volume* of the ball $B(r)$. Observe that only the x-parts of the edges in the ball contribute to the volume and the x-parts of the edges from D_1 do not contribute.

Clearly, 2ϕ is an upper bound on any $V(r)$. Since $c(\delta_x(r)) \geq 0$ and $V(r) \geq \phi/k$ for all $r \in \mathcal{R}$ and $c(\delta_x(r))$ is a step function (i.e., a piece-wise constant function) on $\mathcal{R}$ with at most $2m$ jumps, where $m = |E|$, we obtain the following lemma.

▶ **Lemma 3.** *The function $V(r)$ satisfies the following properties:*

- *$V(r)$ is a nondecreasing piece-wise linear function on $\mathcal{R}$ and $V(r) > 0$ for all $r \in \mathcal{R}$,*
- *$V(r)$ is differentiable on $\mathcal{R}$, except for finitely many points,*
- *for each $r \in \mathcal{R}$, $V'(r) \geq c(\delta_{h-1}(r))$, except for finitely many points,*
- *the maximum ratio between two values of the function $V(r)$ on $\mathcal{R}$ is at most $2k$.*

Proof. Follows from the definitions of the set $\mathcal{R}$ and of the function $V(r)$. ◀

▶ **Lemma 4.** *There exists an $r \in \mathcal{R}$ such that $c(\delta_{h-1}(r)) \leq 2h\log(2k) \cdot V(r)$. Moreover, such a radius can be computed in polynomial time.*

Proof. By Lemma 2, we know that $|\mathcal{R}| \geq 1/(2h)$. Lemma 3 guarantees that we can apply Lemma 1 to the functions $f(r) = V(r)$ and $g(r) = c(\delta_{h-1}(r))$ on $\mathcal{R}$. Thus, there is an $r \in \mathcal{R}$ with $c(\delta_{h-1}(r)) \leq 2h\log(2k)V(r)$.

Since $V(r)$ is a piece-wise linear function on $\mathcal{R}$ and $c(\delta_{h-1}(r))$ is a piece-wise constant function on $\mathcal{R}$ with at most $2m$ pieces, we can efficiently find the value r for which $c(\delta_{h-1}(r))/V(r)$ is minimal, and by the first part of this lemma, this ratio is at most $2h\log(2k)$. ◀

In the current iteration, we first compute the radius r from Lemma 4 and add the edges from $\delta_2(r)$ to F (our final cut). Similar to the case of 2-route cuts, the relation between $c(\delta_2(r))$ and $V(r)$ (Lemma 4) makes it possible to charge the cost of the cut $\delta_2(r)$ to the volume $V(r)$ of the ball $B(r)$. Note that *nothing is charged to any edge* $e \in D_1$ since their x-parts do not contribute to $V(r)$. Before we proceed with the next iteration, we locally modify the graph G as described in the rest of this section.

Consider the set of edges in $\delta(r) \setminus \delta_2(r)$ and let Z be the set of endpoints of these edges that are *not* in $B(r)$. If $|Z| \leq 1$, we remove $B(r)$ and $E(B(r))$ from G and proceed with the next iteration. We can do so because for any $t_j \in B(r)$ we already constructed a 3-route cut, and for any $t_j \in V \setminus B(r)$ any path from t_j to s that goes through $B(r)$ can be reduced so that it does not contain any node from $B(r)$. If $|Z| = 2$, we define H to be a subgraph of G with vertex set $V(H) = B(r) \cup Z$ and edge set $E(H) = \{xy \in E(G) \mid x, y \in B(r)\} \cup (\delta(r) \setminus \delta_{h-1}(r))$. The two nodes in Z are called the *entry nodes* of H and the two edges in $\delta(r) \setminus \delta_{h-1}(r)$ the *entry edges* of H. Let v_i, w_i denote the two entry nodes of H, let $d_y(v_i, w_i, H)$ denote the length of the shortest path (with respect to the length function $y : E \to \mathbb{R}_{\geq 0}$) between v_i and w_i in H and let $\text{mincut}(v_i, w_i, H)$ denote the minimum cut between v_i and w_i in H. If $d_x(v_i, w_i, H) > 1/(2h\log k)$, we add the edges from $\text{mincut}(v_i, w_i, H)$ to F, charge the cost of this cut to the volume of H and remove the subgraph H from the current graph G (with the same justification as for $|Z| \leq 1$); by the *volume* of H we mean $\bar{V}(r) = \sum_{e \in E(H)} c(e)x(e)$.

▶ **Lemma 5.** *If $d_x(v_i, w_i, H) > 1/(2h\log k)$ then $c(\mathit{mincut}(v_i, w_i, H)) < 2h\log k \cdot \bar{V}(r)$.*

Proof. Suppose that $d_x(v_i, w_i, H) > 1/(2h\log k)$ and let $\gamma = c(\text{mincut}(v_i, w_i, H))$. Then it holds for all $\rho \in [0, d_x(v_i, w_i)]$ that $c(\delta(v_i, \rho)) > \gamma$ where $\delta(v_i, \rho)$ is the set of edges crossing distance ρ from v_i in H. Therefore,

$$\bar{V}(r) \geq \int_{\rho=0}^{d_x(v_i, w_i, H)} c(\delta(v_i, \rho)) \geq \gamma \cdot d_x(v_i, w_i) > \gamma/(2h\log k)$$

Hence, $c(\text{mincut}(v_i, w_i, H)) < 2h\log k \cdot \bar{V}(r)$. ◀

If $d_x(v_i, w_i, H) \leq 1/(2h\log k)$, we replace H in G by a new edge v_iw_i and set

$$\begin{aligned} x(v_iw_i) &= d_x(v_i, w_i, H) \ , \\ x_j(v_iw_i) &= d_{x+x_j}(v_i, w_i, H) - d_x(v_i, w_i, H) \ , \forall j > i, \\ c(v_i, w_i) &= c(\text{mincut}(v_i, w_i, H)), \\ \ell(v_iw_i) &= \max\{\ell_1, \ell_2\} \end{aligned} \tag{2}$$

where $\ell_1 = 0$ if $E(H) \cap D_1 = \emptyset$ and $\ell_1 = \max_{e \in E(H) \cap D_1} \ell(e)$ otherwise, and $\ell_2 = 0$ if $E(H) \cap D_2 = \emptyset$ and $\ell_2 = 1 + \max_{e \in E(H) \cap D_2} \ell(e)$ otherwise.

▶ **Lemma 6.** *Replacing the subgraph H by the new edge $v_i w_i$ as described above does not increase the total volume $\sum_e x(e)c(e)$ of the system. Moreover, after the replacement all constraints of the LP are satisfied.*

Proof. The product $d_x(v_i, w_i, H) \cdot c(\mathrm{mincut}(v_i, w_i, H))$ is a lower bound on the volume of H. The claim about the LP constraints is clear from the description of the replacement. ◀

We say that the new edge $v_i w_i$ *represents* the edges in $E(H)$ (to be more precise, $v_i w_i$ represents all edges that were represented by edges in $E(H)$; an edge from the original edge set represents itself). The new edge $v_i w_i$ is added to the set D of restricted edges and all edges in D that are incident to a node in $B(r)$ are removed from D. When some cut is charged later to this new edge $v_i w_i$, then the charge is redistributed recursively to the edges represented by $v_i w_i$, proportionally to their volume. When the edge $v_i w_i$ is cut later, then it means cutting all edges in $\mathrm{mincut}(v_i, w_i, H)$. With this convention, every 3-route cut in the modified graph corresponds to a 3-route cut of the same cost in the original graph. We observe several things.

▶ **Lemma 7.** *For each $e \in D$, every edge represented by e was charged at most $\ell(e)+1$ times due to the $\delta_2(r)$ cuts.*

Proof. By construction, every time something is charged to an edge f of level $\ell(f)$, either the edge is removed from the graph, or the level of the new edge that represents f is set to $\ell(f)+1$ at least. ◀

▶ **Lemma 8.** *For each $e \in D$, $\ell(e) \leq \log k$.*

Proof. By construction, the only possibility for an increase of the maximum level of edges in D is when the level of a new edge $v_i w_i$ is set to $\ell_2 \geq 1$, according to the definition (2). Note that in this case, at least two edges of level $\ell_2 - 1$ are removed from D (and from G). Since for every commodity i we add at most one edge to D, the claim follows. ◀

▶ **Lemma 9.** *For each $e \in D$, $x(e) \leq 1/(2h \log k)$.*

Proof. By the construction and the definition (2) of $x(v_i w_i)$. ◀

The lemmas above guarantee that the set D entering the next iteration satisfies our assumptions listed at the beginning of this section.

▶ **Theorem 10.** *The approximation ratio of the algorithm for the 3-route single-source cut problem is $O(\log^2 k)$.*

Proof. First of all, notice that if some $\mathrm{mincut}(v_i, w_i, H)$ is charged to an edge e, then e will be removed together with H from the system and never be charged again. Hence, Lemma 5 implies that the cost of the part of F that is due to mincuts is at most $O(\phi \log k)$.

It remains to bound the cost of the h-route cuts. By the construction, the cost of every h-route cut $\delta_2(r)$ is charged to the volume $V(r)$ of some ball $B(r)$. By Lemmas 4, 6, 7 and 8, the sum of volumes of all balls to which some h-route cut was charged is at most $O(\phi \log k)$. Thus, by Lemma 4 the total cost of the h-route cuts is at most $O(\phi \log^2 k)$. ◀

4 Multiple Sources

The algorithm for multiple sources is an extension of the single-source algorithm for $h = 3$. Again, it works in iterations. In iteration i the algorithm constructs the ball B around one of the terminals s_i and t_i. In contrast to the single-source problem, there might be commodities with both terminals *inside* B. To deal with these pairs, the algorithm is recursively run in the ball B, with levels re-initialized to 0. There are two main issues that must be addressed: the number of recursive calls working with the same part of the original graph, and the (in)dependence of the subproblems. A minor change from the single-source algorithm is that now we require that $x(e)\%le1/(6h\log k)$ for every $e \in D$.

4.1 Number of overlapping recursive calls.

There are two ways how two recursive calls may work in the same area of the original graph G: (i) One of the two calls is invoked inside the other call. (ii) When the recursive call for B is completed, B is replaced by an edge and the edge is later included in a new ball B' for which another recursive call is invoked.

To guarantee that the depth of the recursion is small, we ensure that every constructed ball contains at most half of the remaining commodities. Then the depth of the recursion is $\log k$ only. In this part of our algorithm and its analysis we use the ideas from the recent paper by Barman and Chawla [2]. Lemma 11 deals with this problem.

To guarantee that there are not too many later recursive calls working in a particular area of the original graph, we apply a lazy strategy: instead of invoking the recursive call immediately after the ball B is defined, the algorithm postpones the call. If the algorithm later defines another ball B' in which the recursive call is to be run, and the ball B (the edge created from B) is contained in B', it is sufficient to perform only the recursive call for B'; this call will take care also about all commodities inside B (note that every two balls are either disjoint, or one of them is contained in the other).

Before we state and prove Lemma 11 we need a few more definitions. To simplify them, in addition to the assumption made in the previous section (i.e., $x(uv) + x_i(uv) = |d_{x+x_i}(v) - d_{x+x_i}(u)|$ for each $uv \in E$), we assume, without loss of generality, that $d_{x+x_i}(s_i) = 1$. Then, for $r \in (0,1)$ and $z \in \{s_i, t_i\}$ we define $B^z(r) = \{u \in V \mid d_{x+x_i}(z,u) \le r\}$, $\delta^{t_i}(r) = \delta(r)$, $\delta^{s_i}(r) = \delta(1-r)$, $\delta^{t_i}_x(r) = \delta_x(r)$, and $\delta^{s_i}_{x_i}(r) = \delta_{x_i}(1-r)$ where for each $u \in V$, $d_{x+x_i}(t_i,u) = d_{x+x_i}(u)$ and $d_{x+x_i}(s_i,u) = 1 - d_{x+x_i}(u)$. For $z \in \{s_i,t_i\}$ we also define $\delta^z_1(r) = \delta^z(r) \setminus \{\mathrm{argmax}_{e\in\delta^z(r)} c(e)\}$, $\delta^z_2(r) = \delta^z_1(r) \setminus \{\mathrm{argmax}_{e\in\delta^z_1(r)} c(e)\}$ and

$$\begin{aligned} D^z_1 &= \{uv \in D \mid \ell(uv) = -1 \text{ or } d(z,uv) \text{ is minimal among all } e \in D \text{ with } \ell(e) = \ell(uv)\} , \\ D^z_2 &= D \setminus D^z_1 . \end{aligned}$$

where $d(t_i, uv) = d(uv)$ and $d(s_i, uv) = 1 - d(t_i, uv)$. Finally, for $z \in \{s_i, t_i\}$ and $r \in [0,1]$, we define $V(z,r) = \phi/k + \int_{\rho\in\mathcal{R}^z\cap[0,r]} c(\delta^z_x(\rho))d\rho$, where $\mathcal{R}^z = \{r \in [0,1] \mid \delta_{x_i}(r) \le h-1,\ \delta^z_x(r) \cap D^z_1 = \emptyset\}$.

▶ **Lemma 11.** *There exist good radii r_s and r_t such that $r_s + r_t \le 1$,*

$$c(\delta_{h-1}(s_i, r_s)) \le 3h\log(2k) \cdot V(s_i, r_s) \quad \text{and} \quad c(\delta_{h-1}(t_i, r_t)) \le 3h\log(2k) \cdot V(t_i, r_t) .$$

Proof. From Lemma 2 it follows that the measure of the set $\{r \in [0,1] \mid |\delta_{x_i}(r)| \le h-1\}$ is at least $1/h$. Since the number of edges in D_1 is at most $2\log k$ and $x(e) \le 1/(6h\log k)$ for every $e \in D_1$, the measure of the radii forbidden due to edges in D_1 is at most $1/(3h)$. As $d_{x+x_i}(t_i, s_i) = 1$, there is a radius r so that $|\mathcal{R}^{s_i} \cap [0,r]| \ge 1/(3h)$ and $|\mathcal{R}^{t_i} \cap [r,1]| \ge 1/(3h)$. It follows from Lemma 1 that there is an $r_s \in \mathcal{R}^{s_i} \cap [0, r]$ with $c(\delta^{s_i}_{h-1}(r_s)) \le 3h \log(2k) \cdot V(s_i, r_s)$ and an $r_t \in \mathcal{R}^{t_i} \cap [r, 1]$ with $c(\delta^{t_i}_{h-1}(r_t)) \le 3h \log(2k) \cdot V(t_i, r_t)$. Since $r_s + r_t \le 1$, the lemma follows. ◀

If r_s and r_t are the radii from Lemma 11 then the sets $B^{s_i}(r_s)$ and $B^{t_i}(r_t)$ are disjoint; thus at least one of them contains at most half of the remaining commodities. We always pick such a ball in our algorithm.

▶ **Corollary 12.** *The depth of the recursion is at most* $\log k$.

4.2 Independence of the Balls

Note that without some special care, the recursive subproblems are *not* independent as the inner part of every ball B is connected to the outside part by two edges. This is in contrast to the case $h = 2$ where the two parts of the graph are connected by a single edge and thus can be treated independently in order to deal with those commodities with both terminals in the same part of the graph. A new type of edges, *forbidden edges*, will help us to control the dependencies.

In the algorithm for the single-source multi-route cut, the input for iteration i consists not only of the current graph with the set of commodities and the corresponding fractional solution of the linear program but also of the set of restricted edges inside of this graph, which helps us to control the dependencies between the iterations. For multiple sources, besides the restricted edges, we will also use the *forbidden* edges. Formally, they will be part of the set of restricted edges but their level will be -1 and the restrictions imposed on them are stronger: they are never cut (be it an h-route cut or a mincut) and they are never charged for any cut.

Assume that we plan to invoke a recursive call for a ball built around the terminal $z \in \{s_i, t_i\}$ with radius r. We distinguish two cases (as in the previous section, v_i and w_i denote the two entry nodes of H):

- If $d_x(v_i, w_i, G \setminus H^z) \le 1/(6h \log k)$, the recursive call is invoked for the subgraph H^z with an extra edge $v_i w_i$ with $x(v_i w_i) = d_x(v_i, w_i, G \setminus H^z)$, $c(v_i w_i) = c(\text{mincut}(v_i, w_i, G \setminus H^z))$ and level -1. For each $j > i$ we also set $x_j(v_i w_i) = d_{x+x_j}(v_i, w_i, G \setminus H^z) - x(v_i w_i)$. The set of commodities consists of those with both terminals in $B^z(r)$, and the set of restricted edges is $(D \cap E(H^z)) \cup \{v_i w_i\}$. Since the x-length of the new edge is very short and each recursion creates at most one such edge, it is possible to impose such restrictions.
- If $d_x(v_i, w_i, G \setminus H^z) > 1/(6h \log k)$, for the recursive call we use the ball $B^z(r)$s. The set of commodities consists again of those with both terminals in $B^z(r)$, and the set of edges with restriction is $D \cap E(H^z)$. The pair $\{v_i, w_i\}$ is added to the set $\mathcal{T}$. At the very end of the algorithm, we disconnect all pairs that are in $\mathcal{T}$.

4.3 Putting it Together

Similarly to the singe-source version of the algorithm, for every part of the set F that the algorithm constructs, the ratio between the cost and the volume is bounded by $O(\log k)$, and to each part of the volume we charge at most $O(\log k)$ times within each recursive call. The only problem is that the set F that was constructed so far need not be a valid h-route cut. The difficulty is with the recursion.

In the recursive calls, when the distance $d_x(v_i, w_i, G \setminus H^z)$ was large (see the previous subsection), we ignored the fact the v_i and w_i were possibly connected *outside* the ball $B^z(r)$. Thus, at this point we have no guarantee that the set F that we constructed so far is a

3-route cut. To ensure that we do have a 3-route cut, we remove an additional set of edges from the graph. To be more specific, it suffices to disconnect the pairs of vertices in $\mathcal{T}$.

We proceed as follows. Consider the instance of the classical multicut problem consisting of the graph $G \setminus D$ and of the set $\mathcal{T}$ that represents the commodities. By construction of the set $\mathcal{T}$, the x-distance between terminals of every pair in $\mathcal{T}$ is at least $1/(6h \log k)$. Thus, if we scale the x-values by $6h \log k$, we get a fractional solution for the multicut problem for this instance. We apply the classical ball-growing rounding algorithm [6] to obtain an $O(\log k)$-approximation of the minimum multicut for this instance. Due to the scaling, the cost of the obtained cut is upper-bounded by $O(h \log^2(k) \cdot \phi)$. We add all edges from this cut to the set F.

Considering the explanation at the beginning of this section and the bound from the previous paragraph, the cost of the set F is $O(\log^3 k \cdot \phi)$, and at this point, F is a valid 3-route cut. The main theorem follows.

▶ **Theorem 13.** *The approximation ratio of the algorithm for the general* 3*-route cut problem is* $O(\log^3 k)$*.*

5 Duality of Multicommodity Multiroute Flows and Cuts

Recall that an *elementary* h*-flow* between s and t is a set of h edge-disjoint paths between s and t, each carrying a unit flow. Let $\mathcal{Q}_i$ denote the set of all elementary h-flows between s_i and t_i and let $\mathcal{Q} = \bigcup_{i=i}^{k} \mathcal{Q}_i$. Then the problem of finding a maximum multicommodity h-route flow has the following linear programming formulation; there is a non-negative variable $f(q)$ for every $q \in \mathcal{Q}$ where the value $f(q)$ represents the total amount of flow sent along the h-route flow q. On the right side of the page we state the dual linear program.

$$\begin{aligned} \max \sum_{q \in \mathcal{Q}} f(q) & & & \qquad (3) \\ \sum_{q \in \mathcal{Q}: e \in q} f(q) &\leq h \cdot c(e) & \forall e \in E \\ f(q) &\geq 0 & \forall q \in \mathcal{Q} \end{aligned}$$

$$\begin{aligned} \min\ h \cdot \sum_{e \in E} c(e) \cdot x(e) & & & \qquad (4) \\ \sum_{e \in q} x(e) &\geq 1 & \forall q \in \mathcal{Q} \\ x(e) &\geq 0 & \forall e \in E \end{aligned}$$

Note that without the factor h in the objective function the linear program (4) is another relaxation of the h-route cut problem (the approximation algorithm of Chekuri and Khanna [4] for 2-route cuts is based on this relaxation). We will refer by (4') to the linear program (4) with the objective function scaled down to $\sum_{e \in E} c(e) \cdot x(e)$.

There are simple examples showing that the linear relaxation (4') is by a factor of h lower (asymptotically) than the linear relaxation of (1). Think about two vertices s and t connected by M parallel edges. Then the fractional optimum for the linear program (4') is M/h (assign a value $1/h$ to every variable) while the fractional optimum of the linear program (1) is $M - h$.

The main technical result of this section is that the gap between the two relaxations is not more than h. A corollary of this result is an approximate duality theorem for multiroute cuts and flows.

▶ **Theorem 14.** *Given an instance of the* h*-route cut problem, let* O_1 *denote the optimum value of the linear program (1) and* O_2 *the optimum value of the linear program (4'). Then* $O_2 \leq O_1 \leq h \cdot O_2$*, and the bound is tight.*

Proof. Since the first inequality is trivial, it suffices to prove the second one. Let x be an optimum solution of the linear program (4'). We are going to derive from x a solution $\bar{x}, x_1, \ldots, x_k \in \mathbb{R}^E$ of the linear program (1) with the objective value being larger by a factor of at most h (i.e., $\sum_{e\in E} c(e)\bar{x}(e) \le h \sum_{e\in E} c(e)x(e)$). For each $e \in E$, let $\bar{x}(e) = h \cdot x(e)$. It suffices to prove that for each i, the following linear program has a feasible solution x_i. As in Section 2, $\mathcal{P}_i$ denotes the set of all paths between s_i and t_i.

$$\begin{aligned} \sum_{e\in p} x_i(e) &\ \ge\ 1 - \sum_{e\in p} \bar{x}(e) \quad \forall p \in \mathcal{P}_i \\ \sum_{e\in E} x_i(e) &\ \le\ h-1 \quad \forall e \in E \\ x_i(e) &\ \ge\ 0 \quad \forall e \in E \end{aligned} \tag{5}$$

Assume, for a contradiction, that the linear program (5) does not have a feasible solution. Then, by Farkas' lemma, there exists a non-negative vector $\lambda \in \mathbb{R}^{\mathcal{P}_i}$ and a non-negative scalar γ such that

$$\begin{aligned} \sum_{p\in\mathcal{P}_i : e\in p} \lambda(p) &\ \le\ 1 \quad \forall e \in E \\ \sum_{p\in\mathcal{P}_i} \lambda(p)(1 - \sum_{e\in p} \bar{x}(e)) &\ >\ h-1 \end{aligned} \tag{6}$$

(without loss of generality, we assume that $\gamma = 1$; note that every vector (λ, γ) obtained by the application of the Farkas' to the linear program (5) satisfies $\gamma > 0$ and thus, we can scale the (λ, γ) to guarantee $\gamma = 1$). In the following discussion, among all vectors λ satisfying the constraints (6) we fix the one for which $\sum_{p\in\mathcal{P}_i} \lambda(p)$ is minimal.

Observe that λ corresponds to a feasible flow between s_i and t_i in the graph G with all edge capacities set to one; the size of the flow is at least $h-1+\sum_{p\in\mathcal{P}_i}\sum_{e\in p}\lambda(p)\bar{x}(e) > h-1$. For each edge $e \in E$, let $\lambda(e) = \sum_{p:e\in p} \lambda(p)$ and let $E' = \{e \in E \mid \lambda(e) > 0\}$ be the subset of edges on which the flow λ is non-zero. Since the flow is realized in a graph with unit capacities and the size of the flow is strictly larger than $h-1$, by Mengers' theorem there exist h edge disjoint paths between s_i and t_i in (V, E'); let $q \in \mathcal{Q}_i$ denote the corresponding elementary h-flow and $\lambda(q) = \min_{e\in q} \lambda(e)$. Let $\lambda' \in \mathbb{R}^{\mathcal{P}_i}$ be (a path-decomposition of) the flow obtained from the flow λ by subtracting $\lambda(q)$ units of flow from every edge $e \in q$. Note that $\sum_{p\in\mathcal{P}_i} \lambda(p) > \sum_{p\in\mathcal{P}_i} \lambda'(p)$. Since we started with a feasible solution x of the linear program (4'), from the definition of $\bar{x}$ we know that $\sum_{e\in q} \bar{x}(e) \ge h$. Observing that

$$\sum_{p\in\mathcal{P}_i} \lambda(p)(1 - \sum_{e\in p}\bar{x}(e)) = \sum_{p\in\mathcal{P}_i} \lambda'(p)(1 - \sum_{e\in p}\bar{x}(e)) + \lambda(q)(h - \sum_{e\in q}\bar{x}(e)) \ ,$$

we conclude that $\sum_{p\in\mathcal{P}_i} \lambda'(p)(1 - \sum_{e\in p}\bar{x}(e)) > h-1$. However, this is a contradiction with the choice of λ: the flow λ' also satisfies the constraints (6) and its size is smaller than the size of λ. Thus, the linear program (5) has a feasible solution, for each i, and the proof is completed. ◀

▶ **Corollary 15** (Duality of multiroute multicommodity flows and cuts). *For any instance with k commodities, the cost of the minimum h-route cut for $h \le 3$ is at least a fraction $1/h$ of the maximum h-route multicommodity flow, and is always at most $O(h^2 \log^3 k)$ times as much.*

Proof. The first relation is trivial: one always has to block at least one of the h paths of every elementary h-flow. The other relation follows from Theorem 14, the duality of the linear programs (3) and (4), and Theorem 13 (the approximation algorithm). ◀

5.1 Sparsest multiroute cut

The *sparsest multiroute cut problem* is a multiroute analog of the sparsest cut problem. By a combination of standard [7, 11] and our techniques we obtain the following results.

▶ **Theorem 16.** *The approximation ratio achievable in polynomial time for the multiroute sparsest cut problem with $h \le 3$ is $O(h^2 \log h \log^3 k \log D)$ where $D = \sum_{i=1}^{k} d_i$.*

▶ **Corollary 17.** *For any instance with k commodities, the sparsest h-route cut for $h \le 3$ is at least as large as the maximum concurrent h-route multicommodity flow, and is always at most $O(h^2 \log h \log^3 k \, \log D)$ times larger.*

Acknowledgments.

The first author would like to thank Jiří Sgall and Thomas Erlebach for stimulating discussions.

References

1 Amitabha Bagchi, Amitabh Chaudhary, Petr Kolman, and Jiří Sgall. A simple combinatorial proof for the duality of multiroute flows and cuts. Technical Report 2004-662, Charles University, Prague, 2004.
2 Siddharth Barman and Shuchi Chawla. Region growing for multi-route cuts. In *Proceedings of the 21th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.
3 Henning Bruhn, Jakub Černý, Alexander Hall, Petr Kolman, and Jiří Sgall. Single source multiroute flows and cuts on uniform capacity networks. *Theory of Computing*, 4(1):1–20, 2008. Preliminary version in Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007.
4 Chandra Chekuri and Sanjeev Khanna. Algorithms for 2-route cut problems. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5125 of *Lecture Notes in Computer Science*, pages 472–484, 2008.
5 L. R. Ford and D. R. Fulkerson. Maximum flow through a network. *Canad. J. Math.*, 8:399–404, 1956.
6 Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.
7 Nabil Kahale. On reducing the cut ratio to the multicut problem. Technical report, 1993. DIMACS Technical report 93-78.
8 Wataru Kishimoto and M. Takeuchi. On m-route flows in a network. *IEICE Transactions*, J-76-A(8):1185–1200, 1993. (in Japanese).
9 Petr Kolman and Christian Scheideler. Approximate duality of multicommodity multiroute flows and cuts on uniform capacity networks. Technical report, 2010. Submitted.
10 Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, November 1999. Preliminary version in Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1988.
11 David B. Shmoys. Cut problems and their application to divide-and-conquer. In Dorith S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 192–235. PWS Publishing Company, 1997.