Theory of Computing Systems manuscript No.
(will be inserted by the editor)

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

Petr Kolman · Christian Scheideler

August 30, 2012

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 $O(\log^4 k \cdot f)$ 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 $O(\log^4 k)$. 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.

Keywords Multicommodity flow · Approximation algorithms · Duality

* P. Kolman was supported by the Center of Excellence – Institute for Theoretical Computer Science, Prague, project P202/12/G061 of GA ČR. C. Scheideler was supported by DFG SCHE 1592/1-1.

P. Kolman
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské nám. 25, 118 00 Prague, Czech republic
E-mail: kolman@kam.mff.cuni.cz

C. Scheideler
Dept. of Computer Science, University of Paderborn
Fürstenallee 11, 33102 Paderborn, Germany
E-mail: scheideler@upb.de

1 Introduction

The celebrated maximum-flow minimum-cut theorem of Ford and Fulkerson [9] 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. [10], building on an earlier work of Leighton and Rao [15] and of others, proved an approximate max-flow min-cut theorem; the approximation factor is logarithmic 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* [12,3] 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.

The main subject of this paper is the relation between the sizes of maximum multicommodity multiroute flows and minimum multicommodity multiroute cuts. Trivially, if f is the size of the maximum such flow, then f/h is a lower bound on the size of a minimum corresponding h-route cut. The harder part is to prove a good upper bound on the size of the cut.

For readers that are not familiar with classical multicommodity cuts, we recommend for background the survey paper by Shmoys [17] or the relevant chapters of textbooks by Vazirani [18] or Williamson and Shmoys [19].

1.1 Related results

Assume that we are given a graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}_+$. In a variant of a multicommodity flow, namely in the *maximum concurrent multicommodity flow*, there are k source-sink pair (s_i, t_i), called commodities, each of them associated with a certain demand $d_i \in \mathbb{R}_+$. The objective is to maximize f such that at least fd_i units of flow are simultaneously routed for every commodity, while obeying all capacity constraints. Its dual problem is the *sparsest cut problem*. Again, an approximate duality theorem holds for

these two problems, as shown by Aumann and Rabani [2] and Linial et al. [16]; the result is a corollary of a general theorem about embedability of metric spaces. Note that for $k = 1$, both variants coincide with the classical problem of single commodity flows and cuts. Using linear programming, both variants of multicommodity flow are solvable in polynomial time while both variants of the multicommodity cuts are NP-hard [8].

The concept of multi-route flows was introduced by Kishimoto and Takeuchi [12]. The problem of finding a maximum (multicommodity) multi-route flow can be formulated using linear programming and is solvable in polynomial time. For a single commodity, Kishimoto Takeuchi [12] also described a combinatorial polynomial-time algorithm for the maximum multi-route flow (cf. [1]) and showed a duality theorem for a non-standard definition of a cut size.

As far as we know, the problem of finding a minimum h-route cut, for $h > 1$, was first considered by Bruhn et al. [6] 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 [7]. 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. For this they reduced the 2-route cut problem to a 1-route cut problem (with a different set of commodities) and then used a standard rounding algorithm. 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 [5] 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. [15,10,17]) to 2-route cuts. A major challenge was to cope with the fact that the balls constructed by the algorithm may contain other commodities.

1.2 Our results and techniques

The main result of this paper is an approximate duality theorem for multicommodity 3-route cuts and flows. In particular, we prove an upper bound of $O(\log^4 k \cdot f)$ on the size of a minimum 3-route cut where f is the size of a maximum 3-route flow and k is the number of source-sink pairs (or *commodities*).

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 $O(\log^4 k)$. This provides a partial answer to open problems of several papers (Bruhn et al. [6], Chekuri and Khanna [7] and Barman and Chawla [5]). 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 (Fig. 1a), 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 (Fig. 1b). 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.

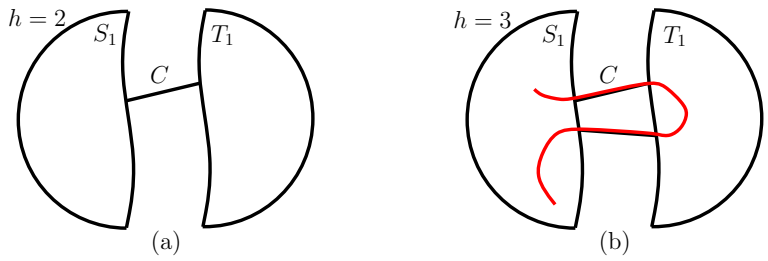

Fig. 1 A figure depicting the difference between 2-route and 3-route cuts.

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 programming relaxations [5,7] of the h-route cut problem; our main tool here is the Farkas' lemma.

In a subsequent work, the techniques from this paper were extended and applied to the multicommodity multiroute single source cut problem with any value of the parameter h, yielding a polynomial time algorithm with approximation ratio polylogarithmic in k, and also an approximate duality theorem in the same setting [14].

A preliminary version of this work was presented at the 28th International Symposium on Theoretical Aspects of Computer Science [13].

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):

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

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

In order to find a good approximate solution for this ILP, we will look at its LP relaxation where the constraint $x(e) \in \{0,1\}$ 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 $O(\phi \log^4 k)$ 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 (Ball Growing) *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. □

In Section 3 we describe an approximation algorithm for $h = 3$ for the single-source case (i.e., $s = s_1 = s_2 = \cdots = s_k$) and in Section 4 we extend the algorithm to the multiple-source case.

3 Single Source

At a high level, the structure of our approximation algorithm is the same as the structure of the approximation algorithm by Garg et al. [10] for classical multicommodity cuts (cf. [17,5,18,19]): while there exists a vertex t_i that is connected by three edge disjoint paths to s, perform a 3-route cut separating t_i and s, charge the 3-route cut to a certain part (volume) of the network (roughly, to the region that was used to define the cut) and proceed with the next iteration. The 3-route cut in each iteration is derived from the fractional solution of the LP relaxation of the minimum 3-route cut problem. These 3-route cuts are added up to some final cut $F \subseteq E$. Our goal is to make sure that $\sum_{e\in F} c(e) = O(\phi \log^2 k)$. We first introduce the notation that we use in the description of iteration i of our algorithm.

We define $d_y^H(u)$ as the length of the shortest path from t_i to the node u in a subgraph H of G, with respect to a length function $y : E \to \mathbb{R}_{\geq 0}$. If the subgraph H is clear from the context or unimportant, we omit the upper index H. 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 the following sets (Fig. 2):

$$\begin{aligned} B(r) &= \{u \in V \mid d^{G_i'}_{x+x_i}(u) \leq r\} \qquad (2)\\ \delta(r) &= \{uv \in E \mid d^{G_i'}_{x+x_i}(u) \leq r < d^{G_i'}_{x+x_i}(v)\}\\ \delta_x(r) &= \{uv \in \delta(r) \mid d^{G_i'}_{x+x_i}(u) \leq r < d^{G_i'}_{x+x_i}(u) + x(uv)\}\\ \delta_{x_i}(r) &= \{uv \in \delta(r) \mid d^{G_i'}_{x+x_i}(v) - x_i(uv) \leq r < d^{G_i'}_{x+x_i}(v)\} \end{aligned}$$

where G_i' is the graph that the algorithm works with in iteration i. Informally, 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 $x_i(uv)$. Then, 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 (Fig. 2). 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., $\delta_1(r) = \delta(r) \setminus \{\text{argmax}_{e\in\delta(r)} c(e)\}$), and for $l > 1$ we denote by $\delta_l(r)$ the set $\delta_{l-1}(r)$ without the most expensive edge (i.e., $\delta_l(r) = \delta_{l-1}(r) \setminus \{\text{argmax}_{e\in\delta_{l-1}(r)} c(e)\}$). 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 $c(E') = \sum_{e\in E'} c(e)$ and $V(E') = \bigcup_{uv\in E'} \{u,v\}$, that is, $c(E')$ is the sum of capacities of edges in E' and $V(E')$

is the set of nodes of the edges in E'. For a graph H, we denote by $V(H)$ the set of nodes and by $E(H)$ the set of edges in H.

3.1 Approximating 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. Assume that x and x_i, for $i = 1, \ldots, k$, is the optimal fractional solution of the linear programming relaxation of the 2-route cut problem, and that ϕ is the optimal objective value of the relaxation.

In iteration i we define $\mathcal{R} = \{r \in [0,1] \mid |\delta_{x_i}(r)| \leq 1\}$. The constraint $\sum_{e \in E} x_i(e) \leq 1$ from the LP(1) implies

$$1 \geq \int_{\rho \in [0,1]} |\delta_{x_i}(\rho)| \, \mathrm{d}\rho \geq \int_{\rho \in [0,1] \setminus \mathcal{R}} |\delta_{x_i}(\rho)| \, \mathrm{d}\rho \geq 2(1 - |\mathcal{R}|) \ ,$$

that is, the measure of the set $\mathcal{R}$ is at least $1/2$. For $r \in [0,1]$, let

$$f(r) = \phi/k + \int_{\rho \in \mathcal{R} \cap [0,r]} c(\delta_x(\rho)) \, \mathrm{d}\rho \quad \text{and}$$
$$g(r) = c(\delta_1(r)) \ .$$

Observe that $g(r) \leq c(\delta_x(r))$ for every $r \in \mathcal{R}$. Thus, the functions f (volume) and g (cut size) satisfy the assumptions of Lemma 1 and 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 [5] (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 [17]). This

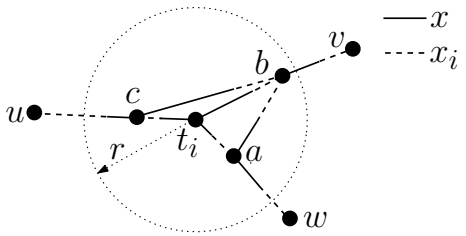

Fig. 2 Assume that a subgraph of a graph G consists of vertices t_i, a, b, c, u, v, w and edges $t_i a, t_i b, t_i c, ab, aw, bc, bv, cu$. We depict every edge e by a concatenation of a solid and a dashed lines such that the length of the solid line is proportional to $x(e)$ and the length of the dashed line is proportional to $x_i(e)$. For a radius r depicted in the picture, the following sets correspond to the definition (2): $B(r) = \{t_i, a, b, c\}$, $\delta(r) = \{aw, bv, cu\}$, $\delta_x(r) = \{bv\}$, $\delta_{x_i}(r) = \{aw, cu\}$.

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 Approximating 3-Route Cuts

In this section, some of the claims hold for any h but some for $h \leq 3$ only. Whenever we have a statement holding for any value of h, we use the letter h in the statement instead of the number 3 that we use in claims specific for $h = 3$. We define $L = \lfloor \log(k+1) \rfloor$.

Given a connected subgraph H of G, we say that a vertex $u \in V(H)$ is an *entry node* of H with respect to G if there exists a vertex $v \in V(G) \setminus V(H)$ with $uv \in E(G)$. An edge $uw \in E(H)$ is an *entry edge* of H with respect to G if $u, w \in V(H)$ and u or w is an entry node of H with respect to G. If H is a subgraph of G with two entry nodes v and w, we denote by $d_y(v, w, H)$ the length of the shortest path (with respect to a length function $y : E \to \mathbb{R}_{\geq 0}$) between v and w in H and by $\text{mincut}(v, w, H) \subset E(H)$ the minimum cut between v and w in H. We make a simple observation.

Lemma 2 *Assume that H is a subgraph of G with two entry nodes v and w. If $d_x(v, w, H) \geq 1/\beta$ for some $\beta \geq 1$, then $c(\text{mincut}(v, w, H)) \leq \beta \cdot \sum_{e \in E(H)} c(e)x(e)$.*

Proof Suppose that $d_x(v_i, w_i, H) \geq 1/\beta$ and let $\gamma = c(\text{mincut}(v_i, w_i, H))$. Then it holds for all $\rho \in [0, d_x(v_i, w_i, H)]$ that $c(\delta(v_i, \rho)) \geq \gamma$ where $\delta(v_i, \rho) = \{ab \in E(H) \mid d_x(v_i, a) \leq \rho \text{ and } d_x(v_i, b) > \rho\}$. Therefore,

$$\sum_{e \in E(H)} c(e)x(e) \geq \int_{\rho=0}^{d_x(v_i, w_i, H)} c(\delta(v_i, \rho)) \, \mathrm{d}\rho \geq \gamma \cdot d_x(v_i, w_i, H) \geq \gamma/\beta \; .$$

Hence, $c(\text{mincut}(v_i, w_i, H)) \leq \beta \cdot \sum_{e \in E(H)} c(e)x(e)$, and the proof is completed.

Let us briefly mention an alternative proof of the lemma. The vector $x' = \beta \cdot x$ represents a feasible fractional cut between v and w in H. The strong duality theorem of linear programming and the maximum-flow minimum-cut theorem imply that there exists an integral cut between v and w in H of the same size, that is, of size $\beta \cdot \sum_{e \in E(H)} c(e)x(e)$. □

As we already pointed out, the algorithm iteratively produces the desired multiroute multicut. In contrast to the cases $h = 1$ and $h = 2$, 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, our algorithm will maintain for every edge e a counter called the *level of an edge*, which represents an upper bound on how many cuts were already charged to the edge e. We denote by $\ell_i(e)$ the level of the edge e at the end of iteration i. The edges with positive level are called *restricted edges*. Initially, $\ell_0(e) = 0$ of every edge $e \in E$. We will prove later (Lemma 6) that for every edge $e \in E$ and every iteration i, $\ell_i(e) \leq L$.

Algorithm 3-RouteCutSingleSource($G = (V, E), s, t_1, \ldots, t_k$)
$F := \emptyset$, $\bar{F} = \emptyset$, $C := \emptyset$, $\mathcal{D} := \emptyset$, $\ell(e) := 0$ for every $e \in E$
while there is 3-connected (s, t_i) **do**
  Contract all $H \in \mathcal{D}$ into a set D of super edges
  $D_1 := \{uv \in D \mid d(uv)$ is minimal among all $e \in D$ with $\ell(e) = \ell(uv)$
    and uv is lex. minimal in case there are multiple edges uv
    with same level and minimal distance$\}$
  $D_2 := D \setminus D_1$
  $r := \min\{\rho \in [0, 1] \mid \rho$ is good and $c(\delta_2(\rho)) \leq 6 \log(2k) \cdot V(\rho)\})$
  $F := F \cup \delta_2(r)$
  $V := V \setminus B(r)$
  $E := E \setminus \{uv \in E \mid u \in B(r)$ or $v \in B(r)\}$
  $D := D \cap E$
  Let H be the extended ball $H(r)$ and v_i, w_i its entry nodes
  if $v_i \neq w_i$ **then**
    if $c(\text{mincut}(v_i, w_i, H)) \leq 6L \cdot \sum_{e \in E(H)} c(e)x(e)$ **then**
      $\bar{F} := \bar{F} \cup \text{mincut}(v_i, w_i, H)$
    else
      $E := E \cup \{v_i w_i\}$
      $D := D \cup \{v_i w_i\}$
      $\mathcal{D} := \mathcal{D} \cup \{H\}$
      $x(v_i w_i) := d_x(v_i, w_i, H)$
      $x_j(v_i w_i) := d_{x+x_j}(v_i, w_i, H) - x(v_i, w_i), \forall j \neq i$
      $c(v_i w_i) := c(\text{mincut}(v_i, w_i, H))$
      $\ell(v_i w_i) := \max\{1 + \max_{e \in E(H) \cap D_2} \ell(e),\ \max_{e \in E(H) \cap D_1} \ell(e)\}$
  else
    $C := C \cup E(H)$
    $E := E \setminus E(H)$
  Replace the super edges in F and $\bar{F}$ by their mincuts, remove them from D,
  and add the edges of their subgraphs that are outside of $H(r)$ to C
  For any entry edge of $H(r)$ that is a super edge e, add the subgraph corresponding
  to e to $H(r)$ and choose its outside entry edge as new entry edge for $H(r)$
  Undo the contraction of all other super edges in D
return $F \cup \bar{F}$

Fig. 3 The 3-Route Cut Algorithm for the single source problem. We use ℓ instead of ℓ_i in this description as the indices are used only in the analysis of the algorithm and are not important for the algorithm itself. Similarly, we overload the symbols V and E to denote not only the node and edge sets of the input graph but also the current graph in every iteration.

The algorithm is summarized in Figure 3. At the start of iteration i, the algorithm has already produced subsets of edges $F_1, F_2, \ldots, F_{i-1}, \bar{F}_1, \bar{F}_2, \ldots, \bar{F}_{i-1} \subset E$ with the property that $\bigcup_j^{i-1}(F_j \cup \bar{F}_j)$ is a single source h-route multicut for s and $t_1, \ldots, t_{i-1}$. The algorithm has also temporarily pruned away some edge set $C \subseteq E$ which is initially empty. Hence, at the beginning of iteration i we are left with the graph

$$G_i = \left(V, E \setminus (C \cup \bigcup_j^{i-1}(F_j \cup \bar{F}_j)) \right) .$$

Apart from these sets, also the following objects are given at the start of iteration i:

- A collection $\mathcal{D}$ of pairwise disjoint, connected induced subgraphs of G_i, each with two entry nodes, and
- a level $\ell_{i-1}(e)$ for every edge in G.

The key properties maintained by the algorithm at the start of iteration i are the following:

(a) None of the terminals $t_1, \ldots, t_{i-1}$ is h-connected with s in G_i.
(b) For all $H, H' \in \mathcal{D}$ with $H \neq H'$, $E(H) \cap E(H') = \emptyset$.
(c) For all $H \in \mathcal{D}$ with entry nodes v and w, $d_x(v, w, H) \leq 1/(6L)$.
(d) $\sum_{j=1}^{i-1} \sum_{e \in F_j \cup \bar{F}_j} c(e) = O(\log k) \sum_{e \in E} \ell_{i-1}(e) x(e) c(e)$.
(e) If $e \in E$ does not belong to any $H \in \mathcal{D}$, then $\ell_{i-1}(e) = 0$.

We are ready to start the description of the iteration i in which we deal with the terminal t_i. The input for the iteration i is the graph G_i. We start the iteration by replacing every subgraph $H \in \mathcal{D}$ with entry nodes v and w by a simple edge vw with the following parameters:

$$\begin{aligned} x(vw) &= d_x(v, w, H) \\ x_i(vw) &= d_{x+x_i}(v, w, H) - d_x(v, w, H) \ , \\ c(v, w) &= c(\text{mincut}(v, w, H)), \\ \ell_{i-1}(vw) &= \max_{e \in E(H)} \ell_{i-1}(e) \ . \end{aligned} \tag{3}$$

Every such edge vw is called a *super edge*. Let G_i' denote the resulting graph. In this part of the algorithm it is important that every $H \in \mathcal{D}$ has just two entry nodes (and not more), a property following from the value (three) of h. The replacement is not substantial for the algorithm but it significantly simplifies the presentation. Note that the vectors x and x_i represent a fractional multiroute cut for the commodity i, that is, $d^{G_i'}_{x+x_i}(t_i) \geq 1$ and $\sum_{e \in E(G_i')} x_i(e) \leq h - 1$.

Let $D = \{e \in E(G_i') \mid \ell_{i-1}(e) > 0\}$. As we will see later, D is the set of all super edges in G_i', and the edges in D correspond exactly to the subgraphs in $\mathcal{D}$ (cf. property (e) above). To *undo the contraction of* a super edge vw means to replace it back by H; to *cut* a super edge vw means to replace it by $H \setminus \text{mincut}(v, w, H)$.

We are going to describe how to pick a radius of a ball around t_i for an h-route cut for t_i; this is the core part of the algorithm. Once we have the right radius, we easily obtain everything we need.

For any edge (or super edge) $uv \in E$ let the *distance* of uv from t_i be defined as $d_i(uv) = \min\{d^{G_i'}_{x+x_i}(u), d^{G_i'}_{x+x_i}(v)\}$. We partition the edges from D into two subsets according to their levels and their distance d from t_i:

$$\begin{aligned} D_1 &= \{e \in D \ \mid \ d_i(e) \text{ is minimal among all } f \in D \text{ with } \ell_{i-1}(f) = \ell_{i-1}(e)\} \\ D_2 &= D \setminus D_1 \end{aligned}$$

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 $d_i(e) \leq d_i(f)$ and $\ell_{i-1}(e) = \ell_{i-1}(f)$.

A radius $r \in [0,1]$ is *forbidden* if $|\delta_{x_i}(r)| > h-1$ (i.e., we would not obtain a valid h-route cut by cutting all edges in $\delta_x(r) = \delta(r) \setminus \delta_{x_i}(r)$) or if there exists an edge $e \in D_1$ such that $e \in \delta_x(r)$ (which we do not want to cut). 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,

$$\mathcal{R} = \{r \in [0,1] \mid |\delta_{x_i}(r)| \leq h-1 \text{ and } \delta_x(r) \cap D_1 = \emptyset\} . \tag{4}$$

Lemma 3 *If $x(e) \leq 1/(2hL)$ for every $e \in D_1$ and $\ell_{i-1}(e) \leq L$ for every edge $e \in E(G'_i)$, then the measure of the set $\mathcal{R}$ of good radii as defined above 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 D_1 is at most L and since $x(e) \leq 1/(2hL)$ for every $e \in D_1$, 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, that is, $\phi = \sum_{e \in E} c(e)x(e)$ for the optimal solution x of the linear program (1). For $r \in [0,1]$ we define

$$V(r) = \frac{\phi}{k} + \int_{\rho \in \mathcal{R} \cap [0,r]} c(\delta_x(\rho)) d\rho . \tag{5}$$

The value $V(r)$ is called the *volume* of the ball $B(r)$ with radius r. We observe that $\phi/k + \int_{\rho \in [0,1]} c(\delta_x(\rho))d\rho \leq \phi/k + \phi$. Thus, for every $r \in [0,1]$, 2ϕ is an upper bound on $V(r)$. Also note that only the x-parts of the edges in the ball $B(r)$ that are *not in* D_1 contribute to the volume $V(r)$.

Lemma 4 (Radius Choice) *If $x(e) \leq 1/(2hL)$ for every $e \in D_1$ and $\ell_{i-1}(e) \leq L$ for every edge $e \in E(G'_i)$, then there exists $r \in \mathcal{R}$ 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 We first observe that 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 $2k$.

The first two properties are obvious from the definition of $V(r)$. Concerning the third property, we note that for every r for which $V(r)$ is differentiable, $V'(r) \geq c(\delta_x(r))$, and since for every $r \in \mathcal{R}$ it holds $c(\delta_x(r)) \geq c(\delta_{h-1}(r))$, the claim follows. The last property follows from the trivial lower bound $V(0) \geq \phi/k$, the above mentioned upper bound $V(1) \leq 2\phi$, and the fact that f is nondecreasing.

By Lemma 3, we know that $|\mathcal{R}| \geq 1/(2h)$. The properties of the function $V(r)$ allow us to apply Lemma 1 to the functions $f(r) = V(r)$ and $g(r) = c(\delta_{h-1}(r))$ on $\mathcal{R}$. Thus, there exits an $r \in \mathcal{R}$ satisfying $c(\delta_{h-1}(r)) \leq 2h \log(2k) V(r)$.

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 $c(\delta_{h-1}(r)) \setminus V(r)$ is minimal, and by the first part of this lemma, this ratio is at most $2h \log(2k)$. □

The algorithm computes a radius r according to the previous lemma and initially sets $F_i := \delta_2(r)$, where each super edge in $\delta_2(r)$ is replaced by the mincut of the subgraph represented by it, and $\bar{F}_i := \emptyset$. Before we proceed, we need one more definition. We say that the *extended ball* $H(r)$ with radius r is the subgraph of the graph G'_i with its node set X and edge set Y defined as follows:

$$\begin{aligned} X &= B(r) \cup Z \quad \text{and} \\ Y &= \{xy \in E(G'_i) \mid x, y \in B(r)\} \cup (\delta(r) \setminus \delta_2(r)) \ , \ \text{where} \\ Z &= V(\delta(r) \setminus \delta_2(r)) \setminus B(r) \ . \end{aligned} \tag{6}$$

Note that the two edges in $\delta(r) \setminus \delta_2(r)$ are the entry edges of $H(r)$, with respect to $G_i \setminus F_i$, and the nodes in the set Z, that is, the endpoints of the entry edges that are not in $B(r)$, are the entry nodes of $H(r)$[1]. Certainly, the size of Z is at most two.

The next step of the algorithm depends on the number of entry nodes of the extended ball $H(r)$. There are two cases to distinguish: $|Z| = 1$ (Fig. 4a) and $|Z| = 2$ (Fig. 4b). If $|Z| = 1$, we add $E(H(r))$ to C and remove from $\mathcal{D}$

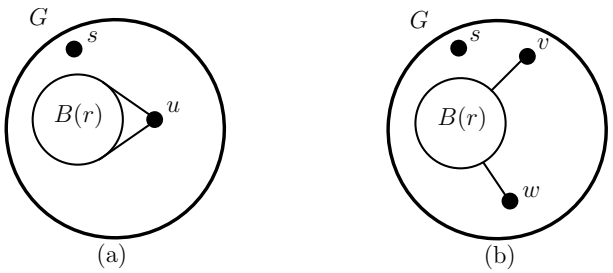

Fig. 4 The set Z has size 1 in the first case ($Z = \{u\}$) and size 2 in the second case ($Z = \{u, v\}$).

[1] To be precise, the entry edges of $H(r)$ are a subset of $\delta(r) \setminus \delta_2(r)$, and the entry nodes of $H(r)$ are a subset of Z: it may happen that (some of) the nodes in Z are not connected to an outside node (see the definition of entry nodes). However, such a pathological case makes the situation only easier and thus, without loss of generality, we assume that all nodes in Z are the entry nodes of $H(r)$.

subgraphs corresponding to the super edges in $H(r)$. The following lemma, whose proof is obvious, clarifies why we can prune all edges in $H(r)$.

Lemma 5 *Let H be any subgraph with at most one entry node. Then any path P between two vertices $v, w \notin V(H)$ can be reduced to a path P' that does not contain any edge of H. Also, any h-route cut Z between v and w, $v, w \notin V(H)$, can be reduced to an h-route cut $Z' \subseteq Z$ that does not contain any edge of H.*

Hence, $H(r)$ is irrelevant for the connectivity of any terminal outside of $H(r)$ and the source s (which is by construction outside of $H(r)$). Thus, we can remove it from G for the rest of the algorithm by adding its edges to C, and put it back at the very end without spoiling the multiroute cut.

If $|Z| = 2$, we distinguish between two further cases. If $d_x(v_i, w_i, H(r)) > 1/(6L)$, we add $\text{mincut}(v_i, w_i, H(r))$ to $\bar{F}_i$; by doing so, we cut $H(r)$ into two subgraphs of just one entry node each. Both of these subgraphs can then be pruned like for the case $|Z| = 1$, and all subgraphs from $\mathcal{D}$ corresponding to super edges contained in these two subgraphs of $H(r)$ can be removed from $\mathcal{D}$.

If $d_x(v_i, w_i, H(r)) \leq 1/(6L)$, then we add $H(r)$ to $\mathcal{D}$. Also, we set the level of every edge $e \in E(H(r))$ to $\max\{p, q\}$ where $p = 0$ if $E(H) \cap D_1 = \emptyset$ and $p = \max_{e \in E(H) \cap D_1} \ell_{i-1}(e)$ otherwise, and $q = 0$ if $E(H) \cap D_2 = \emptyset$ and $q = 1 + \max_{e \in E(H) \cap D_2} \ell_{i-1}(e)$ otherwise. All subgraphs from $\mathcal{D}$ contained in $H(r)$ are removed from $\mathcal{D}$. The levels of the edges outside $E(H(r))$ are left as they are. Then we proceed with the next iteration.

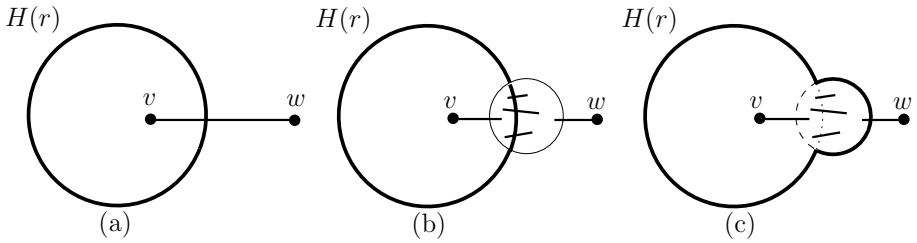

Fig. 5 (a) The extended ball $H(r)$ with a super edge vw as its entry edge (the bold closed curve depicts the boundary of $H(r)$). (b) $H(r)$ after a plain undo of the super edge vw: the number of entry edges of $H(r)$ has increased. (c) $H(r)$ after the undo of the super edge vw and the local modification of $H(r)$: the number of entry edges of $H(r)$ has not changed.

In all the cases listed above, before we start the next iteration, we undo the contraction of all remaining super edges in $\mathcal{D}$. A special care is given to super edges from $\delta(r)$. If a super edge is cut by $\delta_2(r)$, we deal with the remaining part of it that is outside of $H(r)$ in the same way as we treat $H(r)$ for $|Z| = 1$, that is, we remove it from G_i and add its edges to C. If a super edge is an entry edge of $H(r)$ (Fig. 5(a)), then a simple replacement of it by the corresponding subgraph from $\mathcal{D}$ may increase the connectivity between t_i and s and thus destroy the h-route cut (Fig. 5(b)). To deal with this issue, we extend $H(r)$ to

contain the subgraph represented by that super edge which is an entry edge of $H(r)$ (Fig. 5(c)). This way, the number of entry edges of $H(r)$ does not exceed two.

Before stating the main result of this section, we state and prove three auxiliary lemmas.

Lemma 6 *For each $e \in E$ and each iteration i, $\ell_i(e) \leq L$.*

Proof We will show by induction on j that $\ell_j(e) \leq \log(|T_j| + 1)$ for every $e \in E(H)$, where H_j is the extended ball associated with t_j and $T_j = \{t_1, \ldots, t_j\} \cap V(H_j)$ is the set of terminals belonging to $V(H_j)$; as $L = \lfloor \log(k+1) \rfloor$ and $|T_j| \leq k$ for every j, this will prove the lemma.

For $j = 0$ the claim is obvious: at the end of the first iteration, the level of every edge is at most one.

Consider now some iteration $j \geq 1$. Note that t_j is never an entry node of H_j, and since the entry nodes of some H_j may only be nodes not belonging to any $H_{j'}$ or entry nodes of other $H_{j'}$s, none of the terminals in T_j can be an entry node of H_j.

The only case when a level of an edge increases is the case that $|Z| = 2$ with $d_x(v_j, w_j, H_j) \leq 1/(6L)$. Assume that $q > p$ as otherwise there is no level increase. Note that in this case there are at least two edges $e_1, e_2 \in E(H_j)$ of level $q - 1 > 0$. By property (e), these must be super edges representing subgraphs H_{j_1} and H_{j_2} that were constructed when dealing with the terminals t_{j_1} and t_{j_2}. By property (b), these subgraphs are edge-disjoint, and since none of the entry nodes can be a terminal in T_j, the terminal sets T_{j_1} and T_{j_2} must be disjoint. On the other hand, $T_{j_1} \cup T_{j_2} \subsetneq T_i$. By our induction hypothesis, it must hold that $\ell_{i-1}(e_1) \leq \log(|T_{j_1}| + 1)$ and $\ell_{i-1}(e_2) \leq \log(|T_{j_2}| + 1)$. Given that the levels of the edges e_1 and e_2 are equal, we can upper bound $\ell_{j-1}(e_1)$ and $\ell_{j-1}(e_2)$ by $\log((|T_j| - 1)/2 + 1)$. Hence,

$$\ell_j(e) \leq \log((|T_j| + 1)/2) + 1 = \log(|T_j| + 1)$$

which completes the induction. □

Lemma 7 *For each $e \in D$, $x(e) \leq 1/(6L)$.*

Proof Follows from the description of the algorithm: every edge e with strictly positive level in G'_i is a super edge, and by property (c) and the definition of $x(vw)$ of a super edge vw, $x(e) \leq 1/(6L)$. □

Lemma 8 *If the properties (a)-(e) are satisfied at the beginning of iteration i, then they are satisfied at the beginning of the next iteration as well.*

Proof Properties (a)-(c) and (e) follow directly from the algorithm, and property (d) follows from the combination of Lemmas 4, 6 and 7 and our rule of increasing the level of the edges. □

Theorem 1 *The approximation ratio of the algorithm for the 3-route single-source cut problem is $O(\log^2 k)$.*

Proof The correctness of the algorithm follows from the property (a) and Lemma 5. Concerning the approximation ratio, from the property (d) and from the upper bound on $\ell_k(e) \leq L$ from Lemma 6 we know that the cost of the multiroute multicut constructed by the algorithm is $O(\log^2 k) \sum_{e \in E} x(e)c(e)$. As the sum in the previous expression is a lower bound on the cost of any solution, the proof of the theorem is completed. □

4 Multiple Sources

The algorithm for multiple sources is an extension of the single-source algorithm for $h = 3$. Again, it works in iterations. Roughly, in iteration i the algorithm constructs a 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 $x(e) \leq 1/(6hL)$ for every $e \in D$.

4.1 Number of overlapping recursive calls.

To guarantee that the depth of the recursion is small, we ensure that every constructed ball contains at most half of the remaining commodities (i.e., both terminals of the commodity are inside the ball). 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 [5]. Lemma 9 deals with this problem. The difference here is that we cannot treat the balls independently. As in the single source case, it can happen that edges might be part of more than one ball of the same recursion level. Hence, a naive use of the recursion may cause too many recursive calls on the same area even though the depth of the recursion is small.

To guarantee that there are not too many recursive calls on the same area, we apply a lazy strategy: instead of invoking the recursive call immediately after the ball B is defined, the algorithm postpones the calls till all balls in a recursion level have been processed. The reasoning behind this is that if the algorithm later defines another ball B' at the same recursion level that contains the super edge representing the ball B (as inner edge or entry edge), it is sufficient to perform only the recursive call for B'; this call will then also take care of all commodities inside B. In this way recursive calls at a certain recursion level are only made for disjoint parts of the original graph. When executing a recursive call on a ball B, the algorithm always makes sure that the ball B is expanded to the original edges of the graph, that is, it does not contain any super edges apart from special super edges of level -1, which

we will define later. We will make sure that a recursive call has to deal with at most $\log k$ super edges of level -1, and each such super edge e satisfies $x(e) \leq 1/(6hL)$.

Before we state and prove Lemma 9 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 $d_{x+x_i}(s_i) = 1$ (i.e., the distance of s_i from t_i is 1). Then, for $r \in (0,1)$ and $z \in \{s_i, t_i\}$ we define $B^z(r) = \{u \in V \mid d_{x+x_i}(z,u) \leq r\}$, $\delta^{t_i}(r) = \delta(r)$, $\delta^{s_i}(r) = \delta(1-r)$, $\delta_x^{t_i}(r) = \delta_x(r)$, and $\delta_{x_i}^{s_i}(r) = \delta_{x_i}(1-r)$ 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 $\delta_1^z(r) = \delta^z(r) \setminus \{\text{argmax}_{e \in \delta^z(r)} c(e)\}$, $\delta_2^z(r) = \delta_1^z(r) \setminus \{\text{argmax}_{e \in \delta_1^z(r)} c(e)\}$ and

$$\begin{aligned} D_1^z = \{uv \in D \mid \ell(uv) = -1 \text{ or } d(z,uv) \text{ is minimal among all } e \in D \\ \text{with } \ell(e) = \ell(uv) \text{ (where ties are broken arbitrarily)}\} , \\ D_2^z = D \setminus D_1^z . \end{aligned}$$

where $d(t_i, uv) = d(uv)$ and $d(s_i, uv) = 1 - d(t_i, uv)$, and D is the set of super edges. Now we can adjust the notion of a good radius: in this section we always have to specify the vertex we talk about for the good radius. For $z \in \{s_i, t_i\}$, we say that

$$\mathcal{R}^z = \{r \in [0,1] \mid |\delta_{x_i}^z(r)| \leq h-1 \text{ and } \delta_x^z(r) \cap D_1^z = \emptyset\} .$$

is the set of good radii for z, where each edge is viewed as a segment consisting of an x-part followed by an x_i-part on the way from t_i for both choices of z. Finally, for $z \in \{s_i, t_i\}$ and $r \in [0,1]$, we define

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

Lemma 9 *There exist a good radius r_s for s and a good radius r_t for t such that*

$$\begin{aligned} r_s + r_t \leq 1 , \text{ and} \\ c(\delta_{h-1}(s_i, r_s)) \leq 6h \log(2k) \cdot V(s_i, r_s) , \text{ and} \\ c(\delta_{h-1}(t_i, r_t)) \leq 6h \log(2k) \cdot V(t_i, r_t) . \end{aligned}$$

Proof For $z \in \{s_i, t_i\}$ let $\mathcal{S}^z = \{r \in [0,1] \mid |\delta_{x_i}^z(r)| \leq h-1\}$. From Lemma 3 it follows that $|\mathcal{S}^z| \geq 1/h$ for both choices of z. Since $\delta_{x_i}^{s_i}(r) = \delta_{x_i}^{t_i}(1-r)$, we obtain that $|(\mathcal{S}^{s_i} \cap [0,r]) \cup (\mathcal{S}^{t_i} \cap [0,1-r])| \geq 1/h$ for every $r \in [0,1]$. Moreover, as the number of edges in D_1^z is at most $2L$ and $x(e) \leq 1/(6hL)$ for every $e \in D_1^z$, $|\mathcal{S}^z \setminus \mathcal{R}^z| \leq 1/(3h)$, for both choices of z. Therefore, for any radius $r \in [0,1]$,

$$\begin{aligned} |\mathcal{R}^{s_i} \cap [0,r]| + |\mathcal{R}^{t_i} \cap [0,1-r]| &\geq \max\{|\mathcal{S}^{s_i} \cap [0,r]| - 1/(3h), 0\} + \\ &\quad \max\{|\mathcal{S}^{t_i} \cap [0,1-r]| - 1/(3h), 0\} \\ &\geq 1/h - 2/(3h) = 1/(3h) . \end{aligned}$$

Since $|(\mathcal{R}^{s_i} \cap [0,r]|$ when seen as a function of r, is continuous and non-decreasing on the interval from 0 to at least $1/h - 1/(3h)$, there must be an r with $|\mathcal{R}^{s_i} \cap [0,r]| \geq 1/(6h)$ and $|\mathcal{R}^{t_i} \cap [0,1-r]| \geq 1/(6h)$. Thus, it follows from Lemma 1 that there is an $r_s \in \mathcal{R}^{s_i} \cap [0,r]$ with $c(\delta^{s_i}_{h-1}(r_s)) \leq 6h \log(2k) \cdot V(s_i, r_s)$ and an $r_t \in \mathcal{R}^{t_i} \cap [0,1-r]$ with $c(\delta^{t_i}_{h-1}(r_t)) \leq 6h \log(2k) \cdot V(t_i, r_t)$. Since $r_s + r_t \leq 1$, the lemma follows. □

If r_s and r_t are the radii from Lemma 9 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 1 *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 super 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 with their level values 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 so-called *forbidden* edges. To simplify the description, we will view forbidden edges as original, restricted edges (even though they are actually super 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 the (extended) ball H^z built around the terminal $z \in \{s_i, t_i\}$ with radius r. We distinguish between two cases (as in the previous section, v_i and w_i denote the two entry nodes of H^z):

- If $d_x(v_i, w_i, G \setminus H^z) \leq 1/(12hL^2)$, the recursive call is invoked for the subgraph H^z with an extra edge v_iw_i with $x(v_iw_i) = d_x(v_i, w_i, G \setminus H^z)$, $c(v_iw_i) = c(\text{mincut}(v_i, w_i, G \setminus H^z))$ and level -1. For each $j \neq i$ we also set $x_j(v_iw_i) = d_{x+x_j}(v_i, w_i, G \setminus H^z) - x(v_iw_i)$. The set of commodities consists of those with both terminals in $B^z(r)$. The set of restricted edges is preset to $\{uv \in D \cap E(H^z) \mid \ell(uv) = -1\} \cup \{v_iw_i\}$; the level of every other edge is set to zero at the beginning of the recursive call. Recall that we do not allow the restricted edges of level -1 to be cut or to be charged for a cut, be it an h-route cut or a mincut. This is the reason why we use a different

Algorithm 3-RouteCutMultipleSources($G=(V,E), P=\{(s_1,t_1),\ldots,(s_k,t_k)\}, D_{-1}$)
// D_{-1} only contains restricted edges of level -1
$F:=\emptyset$, $\mathcal{D}=\emptyset$, $\ell(e):=0$ for every $e\in E\setminus D_{-1}$
$\mathcal{I}:=\emptyset$ // set of candidates for recursive calls
while there is h-connected $(s_i,t_i)\in P$ **do**
  Contract all $H\in\mathcal{D}$ into a set of super edges D
  $D:=D\cup D_{-1}$
  (r,z) :=findradius(G,D,s_i,t_i) // $z\in\{s_i,t_i\}$
  $F:=F\cup\delta^z_2(r)$
  $\mathcal{I}':=\{(\bar{H},\bar{P})\in\mathcal{I}\mid \bar{H}\subseteq H^z\}$
  $P':=\{(s,t)\in P\mid s,t\in B^z(r)\}\cup\bigcup_{(\bar{H},\bar{P})\in\mathcal{I}'}\bar{P}$
  $\mathcal{I}:=(\mathcal{I}\setminus\mathcal{I}')\cup\{(H^z,P')\}$ // add a new instance for recursive call
  $P:=P\setminus\{(s,t)\in P\mid s\in B^z(r)$ or $t\in B^z(r)\}$
  $V:=V\setminus B^z(r)$
  $E:=E\setminus\{uv\in E\mid u\in B^z(r)$ or $v\in B^z(r)\}$
  $D:=D\cap E$
  Let H^z be the extended ball $H^z(r)$ and v_i,w_i be its entry nodes
  if $v_i\neq w_i$ **then**
    if $d_x(v_i,w_i,H^z)\geq 1/(6hL)$ **then**
      $F:=F\cup$ mincut2(v_i,w_i,H^z)
    else
      $E:=E\cup\{v_iw_i\}$
      $D:=D\cup\{v_iw_i\}$
      $\mathcal{D}:=\mathcal{D}\cup\{H^z\}$
      $x(v_iw_i):=d_x(v_i,w_i,H^z)$
      $x_j(v_iw_i):=d_{x+x_j}(v_i,w_i,H^z)-x(v_iw_i),\forall j\neq i$
      $c(v_i,w_i):=c($mincut2$(v_i,w_i,H^z))$
      $\ell(v_i,w_i):=\max\{1+\max_{e\in E(H)\cap D_2}\ell(e),\ \max_{e\in E(H)\cap D_1}\ell(e)\}$
  else
    $C:=C\cup E(H)$
    $E:=E\setminus E(H)$
  Replace the super edges in F by their mincuts, remove them from D,
  and add the edges of their subgraphs that are outside of $H(r)$ to C
  For any entry edge of $H(r)$ that is a super edge e, add the subgraph corresponding
  to e to $H(r)$ and choose its outside entry edge as new entry edge for $H(r)$
  Undo the contraction of all other super edges in D
for each $(\bar{H},\bar{P})\in\mathcal{I}$ **do**
  if $d_x(v_i,w_i,G\setminus\bar{H})\leq 1/(12hL^2)$ **then**
    $x(v_iw_i):=d_x(v_i,w_i,G\setminus\bar{H})$
    $x_j(v_iw_i):=d_{x+x_j}(v_i,w_i,G\setminus\bar{H})-x(v_iw_i),\forall j\neq i$
    $c(v_iw_i):=c($mincut$(v_i,w_i,G\setminus\bar{H}))$
    $\ell(v_iw_i):=-1$
    $E(\bar{H}):=E(\bar{H})\cup\{v_iw_i\}$
    $D':=D_{-1}\cup\{v_iw_i\}$
  else
    $\mathcal{T}:=\mathcal{T}\cup\{v_iw_i\}$
    $D':=D_{-1}$
  $F:=F\cup$ 3-RouteCutMultipleSources$(\bar{H},\bar{P},D')$
// at the highest recursion level, we add a 2-route cut for $\mathcal{T}$ to F
return F

Fig. 6 The 3-route cut algorithm for a multiple sources. Initially, $C:=\emptyset$ and $\mathcal{T}:=\emptyset$. The algorithm is called with parameters $G=(V,E)$, $P=\{(s_1,t_1),\ldots,(s_k,t_k)\}$, and $D_{-1}=\emptyset$. Procedure findradius(G,D,s_i,t_i) selects a radius and one of the terminals as specified in Section 4.1. Procedure mincut2(v_i,w_i,H) works as specified in Lemma 10.

mincut procedure, called mincut2, which computes a smallest cut in H^z *avoiding* the edges of level -1 — see Lemma 10 and Fig. 6.
- If $d_x(v_i, w_i, G \setminus H^z) > 1/(12hL^2)$, we also use the expanded subgraph H^z for the recursive call but without the edge v_iw_i. Instead, the pair $\{v_i, w_i\}$ is added to a global set $\mathcal{T}$. The set of commodities consists again of those with both terminals in $B^z(r)$, and the set of restricted edges is preset to $\{uv \in D \cap E(H^z) \mid \ell(uv) = -1\}$. Once all recursive calls have been completed at all levels of recursion, we compute a 2-route cut for all pairs in $\mathcal{T}$ in order to decrease their connectivity to one. We explain in Section 4.3 why and how to do that.

It remains to discuss the details of the first case. Let $\bar{V}(H^z)$ be the volume of H^z without the forbidden edges. It holds:

Lemma 10 *If $d_x(v_i, w_i, H^z) \geq 1/(6hL)$ then $c(mincut2(v_i, w_i, H^z)) \leq 12hL \cdot \bar{V}(H^z)$.*

Proof In the following, consider all x_i-values in H^z to be 0. A radius $\rho \in [0, d_x(v_i, w_i, H^z)]$ is *forbidden* if $\delta(v_i, \rho)$ contains a forbidden edge. Let R be the set of forbidden radii and $\gamma = \min_{\rho \in [0, d_x(v_i, w_i, H^z)] \setminus R} c(\delta(v_i, \rho))$. As each forbidden edge has an x-length of at most $1/(12hL^2)$ and there are at most $\log k$ forbidden edges, $|R| \leq 1/(12hL)$. Also, $\gamma \geq c(\text{mincut2}(v_i, w_i, H^z))$. Hence, similarly as in the proof of Lemma 2,

$$\begin{aligned}\bar{V}(H^z) &\geq \int_{\rho \in [0, d_x(v_i, w_i, H^z)] \setminus R} c(\delta(v_i, \rho)) \;\geq\; \gamma \cdot (d_x(v_i, w_i, H^z) - |R|) \\ &\geq c(\text{mincut2}(v_i, w_i, H^z))/(12hL)\end{aligned}$$

We conclude that $c(\text{mincut2}(v_i, w_i, H^z)) \leq 12hL \cdot \bar{V}(H^z)$. □

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 that the v_i and w_i were possibly connected *outside* of 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 find a 2-route cut for the pairs in $\mathcal{T}$.

We proceed as follows. Consider any pair $\{v_i, w_i\}$ in $\mathcal{T}$. We know that $d_x(v_i, w_i, G \setminus H^z) > 1/(12hL^2)$. Hence, when scaling all x-lengths of the edges in G by a factor of $12hL^2$, we obtain a feasible fractional solution of the old LP

relaxation of the minimum multicommodity 2-route cut problem that we recall as LP (8) in Section 5. Therefore, it follows from the $O(\log^2 k)$-approximation algorithm of Barman and Chawla [5] and Theorem 3 below that there is a 2-route cut F' for $\mathcal{T}$ in G with $c(F') = O(\log^4 k \cdot \phi)$. We add F' to the set of edges F to obtain the final cut returned by the algorithm. Note that a 2-route cut for the pairs in $\mathcal{T}$ indeed suffices for the final F to be a 3-route cut.

Lemma 11 *For every source-sink pair (s_j, t_j) it holds that after removing F and F' from G, the nodes s_j and t_j are at most 2-connected.*

Proof Suppose that the source-sink pair (s_j, t_j) was recursively processed in some ball H^z with entry nodes v_i, w_i. If v_i and w_i are still connected after the removal of F and F', then they are either connected in H^z or in $G \setminus H^z$ but not in both: otherwise they would still be 2-connected. In the rest of the proof we distinguish these two cases.

If v_i and w_i are connected in H^z but not in $G \setminus H^z$, then the final connectivity of s_j and t_j is at most 2 as we know that they are at most 2-connected in H^z (after removing the edges in F).

If v_i and w_i are connected in $G \setminus H^z$ but not in H^z, and a path between v_i and w_i in $G \setminus H^z$ can be extended into a path between s_j and t_j in G without F and F' (otherwise the claim of the lemma is obvious), then there is no path between s_j and t_j in H^z as otherwise v_i and w_i would also be connected inside H^z, contradicting out assumption. Hence, s_j and t_j are at best be 1-connected in this case. □

Considering the explanation at the beginning of this section and the bound from the previous paragraph, the cost of the set F is $O(\log^4 k \cdot \phi)$, and at this point, F is a valid 3-route cut. The main theorem follows. A detailed description of the algorithm is given in Figure 4.1.

Theorem 2 *The approximation ratio of the algorithm for the general 3-route cut problem is $O(\log^4 k)$.*

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.
Note that without the factor h in the objective function the linear program (8) is another relaxation of the h-route cut problem (the approximation algorithm of Chekuri and Khanna [7] for 2-route cuts is based on this relaxation). We will refer by (8') to the linear program (8) with the objective function scaled down to $\sum_{e \in E} c(e) \cdot x(e)$.

$$\max \sum_{q\in\mathcal{Q}} f(q) \qquad (7)$$
$$\sum_{q\in\mathcal{Q}:e\in q} f(q) \le h\cdot c(e) \quad \forall e\in E$$
$$f(q)\ge 0 \quad \forall q\in\mathcal{Q}$$

$$\min\ h\cdot\sum_{e\in E} c(e)\cdot x(e) \qquad (8)$$
$$\sum_{e\in q} x(e)\ge 1 \quad \forall q\in\mathcal{Q}$$
$$x(e)\ge 0 \quad \forall e\in E$$

There are simple examples showing that the linear relaxation (8') 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 (8') 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; note that no previous results from this paper are needed in the proof of this result in Theorem 3. As a corollary we obtain an approximate duality theorem for multiroute cuts and flows.

Theorem 3 *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 (8'). Then $O_2 \le O_1 \le h\cdot O_2$, 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 (8'). 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., $\sum_{e\in E} c(e)\bar{x}(e) \le h\sum_{e\in E} c(e)x(e)$).

It is tempting to argue as follows. For every commodity i, the maximum number of edge disjoint paths between s_i and t_i of x-length strictly less then $1/h$, is $h-1$. Thus, by Menger's theorem, there exists a set F_i of $h-1$ edges whose removal destroys all $s_i - t_i$ paths of x-length less then $1/h$. For each edge $e\in F_i$ we set $x_i(e)=1$, and for each $e\in E$ we set $\bar{x}(e) = h\cdot x(e)$. Unfortunately, Menger's theorem does not hold for paths of bounded length (cf. [4]) and thus, this reasoning is completely wrong.

We have to argue more carefully. 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.

$$\sum_{e\in p} x_i(e) \ge 1 - \sum_{e\in p}\bar{x}(e) \quad \forall p\in\mathcal{P}_i \qquad (9)$$
$$\sum_{e\in E} x_i(e) \le h-1 \qquad \forall e\in E$$
$$x_i(e)\ge 0 \qquad \forall e\in E$$

Assume, for a contradiction, that the linear program (9) 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

$$\sum_{p \in \mathcal{P}_i : e \in p} \lambda(p) \leq 1 \quad \forall e \in E \tag{10}$$
$$\sum_{p \in \mathcal{P}_i} \lambda(p)(1 - \sum_{e \in p} \bar{x}(e)) > h - 1$$

(without loss of generality, we assume that $\gamma = 1$ and therefore it does not show up in the above inequalities; note that every vector (λ, γ) obtained by the application of the Farkas' lemma to the linear program (9) 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 (10) 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 Menger's 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 (8'), from the definition of $\bar{x}$ we know that $\sum_{e \in q} \bar{x}(e) \geq h$. 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 (10) and its size is smaller than the size of λ. Thus, the linear program (9) has a feasible solution, for each i, and the proof is completed. □

Corollary 2 (Duality of multiroute multicommodity flows and cuts) *For any instance with k commodities, the cost of the minimum h-route cut for $h \leq 3$ is at least a fraction $1/h$ of the maximum h-route multicommodity flow, and is always at most $O(\log^4 k)$ 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 3, the duality of the linear programs (7) and (8), and Theorem 2 (the approximation algorithm). □

5.1 Sparsest multiroute cut

The *sparsest multiroute cut problem* is a multiroute analog of the sparsest cut problem. As input we are given a graph, k pairs of vertices s_i, t_i and k integers d_i called *demands*. The task is to find a subset of edges F such that the ratio of the sum of capacities of edges in F and the sum of demands of commodities that are at most $h-1$-connected after the removal of F, is minimized. The approximation algorithm is based on a linear programming relaxation of the sparsest cut problem; we refer to the survey paper by Shmoys [17] for more details about the derivation of the linear relaxation. As in the previous section, $\mathcal{Q}_i$ denotes the set of all elementary h-flows between s_i and t_i.

$$\begin{aligned} \min \sum_{e \in E} c(e)x(e) & \qquad (11) \\ \sum_{i=1}^{k} d_i y_i &= 1 \\ \sum_{e \in q} x(e) &\geq y_i \qquad \forall i \in [k], q \in \mathcal{Q}_i \\ x(e) &\geq 0 \qquad \forall e \in E \\ y_i &\geq 0 \qquad \forall i \in [k] \end{aligned}$$

Sparsest multiroute cut. Here we describe a rounding procedure for the LP (11). We solve the linear program and then, by standard techniques [11,17], find a nonempty subset $S \subseteq [k]$ of commodities such that $y_{\min} \geq 1/(\sum_{i \in S} d_i H(D))$ where $y_{\min} = \min_{i \in S} y_i$, $D = \sum_{i=1}^{k} d_i$ and $H(j)$ is the j-th Harmonic number. We scale the fractional solution x by $1/y_{\min}$ to obtain a feasible fractional solution $\bar{x} = x/y_{\min}$ for the linear program (8') with the set of commodities given by the set S. Then we apply the construction from the proof of Theorem 3 to obtain from $\bar{x}$ a feasible fractional solution of the linear program (1); the cost of the scaled fractional solution is larger than the fractional optimum of the original relaxation of the multiroute sparsest cut by a factor of $h/y_{\min} \leq h \sum_{i \in S} d_i H(D)$ at most. To round this fractional solution to an integral one we use the algorithm described in Section 3. Clearly, the integral solution for the minimum h-route cut instance S corresponds also to an integral solution of the original sparsest h-route cut instance.

Realizing that the objective of the sparsest multiroute cut problem is to minimize the ratio between the capacity of the removed edges and the sum of demands of disconnected commodities (which is $\sum_{i \in S} d_i$), the main results of this section follow.

Theorem 4 *The approximation ratio achievable in polynomial time for the multiroute sparsest cut problem with $h \leq 3$ is $O(\log^4 k \log D)$ where $D = \sum_{i=1}^{k} d_i$.*

Corollary 3 *For any instance with k commodities, the sparsest h-route cut for $h \leq 3$ is at least as large as the maximum concurrent h-route multicommodity flow, and is always at most $O(\log^4 k \log D)$ times larger.*

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

References

1. Charu C. Aggarwal and James B. Orlin. On multiroute maximum flows in networks. *Networks*, 39:43–52, 2002.
2. Yonatan Aumann and Yuval Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.
3. 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.
4. Georg Baier, Thomas Erlebach, Alexander Hall, Ekkehard Köhler, Petr Kolman, Ondrej Pangrác, Heiko Schilling, and Martin Skutella. Length-bounded cuts and flows. *ACM Transactions on Algorithms*, 7(1):4–27, 2010.
5. 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.
6. 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 Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007.
7. Chandra Chekuri and Sanjeev Khanna. Algorithms for 2-route cut problems. In *Proceedings of the 35th International Colloquium on Automata (ICALP)*, volume 5125 of *Lecture Notes in Computer Science*, pages 472–484, 2008.
8. Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994. Preliminary version in Proceedings of the 24th ACM Symposium on Theory of Computing (STOC), 1992.
9. L. R. Ford and D. R. Fulkerson. Maximum flow through a network. *Canad. J. Math.*, 8:399–404, 1956.
10. 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.
11. Nabil Kahale. On reducing the cut ratio to the multicut problem. Technical report, 1993. DIMACS Technical report 93-78.
12. Wataru Kishimoto and M. Takeuchi. On m-route flows in a network. *IEICE Transactions*, J-76-A(8):1185–1200, 1993. (in Japanese).
13. Petr Kolman and Christian Scheideler. Towards duality of multicommodity multiroute cuts and flows: Multilevel ball-growing. In *Proceedings of the 28th International Symposium on Theoretical Aspects of Computer Science (STACS)*, Leibniz International Proceedings in Informatics (LIPIcs), 2011.
14. Petr Kolman and Christian Scheideler. Approximate duality of multicommodity multiroute flows and cuts: single source case. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 800–810, 2012.
15. 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.
16. N. Linial, E. London, and Yu. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.

17. 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.
18. Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2001.
19. D.P. Williamson and D.B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.