Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 377 (2007) 35–42
Theoretical Computer Science
www.elsevier.com/locate/tcs

The multi-multiway cut problem[☆]

Adi Avidor[a,*], Michael Langberg[b,1]

[a] School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
[b] Computer Science Division, The Open University of Israel, Raanana 43107, Israel

Received 5 March 2006; received in revised form 8 December 2006; accepted 4 February 2007

Communicated by A. Fiat

Abstract

In this paper, we define and study a natural generalization of the multicut and multiway cut problems: the *minimum multi-multiway cut* problem. The input to the problem is a weighted undirected graph $G = (V, E)$ and k sets $S_1, S_2, \ldots, S_k$ of vertices. The goal is to find a subset of edges of minimum total weight whose removal completely disconnects each one of the sets $S_1, S_2, \ldots, S_k$, i.e., disconnects every pair of vertices u and v such that $u, v \in S_i$, for some i. This problem generalizes both the multicut problem, when $|S_i| = 2$, for $1 \leq i \leq k$, and the multiway cut problem, when $k = 1$.

We present an approximation algorithm for the multi-multiway cut problem with an approximation ratio which matches that obtained by Garg, Vazirani, and Yannakakis on the standard multicut problem. Namely, our algorithm has an $O(\log k)$ approximation ratio. Moreover, we consider instances of the minimum multi-multiway cut problem which are known to have an optimal solution of *light* weight. We show that our algorithm has an approximation ratio substantially better than $O(\log k)$ when restricted to such "light" instances. Specifically, we obtain an $O(\log LP)$-approximation algorithm for the problem when all edge weights are at least 1 (here LP denotes the value of a natural linear programming relaxation of the problem). The latter improves the $O(\log LP \log\log LP)$ approximation ratio for the minimum multicut problem (implied by the work of Seymour and Even et al.).
© 2007 Elsevier B.V. All rights reserved.

Keywords: Approximation algorithms; Linear programming; Minimum multicut; Minimum multiway cut

1. Introduction

The input to the *minimum multicut* problem is an undirected graph $G = (V, E)$ with a weight (or cost) function $w : E \to R^+$ defined on its edges, and a collection $(s_1, t_1), \ldots, (s_k, t_k)$ of vertex pairs. The objective is to find a subset of edges of minimum total weight whose removal disconnects s_i from t_i, for every $1 \leq i \leq k$. The problem is known to be APX-hard [8]. An $O(\log k)$-approximation algorithm for the problem was obtained by Garg, Vazirani and Yannakakis [13].

[☆] A preliminary version of this work appeared in the proceedings of the 9th Scandinavian Workshop on Algorithm Theory (Lecture Notes in Computer Science, vol. 3111) 2004, pp. 273–284.

[*] Corresponding author. Tel.: +972 54 5320259; fax: +972 3 6409357.
E-mail addresses: adi@tau.ac.il (A. Avidor), mikel@openu.ac.il (M. Langberg).

[1] Part of this work was done while visiting the School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel.

0304-3975/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2007.02.026

The *minimum multiway cut* problem is a subproblem of the minimum multicut problem. The input consists of a weighted undirected graph $G = (V, E)$, as in the multicut problem, and a set $\{t_1, t_2, \ldots, t_k\}$ of vertices. The goal is to find a subset of edges of minimum total weight whose removal disconnects t_i from t_j, for every $1 \le i < j \le k$. The problem is also known to be APX-hard [8]. A $(\frac{3}{2} - \frac{1}{k})$-approximation algorithm for the problem was obtained by Calinescu, Karloff and Rabani [4]. An improved $(1.3438 - \varepsilon_k)$-approximation algorithm for the problem was obtained by Karger et al. [15]. In particular, for $k = 3$ the algorithm of Karger et al. [15] achieves an approximation ratio of $12/11$, which matches the integrality gap of the linear programming relaxation of [4]. This result was also obtained independently by Cunningham and Tang [6].

In this work, we define and study a natural generalization of both the multicut and multiway cut problems: the *minimum multi-multiway cut* problem. The input of the minimum multi-multiway cut problem consists of an undirected graph $G = (V, E)$ with a weight function $w : E \to R^+$ defined on its edges, and k *sets* of vertices $S_1, S_2, \ldots, S_k$ (also referred to as groups). The goal is to find a subset of edges of minimum total weight whose removal disconnects, for every $1 \le i \le k$, every two vertices $u, v \in S_i$. When $|S_i| = 2$, for all $1 \le i \le k$, the minimum multi-multiway cut problem is exactly the minimum multicut problem, and when $k = 1$, the minimum multi-multiway cut problem is the minimum multiway cut problem. On the other hand, note that any instance of minimum multi-multiway cut can be reduced to a minimum multicut instance, by simply listing all the pairs that have to be disconnected. Using the Garg, Vazirani and Yannakakis algorithm, this immediately implies an $O(\log n)$ approximation algorithm for minimum multi-multiway cut. We improve upon this ratio in this work.

The minimum multicut problem. The minimum multicut problem (and its relation to multicommodity flow) have been extensively studied during the last few decades. The problem in which $k = 1$ is the standard $s-t$ cut problem, and is known to be solved exactly in polynomial time [10]. The case in which $k = 2$ was also shown to be polynomially solvable by Yannakakis et al. [23] using multiple applications of the max-flow algorithm. For any $k \ge 3$ the problem was proven to be APX-hard by Dahlhaus et al. [8] and thus cannot permit a PTAS, unless P = NP. In a recent paper by Chawla et al. [3] the minimum multicut problem is shown to be NP-hard to approximate within any constant factor, assuming the Unique Games Conjecture of Khot [14]. Chawla et al. also show that a stronger version of the conjecture implies an inapproximability factor of $\Omega(\log\log|V|)$.

The currently best known approximation ratio for the minimum multicut problem is obtained in the work of Garg, Vazirani, and Yannakakis [13]. They present a polynomial algorithm that, given a graph G and a set of k pairs of vertices, finds a multicut of weight at most $O(\log k)$ times the optimal multicut in G. Their algorithm is based on a natural linear programming relaxation of the minimum multicut problem and has the following outline. By solving the relaxation, a *fractional* multicut of the given graph G is obtained. It can be seen that this fractional solution implies a semi-metric on the vertices of G. This semi-metric is now used to round the fractional multicut into an integral one. Namely, the so called *region growing* scheme (introduced by Leighton and Rao [18] and used also by Klein et al. [17]) is applied to define for each pair (s_i, t_i) a region, i.e., a subset of vertices, which are in this case a *ball* of a specific radius centered at s_i. The multicut obtained by the algorithm is now defined as all edges in E with are cut by one of the defined regions.

Several results in the field of approximation algorithms have been inspired by the region growing technique for rounding the solution of linear programs. These include applications of the divide and conquer paradigm (see for example a survey by Shmoys [22]), the design of approximation algorithms for the minimum multicut problem on directed graphs [16,9,5,11] and the results obtained for the minimum correlation clustering problem [7,2].

In this work we study the region growing rounding technique when applied to the multi-multiway cut problem.

Our results. In this paper we present two main results. First, we present an approximation algorithm for the multi-multiway cut problem with an approximation ratio which matches that obtained by [13] on the standard multicut problem. Namely, our algorithm has an $O(\log k)$ approximation ratio. Our algorithm solves a natural linear programming relaxation of the multi-multiway cut problem, and rounds the fractional solution obtained using an *enhanced* region growing technique. Roughly speaking, the region growing technique used in this work differs from that used in previous works as in our case multiple regions are grown in a simultaneous manner rather than one by one.

Secondly, we consider instances to the minimum multi-multiway cut problem which are known to have an optimal solution of *light* weight. Denote such instances as light instances. We show that our algorithm has an approximation ratio substantially better than $O(\log k)$ when restricted to such light instances. Considering the connection between the minimum multi-multiway cut problem and the closely related minimum uncut problem (defined formally below), we show that our result on light instances of minimum multi-multiway cut implies a result of independent interest on the minimum uncut problem. Our results can be summarized as follows.

Theorem 1.1 (*General Multi-Multiway Cuts*). *There exists a polynomial time algorithm which approximates the minimum multi-multiway cut problem within an approximation ratio of* $4\ln(k+1)$.

Theorem 1.2 (*Light Multi-Multiway Cuts*). *Let I be an instance of the minimum multi-multiway cut problem. Let Opt_I be the weight of the optimal multi-multiway cut of instance I. If $w(e) \geq 1$ for all $e \in E$, then one can approximate the minimum multi-multiway cut problem on I within an approximation ratio of* $4\ln(2Opt_I)$.

Corollary 1.3 (*Light Minimum Uncut*). *If an undirected graph $G = (V, E)$ can be made bipartite by the deletion of k edges, then a set of $O(k \log k)$ edges whose deletion makes the graph bipartite can be found in polynomial time.*

A few remarks are in place. Recall that the multi-multiway cut problem is a generalization of both the multicut and multiway cut problems. Hence, our results on the multi-multiway cut problem apply to both these problems as well. Specifically, Theorem 1.2 implies a $4\ln(2Opt_I)$ approximation ratio for the standard minimum multicut problem.

To the best of our knowledge, light instances of the minimum multicut problem have not been addressed directly in the past. However, light instances of the *symmetric* multicut problem on *directed* graphs[2] have been considered. Namely, Seymour [21] proved an existential result which implies (via [9]) an $O(\log(Opt_I)\log\log(Opt_I))$ approximation algorithm for the symmetric multicut problem in the directed case (under the same setting as Theorem 1.2). This in turn implies an $O(\log(Opt_I)\log\log(Opt_I))$ approximation algorithm for the undirected case as well. Hence, in this case our contribution can also be viewed both as a direct proof and an improved result for the "light multicut" problem on undirected graphs.

Our second remark addresses Corollary 1.3 which discusses the familiar minimum uncut problem. Let $G = (V, E)$ be an undirected graph with a nonnegative weight function $w : E \to R^+$ defined on its edges. The minimum uncut problem is the problem of finding a set of edges of minimum weight whose removal disconnects all odd cycles in G, i.e., the resulting graph is bipartite. This problem is also known as a special case of the minimum $2\mathrm{CNF}_{\equiv}$ deletion problem. The problem is known to be APX-hard [20], and has an $O(\sqrt{\log|V|})$ approximation algorithm [1]. Assuming the Unique Games Conjecture the minimum uncut problem is NP-hard to approximate within any constant factor [14]. The parameterized complexity of the minimum uncut problem has also been considered. Namely, in a recent work, Guo et al. [12] show that given an undirected graph that can be made bipartite by deleting k of its edges, one can find in time $O(2^k poly(|V|))$ a subset of edges of size k whose removal yield a bipartite graph. In this case, Corollary 1.3 implies an $O(poly(k, |V|))$-time algorithm which finds $O(k \log k)$ such edges.

Finally, in a recent work, Nagarajan and Ravi [19] consider the *requirement cut* problem, which generalizes the multi-multiway cut problem discussed in this work. The input of the requirement cut problem consists of an undirected graph $G = (V, E)$ with a weight function $w : E \to R^+$ defined on its edges, k sets of vertices $S_1, S_2, \ldots, S_k$ and k *requirements* $r_1, \ldots, r_k$. The goal is to find a subset of edges of minimum total weight whose removal separates each set S_i into at least r_i disconnected components. When $r_i = |S_i|$ the requirement cut problem reduces to the multi-multiway cut problem. Nagarajan and Ravi show that the requirement cut problem can be approximated within an approximation ratio of $O(\log n \log(k \max_i r_i))$ and that under certain complexity assumptions it cannot be approximated beyond a ratio of $\Omega(\log k)$ by a reduction from the set-cover problem. When restricted to the multi-multiway cut problem, the results of [19] do not match those presented in this work. We may also mention that Nagarajan and Ravi use the rounding procedure presented in this work to obtain a better approximation algorithm for the requirement cut problem where the sizes of the sets are not too large. More specifically, [19] shows how to use our approach to obtain a $O((\max_i |S_i|)\log k)$-approximation algorithm for the requirements cut problem.

Organization. The remainder of the paper is organized as follows. In Section 2 we present our algorithm for the minimum multi-multiway cut problem. The proof of Theorems 1.1 and 1.2 appear in Section 2.4. In Section 3 we present the proof of Corollary 1.3.

[2] In the symmetric multicut problem on directed graphs we are given a directed graph with k pairs of vertices (s_i, t_i), and our objective is to find a set of edges of minimum weight which disconnects all cycles containing s_i and t_i for all $i = 1, \ldots, k$.

2. The multi-multiway cut problem

In this section we present our approximation algorithm for the multi-multiway cut problem.

2.1. Multi-multiway cut linear programming relaxation

A multi-multiway cut can be represented by a set of Boolean variables $x(e)$, one for each edge $e \in E$. If $e \in E$ belongs to the multi-multiway cut $x(e) = 1$, otherwise $x(e) = 0$. Our objective is to find a minimum weight multi-multiway cut which disconnects every path connecting pairs of vertices from the same group. We denote by $\mathcal{P}$ the set of all such paths. The multi-multiway cut problem may be posed as the following integer program:

$$\begin{aligned} \min \quad & \sum_{e \in E} w(e)x(e) \\ & \sum_{e \in P} x(e) \geq 1 \quad \forall P \in \mathcal{P} \\ & x(e) \in \{0, 1\} \quad \forall e \in E. \end{aligned}$$

By relaxing the integrality condition, this integer program may be relaxed to obtain the multi-multiway cut linear programming relaxation:

$$\begin{aligned} \min \quad & \sum_{e \in E} w(e)x(e) \\ & \sum_{e \in P} x(e) \geq 1 \quad \forall P \in \mathcal{P} \\ & x(e) \geq 0 \qquad \forall e \in E. \end{aligned}$$

It is not hard to verify that this relaxation can be solved in polynomial time regardless of the fact that it may involve exponentially many constraints. This follows from the observation that the variables $x(e)$ imply a semi-metric on the vertices of the given graph G. Namely, one can define the distance between any two vertices u and v by the length of the shortest path between u and v, where every edge e in E has length $x(e)$. Given this semi-metric, the constraints above are equivalent to the requirement that the distance between every pair of vertices belonging to the same group is at least one. This, in turn implies a natural separation oracle for the multi-multiway cut linear programming relaxation. There is also an equivalent linear program of polynomial size. This relaxation has an integrality gap of $\Omega(\log k)$. Letting *LP* be the value of the linear relaxation, the integrality gap is also $\Omega(\log(LP))$ (even on graphs with edge weights ≥ 1). Both are implied by the integrality gap of the natural minimum multicut linear programming relaxation [13]. This implies that our analysis is tight.

2.2. Definitions and notations

Let $x = \{x(e)\}_{e \in E}$ be an optimal solution to the multi-multiway cut linear programming relaxation. Denote by *LP* the value of the linear program at x, and denote by $w_{\min}$ the minimal weight of an edge $e \in E$. As mentioned above, we define the *length* of an edge $e \in E$ to be $x(e)$, and the *length of a path* to be the sum of the lengths of its edges. The *distance* between a pair of vertices $u, v \in V$, denoted by $dist_x(u, v)$, is now defined to be the length of the shortest path between u and v. Let $Cut(S) = \{(u, v) \in E \mid u \in S,\ v \in V \setminus S\}$, for any set of vertices $S \subseteq V$. Also let $wt(E') = \sum_{e \in E'} w(e)$ for any set of edges $E' \subseteq E$. We denote the set of all distances of vertices from terminals in S_i by $Dist_i = \{dist_x(s, u) \mid s \in S_i,\ u \in V\}$. For $r \in [0, \infty)$ the *ball* of radius r centered at s_{ij} is defined as $Ball_{ij}(r) = \{v \in V \mid dist_x(s_{ij}, v) \leq r\}$, where $1 \leq i \leq k$ and $1 \leq j \leq |S_i|$. Finally, throughout our work, for various functions f let $f(x^-) = \lim_{y \to x^-} f(y)$.

Roughly speaking, we will be interested in two properties of a given set of balls. The first is the number of edges *cut* by these balls. The second is the number of edges inside the given set of balls, referred to as the *volume* of the balls.

More specifically, for each i, we consider the set of balls centered at vertices of S_i where each ball is of *equal* radius $r \in [0, \infty)$. We define (an upper bound on) the weight of the edges 'leaving' these balls as

$$c_i(r) = \sum_{j=1}^{|S_i|} wt(Cut(Ball_{ij}(r))).$$

CUTALG(α, δ)

1. $CUT \leftarrow \phi$
2. Solve the multi-multiway cut linear programming relaxation.
3. While there is a path from some $s' \in S_i$ to $s'' \in S_i$ (where $1 \le i \le k$)
4. Set r_i to be the radius r in $\left(Dist_i \cup \{\frac{1}{2}\}\right) \bigcap (\delta, \frac{1}{2}]$ that minimizes the value of $\frac{c_i(r^-)}{v_i(r^-)}$
5. $F \leftarrow \bigcup_{j=1}^{|S_i|} Cut(Ball_{ij}(r_i^-))$
6. $CUT \leftarrow CUT \cup F$
7. $V \leftarrow V \setminus \bigcup_{j=1}^{|S_i|} Ball_{ij}(r_i^-)$
8. $E \leftarrow E \cap (V \times V)$
9. $\forall l \in \{1, \dots, k\} \quad S_l \leftarrow S_l \cap V$
10. Return CUT

Fig. 1. Approximation algorithm for minimum multi-multiway cut.

The volume of this set of balls is defined to be

$$v_i(r) = \alpha \cdot LP + \sum_{j=1}^{|S_i|}\left(\sum_{\substack{e=(u,v)\in E \\ u,v\in Ball_{ij}(r)}} w(e)x(e) + \sum_{\substack{e=(u,v)\in E \\ u\in Ball_{ij}(r) \\ v\notin Ball_{ij}(r)}} w(e)(r - dist_x(s_{ij}, u))\right),$$

where α is a parameter, which does not depend on r, and will be specified later.

A few remarks are in place. First notice that in $c_i(r)$, an edge may contribute more than once, as $Cut(Ball_{ij}(r)) \cap Cut(Ball_{ij'}(r))$ is not necessarily empty (where $1 \le j \ne j' \le |S_i|$) and thus $c_i(r)$ is an upper bound on the value of the cut. Secondly, in the definition of $v_i(r)$, the summand $\alpha \cdot LP$ should be viewed as the volume of a set of balls all of radius 0. Finally, note that the function v_i in $[0, \infty)$ is not necessarily continuous but is always continuous from the right.

2.3. Algorithm

Our polynomial time approximation algorithm for multi-multiway cut is described in Fig. 1. Roughly speaking, our algorithm follows the algorithmic paradigm presented in [13]: after solving the linear programming relaxation from Section 2.1, our algorithm rounds the fractional solution using a region growing rounding technique. Namely, for every set S_i which includes a pair of connected vertices, we simultaneously *grow* balls of a specific equal radius $r_i < 1/2$ around all vertices in S_i. The edges in the cut produced by these balls are added to the solution, while the vertices (and their adjacent edges) in the balls are removed from the graph. The radius r_i picked is determined by the values $c_i(r)$ and $v_i(r)$ defined previously.

Our algorithm depends on two parameters $\alpha \ge 0$ and $\delta \in [0, 1/2)$. We assume w.l.o.g. that $w(e) > 0$ for all $e \in E$ and that $LP > 0$. Therefore, if there exists a path between two (or more) vertices in S_i, then $v_i(r) > 0$ for all $r \in (\delta, \infty)$. In particular, Step 4 of the algorithm will be well defined. Note that the vertex set V and the groups S_i change after each round of the While Loop in Step 3. This implies changes in the functions c_i and v_i.

In the next subsection we prove the following theorem:

Theorem 2.1. CUTALG(α, δ) *produces a cut of weight at most*

$$\frac{2(1+k\alpha)}{1-2\delta} \ln\left(\frac{(1+\alpha)LP}{\alpha LP + 2\delta w_{\min}}\right) LP.$$

As an immediate result of Theorem 2.1 we obtain Corollaries 2.2 and 2.3 below, which imply Theorems 1.1 and 1.2 stated in the introduction, respectively.

Corollary 2.2. CUTALG($1/k, 0$) *is a* $4\ln(k+1)$*-approximation algorithm for multi-multiway cut.*

Corollary 2.3. *If* $w(e) \ge 1$ *for all* $e \in E$*,* CUTALG($0, 1/4$) *is a* $4\ln(2LP)$*-approximation algorithm for multi-multiway cut.*

2.4. Analysis

In this subsection we prove Theorem 2.1. For the following two lemmas consider the sets S_i and the functions v_i and c_i in any round of the While Loop of Step 3 in CUTALG(α, δ).

Lemma 2.4. *The function v_i is differentiable in $(0, \infty)$ except for a finite number of points. In addition, if v_i is differentiable at r then $c_i(r) = \frac{\mathrm{d}}{\mathrm{d}r} v_i(r)$.*

Proof. By the definition of v_i, if v_i is not differentiable at r, then r must be equal to $dist_x(s_{ij}, u)$ for some $j \in \{1, \ldots, |S_i|\}$ and some vertex $u \in V$. Therefore, there is only a finite number of values in $(0, \infty)$ for which v_i is not differentiable. The second statement stems from the definition of the functions v_i and c_i. □

Lemma 2.5. *For every $\delta \in [0, 1/2)$, if there is a path between vertices in S_i, then there exists $r \in (\delta, 1/2)$ such that $c_i(r) \le \frac{2}{1-2\delta} \ln\left(\frac{(1+\alpha)LP}{v_i(\delta)}\right) v_i(r)$.*

Proof. Assume on the contrary that for every $r \in (\delta, 1/2)$

$$c_i(r) > \frac{2}{1-2\delta} \ln\left(\frac{(1+\alpha)LP}{v_i(\delta)}\right) v_i(r).$$

Recall that $v_i(r) > 0$ for all $r \in (\delta, 1/2)$, as there exists a path between vertices in S_i and we assume that $w(e) > 0$ for all $e \in E$. Therefore,

$$\int_\delta^{1/2} \frac{c_i(r)}{v_i(r)} \mathrm{d}r > (1/2 - \delta) \frac{2}{1-2\delta} \ln\left(\frac{(1+\alpha)LP}{v_i(\delta)}\right) = \ln\left(\frac{(1+\alpha)LP}{v_i(\delta)}\right).$$

By Lemma 2.4, v_i is not differentiable at only a finite number of points, say $s_1 \le \cdots \le s_l$. Set s_0 to be δ and s_{l+1} to be $1/2$. Now, by Lemma 2.4, and by the fact that $v_i(r)$ is monotone increasing and continuous from the right:

$$\begin{aligned}\int_\delta^{1/2} \frac{c_i(r)}{v_i(r)} \mathrm{d}r &= \sum_{j=0}^{l} \int_{s_j}^{s_{j+1}} \frac{\frac{\mathrm{d}}{\mathrm{d}r} v_i(r)}{v_i(r)} \mathrm{d}r \\ &= \sum_{j=0}^{l} \left(\ln v_i(s_{j+1}^-) - \ln v_i(s_j)\right) \\ &\le \ln v_i((1/2)^-) - \ln v_i(\delta) \\ &= \ln \frac{v_i(\frac{1}{2}^-)}{v_i(\delta)}.\end{aligned}$$

As $v_i(\frac{1}{2}^-) \le (1+\alpha)LP$, the latter yields a contradiction. □

Proof (*of Theorem* 2.1). Let $I = \{i_1, \ldots, i_s\} \subseteq \{1, \ldots, k\}$ be the ordered set of group indices for which algorithm CUTALG(α, δ) entered the While Loop (the parameter s denotes the size of I). Recall that in each iteration of the algorithm the graph $G = (V, E)$, the groups S_i, the sets $Ball_{ij}$ and the functions v_i, c_i and $dist_x$ change. In what follows we denote the graph, sets and functions corresponding to the ℓ'th iteration of the While Loop as $G^\ell = (V^\ell, E^\ell)$, S_i^ℓ, $Ball_{ij}^\ell$, v_i^ℓ, c_i^ℓ, and $dist_x^\ell$. (where $1 \le \ell \le s$).

First, observe that the set of edges returned by the algorithm disconnects all terminals within a group, since every path between terminals has length at least 1, while the radius of the balls was chosen to be less than 1/2 (Step 5 of the algorithm).

The weight of the multicut produced by the algorithm is at most $\sum_{\ell=1}^{s} c_{i_\ell}^\ell(r_{i_\ell}^-)$. Here, the definition of r_{i_ℓ} is from Step 4 of our algorithm.

By Lemma 2.5, for each $\ell \in \{1, \ldots, s\}$ and index $i = i_\ell \in I$ there exists $r_i' \in (\delta, 1/2)$ such that $c_i^\ell(r_i') \le \frac{2}{1-2\delta} \ln\left(\frac{(1+\alpha)LP}{v_i^\ell(\delta)}\right) v_i^\ell(r_i')$. By the choice of r_i in Step 4 of CUTALG(α, δ), it is not hard to verify that $c_i^\ell(r_i^-)/v_i^\ell(r_i^-) \le c_i^\ell(r_i')/v_i^\ell(r_i')$. This follows from the fact that the radius r in $(\delta, 1/2]$ that minimizes the value of $c_i^\ell(r^-)/v_i^\ell(r^-)$ is actually in the set $Dist_i \cup \{1/2\}$.

Therefore the weight of the cut produced by the algorithm is at most

$$\sum_{\ell=1}^{s} \frac{2}{1-2\delta} \ln\left(\frac{(1+\alpha)LP}{v_{i_\ell}^{\ell}(\delta)}\right) v_{i_\ell}^{\ell}(r_{i_\ell}^{-}).$$

Consider a certain value $\ell \in \{1, \ldots, s\}$. Let $i = i_\ell$. Since there exists a path between at least two vertices in the group S_i^ℓ,

$$\begin{aligned} v_i^\ell(\delta) &\geq \alpha LP + \sum_{j=1}^{|S_i^\ell|}\left(\sum_{\substack{e=(u,v)\in E^\ell \\ u,v\in Ball_{ij}^\ell(\delta)}} w_{\min} x(e) + \sum_{\substack{e=(u,v)\in E^\ell \\ u\in Ball_{ij}^\ell(\delta) \\ v\notin Ball_{ij}^\ell(\delta)}} w_{\min}(\delta - dist_x^\ell(s_{ij}, u))\right) \\ &\geq \alpha LP + 2\delta w_{\min}. \end{aligned}$$

Hence, the weight of the cut produced by the algorithm is bounded by

$$\frac{2}{1-2\delta} \ln\left(\frac{(1+\alpha)LP}{\alpha LP + 2\delta w_{\min}}\right) \sum_{\ell=1}^{s} v_{i_\ell}^{\ell}(r_{i_\ell}^{-}).$$

Observe that by the definition of $v_{i_\ell}^\ell$ and by Steps 7, 8 and 9 of the algorithm it holds that $\sum_{\ell=1}^{s} v_{i_\ell}^{\ell}(r_{i_\ell}^{-}) \leq k\alpha LP + LP$, yielding the desired bound on the weight of the cut. □

3. "Light" minimum uncut

Corollary 1.3. *If an undirected graph $G = (V, E)$ can be made bipartite by the deletion of k edges, then a set of $O(k \log k)$ edges whose deletion makes the graph bipartite can be found in polynomial time.*

Proof. Let $G = (V, E)$ be an undirected graph of size n which can be made bipartite by the deletion of k edges. Using the reduction presented in [17] one can (efficiently) obtain a graph $G' = (V', E')$ (with unit edge weights) and a set of n pairs of vertices $\{(s_i, t_i)\}_{i=1}^n$ with the following properties: (a) the minimum multicut on input G', $\{(s_i, t_i)\}_{i=1}^n$ is of value at most $2k$, and (b) given any multicut of G' of weight w one can (efficiently) find at most w edges in G whose removal results in a bipartite graph. Now using CUTALG(0, 1/4) on input G', $\{(s_i, t_i)\}_{i=1}^n$ we obtain (Corollary 2.3) a multicut of G' of weight at most $O(k \log k)$, which by the above implies our assertion. □

4. Concluding remarks

In this work we have defined and analyzed the multi-multiway cut problem, which is a generalization of both the multicut and the multiway cut problems. We have presented an approximation algorithm for the minimum multi-multiway cut problem with an approximation ratio that matches the currently best known approximation ratio for the minimum multicut problem. Moreover, we have shown that our algorithm performs significantly better on instances which are known to have a "light weight" multicut.

The question whether there exists an algorithm (for both the minimum multi-multicut and the minimum multicut problems) with an approximation ratio that improves over the presented ratio of $O(\log k)$ remains an intriguing open problem. It is not likely that such an algorithm will use in a direct manner the standard relaxation of the multi-multiway cut problem of Section 2 due to its large integrality gap. In a recent work of Agarwal et al. [1] the integrality gap of a natural semidefinite programming relaxation for minimum multicut is also shown to be $\Omega(\log |V|)$. A similar semidefinite programming relaxation for minimum multi-multiway cut can be formulated. As before, this implies that an improved approximation algorithm that uses this relaxation naively is not probable.

Acknowledgments

The authors would like to thank Uri Zwick for many helpful discussions.

The first author's work was supported in part by THE ISRAEL SCIENCE FOUNDATION founded by The Israel Academy of Sciences and Humanities.

References

[1] A. Agarwal, M. Charikar, K. Makarychev, Y. Makarychev, $O(\sqrt{\log n})$ approximation algorithms for Min UnCut, Min 2CNF deletion, and directed cut problems, in: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, ACM-SIAM, 2005, pp, 573–581.

[2] M. Charikar, V. Guruswami, A. Wirth, Clustering with qualitative information, in: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003, pp. 524–533.

[3] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, D. Sivakumar, On the hardness of approximating multicut and sparsest-cut, in: Proceedings of the 21st IEEE Conference on Computational Complexity, 2006 (in press).

[4] G. Calinescu, H. Karloff, Y. Rabani, An improved approximation algorithm for MULTIWAY CUT, in: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM-SIAM, 1998, pp. 48–52.

[5] J. Cheriyan, H.J. Karloff, Y. Rabani, Approximating directed multicuts, Combinatorica 25 (3) (2005) 251–269.

[6] W.H. Cunningham, L. Tang, Optimal 3-terminal cuts and linear programming, in: Proceedings of the 7th International IPCO Conference on Integer Programming and Combinatorial Optimization, in: Lecture Notes in Computer Science, vol. 1610, 1999, pp. 114–125.

[7] E.D. Demaine, N. Immorlica, Correlation clustering with partial information, in: Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, in: Lecture Notes in Computer Science, vol. 2764, 2003, pp. 1–13.

[8] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour, M. Yannakakis, The complexity of multiterminal cuts, SIAM Journal on Computing 23 (1994) 864–894.

[9] G. Even, J. Naor, B. Schieber, M. Sudan, Approximating minimum feedback sets and multicuts in directed graphs, Algorithmica 20 (2) (1998) 151–174.

[10] L.R. Ford, D.R. Fulkerson, Maximal flow through a network, Canadian Journal of Mathematics 8 (1956) 399–404.

[11] A. Gupta, Improved results for directed multicut, in: Proceedings of 14th Annunal ACM-SIAM Sympposium on Discrete Algorithms, 2003, pp. 454–455.

[12] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, S. Wernicke, Improved fixed-parameter algorithms for two feedback set problems, in: Proceedings of Workshop on Algorithms and Data Structures, in: Lecture Notes in Computer Science, vol. 3608, 2005, pp. 158–168.

[13] N. Garg, V. Vazirani, M. Yannakakis, Approximate max-flow min-(multi)cut theorems and their applications, SIAM Journal on Computing 25 (2) (1996) 235–251.

[14] S. Khot, On the power of unique 2-prover 1-round games, in: Proceedings of 34th Annual ACM Symposium on the Theory of Computing, 2002, pp. 767–775.

[15] D.R. Karger, P.N. Klein, C. Stein, M. Thorup, N.E. Young, Rounding algorithms for a geometric embedding of minimum multiway cut, Mathematics of Operations Research 29 (3) (2004) 436–461.

[16] P.N. Klein, S.A. Plotkin, S. Rao, É. Tardos, New network decompositions theorems with applications, Brown University Technical Report CS-93-30, 1993.

[17] P.N. Klein, S. Rao, A. Agrawal, R. Ravi, An approximate max-flow min-cut relation for undirected multicommodity flow, with applications, Combinatorica 15 (2) (1995) 187–202.

[18] F.T. Leighton, S. Rao, Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms, Journal of the ACM 46 (6) (1999) 787–832.

[19] V. Nagarajan, R. Ravi, Approximation algorithms for requirement cut on graphs, in: Proceedings of the 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, 2005, pp. 209–220.

[20] C.H. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, Journal of Computer and System Sciences 43 (3) (1991) 425–440.

[21] P.D. Seymour, Packing directed circuits fractionally, Combinatorica 15 (2) (1995) 281–288.

[22] D. Shmoys, Cut problems and their applications to divide-and-conquer, in: D. Hochbaum (Ed.), Approximation Algorithms for NP-Hard Problems, PWS Publishing, 1996, pp. 192–235.

[23] M. Yannakakis, P.C. Kanellakis, S.C. Cosmadakis, C.H. Papadimitriou, Cutting and partitioning a graph after a fixed pattern, in: Proceedings of the 10th International Colloquium on Automata, Languages and Programming, in: Lecture Notes in Computer Science, vol. 154, 1983, pp. 712–722.