The Multi-Multiway Cut Problem

Adi Avidor * and Michael Langberg **

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 $S_1, S_2, \dots, S_k$ of vertices. The goal is to find a subset of edges of minimum total weight whose removal completely disconnects each one of the sets $S_1, S_2, \dots, S_k$, 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 $1 \le i \le k$, 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 [GVY96] on the standard multicut problem. Namely, our algorithm has an $O(\log 2k)$ 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 $O(\log 2k)$ 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, where LP is the value of a natural LP-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 [Sey95] and Even *et al.* [ENSS98]).

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 $(s_1, t_1), \dots, (s_k, t_k)$ 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 $1 \le i \le k$. The problem is known to be APX-hard ([DJP$^+$94]). An $O(\log k)$-approximation algorithm for the problem was obtained by Garg, Vazirani and Yannakakis [GVY96].

* School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: `adi@tau.ac.il`. Work supported in part by THE ISRAEL SCIENCE FOUNDATION founded by The Israel Academy of Sciences and Humanities.

** Department of Computer Science, California Institute of Technology, Pasadena, California 91125. E-mail: `mikel@cs.caltech.edu`. Part of this work was done while visiting the School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel.

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 $1 \leq i < j \leq k$. The problem is also known to be APX-hard ([DJP$^+$94]). A $(\frac{3}{2} - \frac{1}{k})$-approximation algorithm for the problem was obtained by Calinescu, Karloff and Rabani [CKR98]. An improved $(1.3438 - \varepsilon_k)$-approximation algorithm for the problem was obtained by Karger *et al.* [KKS$^+$99]. In particular, for $k = 3$ the algorithm of Karger *et al.* [KKS$^+$99] achieves an approximation ratio of $12/11$, which matches the integrality gap of the linear programming relaxation. This result was also obtained independently by Cunningham and Tang [CT99].

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 $1 \leq i \leq k$, every two vertices $u, v \in S_i$. When $|S_i| = 2$, for all $1 \leq i \leq k$, 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.

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 [FF56]. The case in which $k = 2$ was also shown to be polynomially solvable by Yannakakis *et al.* [YKCP83] using multiple applications of the max-flow algorithm. For any $k \geq 3$ the problem was proven to be APX-hard by Dahlhaus *et al.* [DJP$^+$94] and thus cannot permit a PTAS (unless P=NP).

The currently best known approximation ratio for the minimum multicut problem is obtained in the work of Garg, Vazirani, and Yannakakis [GVY96]. 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 [LR99] and used also by Klein *et al.* [KRAR95]) 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 [Shm96]), the design of approximation algorithms for the minimum multicut problem on directed graphs [KPRT93,ENSS98,CKR01,Gup03], and the results recently obtained for the minimum correlation clustering problem [DI03,CGW03].

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 [GVY96] 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 multi-multiway cut problem and the closely related minimum uncut problem, we show that our result on light instances of minimum multi-multiway cut imply a result of independent interest on the minimum uncut problem. Our results can be summarized as follows.

Theorem 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 2 (Light multi-multiway cuts). *Let I be an instance to 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 (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 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 [1] have been considered. Namely, Seymour [Sey95] proved an existential result which implies (via [ENSS98]) 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 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 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 [PY91], and has an $O(\log |V|)$ approximation algorithm [GVY96]. The later follows by reducing an instance of the minimum uncut problem to an instance of minimum multicut in which there are $|V|$ pairs of terminals. The parameterized complexity of the minimum uncut problem has also been considered in the past (*e.g.* by Mahajan and Raman [MR99]). To the best of our knowledge, the question whether this problem is fixed parameter tractable remains open [Dow03]. In other words, given that an undirected graph can be made bipartite by deleting k of its edges, no $O(f(k)poly(|V|))$-time algorithm for finding such k edges is known (f is not necessarily polynomial in k). In this case, Corollary 1 implies an $O(poly(k, |V|))$-time algorithm which finds $O(k \log k)$ such edges. We elaborate on additional results regarding the minimum uncut problem in the discussion section and the appendix of this work.

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 and 2 appear in Section 2.4. In Section 3 we present the proof of Corollary 1. Finally, in Section 4 we briefly discuss some future research directions, regrading both the minimum multi-multiway cut and minimum uncut problems (some which are further elaborated in the Appendix).

2 The multi-multiway cut problem

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

[1] 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 between s_i and t_i for all $i = 1, \ldots, k$.

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:

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

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

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

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 ([GVY96]). 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 $Cut(S) = \{(u,v) \mid u \in S,\ v \in V \setminus S\}$, 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

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

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 latter.

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 $1 \leq j \neq j' \leq |S_i|$) 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 Figure 1. Roughly speaking, 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 $r_i \leq 1/2$ around all vertices in S_i. The edges in the cut produced by these balls are then 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 $\alpha \geq 0$ and $\delta \in [0, 1/2)$. We assume w.l.o.g. that $w(e) > 0$ for all $e \in E$. 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 value of the functions c_i and v_i may change during the execution of the algorithm, as edges are removed from E.

In the next subsection we prove the following theorem:

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 $1 \leq i \leq k$)
4. Set r_i to be the radius r in $\left(Dist_i \cup \{\frac{1}{2}\}\right) \cap (\delta, \frac{1}{2}]$ 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. $E \leftarrow E \setminus F$
8. Return CUT

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

Theorem 3. 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 3 we obtain Corollaries 2 and 3 below, which imply Theorems 1 and 2 stated in the Introduction, respectively.

Corollary 2. CutAlg$(1/k, 0)$ *is a* $4\ln(k+1)$*-approximation algorithm for multi-multiway cut.*

Corollary 3. *If* $w(e) \geq 1$ *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 3.

Lemma 1. *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* $c_i(r) = \frac{d}{dr}v_i(r)$.

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

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,

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

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

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

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

Proof. **(of Theorem 3):** 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 was chosen to be less or equal to 1/2 (by Step 5 of the algorithm this also holds for $r = 1/2$).

Let $I \subseteq \{1, \ldots, k\}$ be the set of group indices for which the algorithm entered the while loop. The weight of the multicut produced by the algorithm is at most $\sum_{i \in I} c_i(r_i^-)$ (recall the definition of r_i from Step 4 of our algorithm).

By Lemma 2, for each $i \in I$ there exists $r_i' \in (\delta, 1/2)$ such that $c_i(r_i') \leq \frac{2}{1-2\delta} \ln\left(\frac{(1+\alpha)LP}{v_i(\delta)}\right) v_i(r_i')$. By the choice of r_i (Step 4 of the algorithm), it is not hard to verify that $c_i(r_i^-)/v_i(r_i^-) \leq c_i(r_i')/v_i(r_i')$. This follows from the fact that the radius r in $(\delta, \frac{1}{2}]$ that minimizes the value of $\frac{c_i(r^-)}{v_i(r^-)}$ is actually in the set $Dist_i \cup \{\frac{1}{2}\}$. Therefore the weight of the multicut produced by the algorithm is at most $\sum_{i \in I} \frac{2}{1-2\delta} \ln\left(\frac{(1+\alpha)LP}{v_i(\delta)}\right) v_i(r_i^-)$. Since for all $i \in I$ there exists a path between at least two vertices in the group S_i,

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

Hence, the weight of the multicut produced by the algorithm is bounded by $\frac{2}{1-2\delta}\ln\left(\frac{(1+\alpha)LP}{\alpha LP+2\delta w_{min}}\right)\sum_{i\in I} v_i(r_i^-)$. Observe that by the definition of v_i and by Step 7 of the algorithm $\sum_{i\in I} v_i(r_i^-) \leq k\alpha LP + LP$. Therefore the proof is completed.

3 "Light" minimum uncut

Corollary 1. *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 [KRAR95] 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 3) a multicut of G' of weight at most $O(k \log k)$, which by the above implies our assertion.

4 Discussion

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 the standard relaxation of the multi-multiway cut problem we have presented in Section 2 in a direct manner due to its large integrality gap. This holds as well for the linear program enhanced with the so-called metric constraints.

When considering the minimum uncut problem, we find ourselves in a similar situation. Again, the standard linear programming relaxation has an integrality gap of $\Omega(\log n)$, which matches the currently best known approximation ratio for the problem (here n is the size of the given graph). In the appendix of this paper, we show that such an integrality gap holds also when the standard relaxation is enhanced with additional *odd cycle* or *metric* constraints. The integrality gap of the relaxation enhanced with *both* odd cycle and metric constraints is yet to be resolved.

One may ask similar questions regarding semidefinite programming relaxations of both the minimum multi-multiway cut problem and the minimum uncut problem. We have noticed that a *natural* semidefinite programming relaxation for minimum uncut (which is similar to the maximum cut semidefinite programming relaxation of [GW95]) has an integrality gap of $\Omega(n)$. This leads to the study of the quality of enhanced semidefinite relaxations, such with additional constraints. A standard set of constraints one may consider are the so-called triangle constraints (*e.g.* [FG95]). The integrality gap of such an enhanced relaxation is yet to be resolved. Nevertheless, one can show that using the standard *random hyperplane* rounding technique of [GW95] on such an enhanced relaxation cannot yield an approximation ratio better than $\Omega(\sqrt{n})$, implying that some other rounding technique is needed. Due to space limitations, our results on semidefinite programming relaxations for minimum uncut are omitted.

Acknowledgments

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

References

[CGW03] M. Chrikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. *In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003.

[CKR98] G. Calinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. *In Proceedings of the 30th Ann. ACM Symp. on Theory of Comp., ACM-SIAM*, 1998.

[CKR01] J. Cheriyan, H. J. Karloff, and Y. Rabani. Approximating directed multicuts. *In Proceedings of IEEE Symposium on Foundations of Computer Science*, 2001.

[CT99] W. H. Cunningham and L. Tang. Optimal 3-terminal cuts and linear programming. In *Proceedings of IPCO (Lecture Notes in Computer Science 1610)*, pages 114–125, 1999.

[DI03] E. D. Demaine and N. Immorlica. Correlation clustering with partial information. *In Proceedings of APPROX (Lecture Notes in Computer Science 2764)*, 2003.

[DJP+94] R. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

[Dow03] R. Downey. Parameterized complexity for the skeptic. *In Proceedings of 18th IEEE Conference on Computational Complexity*, 2003.

[ENSS98] G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

[FF56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[FG95] U. Feige and M. X. Goemans. Approximating the value of two prover proof systems with applications to Max-2-Sat and Max-Dicut. *In Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems*, pages 182–189, 1995.

[Gup03] A. Gupta. Improved results for directed multicut. *In Proceedings 14th Ann. ACM-SIAM Symp. on Discrete Algorithms, ACM-SIAM*, pages 454–455, 2003.

[GVY96] N. Garg, V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of ACM*, 42:1115–1145, 1995.

[KKS+99] D. R. Karger, P. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *In Proceedings of ACM Symposium on the Theory of Computing*, pages 668–678, 1999.

[KPRT93] P. N. Klein, S. A. Plotkin, S. Rao, and S. E. Tardos. New network decompositions theorems with applications. *Brown University Technical Report CS-93-30*, 1993.

[KRAR95] P. Klein, S. Rao, A. Agrawal, and R. Ravi. An approximate max-flow min-cut relation for undirected multicommodity flow, with applications. *Combinatorica*, 15:187–202, 1995.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[LR99] F. T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

[MR99] M. Mahajan and V. Raman. Parameterizing above guaranteed values: MaxSat and MaxCut. *Journal of Algorithms*, 31:355–354, 1999.

[PY91] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[Sey95] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.

[Shm96] D. Shmoys. Cut problems and their applications to divide-and-conquer. *In D. Hochbaum, editor, Approximation Algorithms for NP-Hard Problems, PWS Publishing*, pages 192–235, 1996.

[YKCP83] M. Yannakakis, P. C. Kanellakis, S. C. Cosmadakis, and C. H. Papadimitriou. Cutting and partitioning a graph after a fixed pattern. *In Proceedings of ICALP (Lecture Notes in Computer Science 154)*, pages 712–722, 1983.

Appendix A

LP relaxations of the minimum uncut problem

In this section we mention the limitations of a few natural linear programming relaxations of the minimum uncut problem. For a graph G we denote by n the number of vertices in G and by $\mathcal{C}$ the set of all odd cycles in G. The minimum uncut problem may be posed as the following integer program:

$$\begin{array}{l} \min \sum_{e\in E} w(e)x(e) \\ \qquad \sum_{e\in C} x(e) \geq 1 \;\; \forall C \in \mathcal{C} \\ \qquad\;\; x(e) \in \{0,1\} \quad \forall e \in E \end{array}$$

Here, if an edge $e \in E$ belongs to the cut we let $x(e) = 0$, otherwise $x(e) = 1$.

Viewing the non-edges of G as edges of weight 0, we may add a Boolean variable $x(ij)$ for each pair of vertices $i \neq j \in V$ such that $ij \notin E$, and add two sets of constraints. One set, called the *metric constraints*, corresponds to the fact that $1 - x$ forms a semi-metric on the vertices of G. This set includes the inequalities $1 - x(ij) \leq 1 - x(ik) + 1 - x(jk)$ for all distinct $i, j, k \in V$. The second set corresponds to any "cycle" of odd length ℓ and includes the inequalities $\sum_{j=1}^{\ell} x(i_j i_{j+1}) \geq 1$ for all $i_1, \ldots, i_\ell \in V$ (here, $i_{\ell+1} = i_1$).

Relaxing the integrality condition, we obtain the *standard linear programming relaxation* for minimum uncut:

$$\begin{array}{l} \min \sum_{e\in E} w(e)x(e) \\ \qquad \sum_{e\in C} x(e) \geq 1 \;\; \forall C \in \mathcal{C} \\ \qquad\;\; x(e) \geq 0 \qquad \forall e \in E \end{array}$$

As in the multicut linear programming relaxation, this relaxation can be solved in polynomial time regardless of the fact that it may involve exponentially many constraints. The next two propositions mention two integrality gaps.

Proposition 1. *For any sufficiently large n there is a graph G such that the standard linear programming relaxation of minimum uncut with the metric constraints has an integrality gap of $\Omega(\log n)$ on G.*

Proof. Let G be a constant degree expander graph with girth $g = \Omega(\log n)$ and optimal uncut of size $\Omega(n)$ (for existence of such graphs see for example the explicit construction of Lubotzky, Phillips and Sarnak [LPS88]). For each edge $e \in E$ assign $x(e) = 1/g$, and for each non-edge $ij \notin E$ assign $x(ij) = 1/2$. Then, the value of the linear programming relaxation is $O(|E|/g)$, while the optimal uncut is of size $\Omega(|E|)$.

Proposition 2. *For any sufficiently large n there is a graph G such that the standard linear programming relaxation of minimum uncut with the second set of constraints has an integrality gap of $\Omega(\log n)$ on G.*

Proof. Use the same graph as in Proposition 1. For each edge $e \in E$ assign $x(e) = 1/g$, and for each non-edge $ij \notin E$ assign $x(ij) = 1$.