Optimal 3-terminal cuts and linear programming

William H. Cunningham[1] and Lawrence Tang[2]

[1] Department of Combinatorics & Optimization, University of Waterloo, Waterloo, ON, Canada, N2L 3G1

[2] Department of Mathematics, University of British Columbia, Vancouver, BC, Canada V6T 1Y8

Abstract. Given an undirected graph $G = (V, E)$ and three specified terminal nodes t_1, t_2, t_3, a *3-cut* is a subset A of E such that no two terminals are in the same component of $G \backslash A$. If a non-negative edge weight c_e is specified for each $e \in E$, the *optimal 3-cut problem* is to find a 3-cut of minimum total weight. This problem is NP-hard, and in fact, is max-SNP-hard. An approximation algorithm having performance guarantee $\frac{7}{6}$ has recently been given by Călinescu, Karloff, and Rabani. It is based on a certain linear programming relaxation, for which it is shown that the optimal 3-cut has weight at most $\frac{7}{6}$ times the optimal LP value. It is proved here that $\frac{7}{6}$ can be improved to $\frac{12}{11}$, and that this is best possible. As a consequence, we obtain an approximation algorithm for the optimal 3-cut problem having performance guarantee $\frac{12}{11}$.

1 Introduction

Given an undirected graph $G = (V, E)$ and k specified terminal nodes $t_1, \ldots, t_k$, a *k-cut* is a subset A of E such that no two terminals are in the same component of $G \backslash A$. If a non-negative edge-weight c_e is specified for each $e \in E$, the *optimal k-cut problem* is to find a k-cut of minimum total weight. This problem was shown by Dahlhaus, Johnson, Papadimitriou, Seymour, and Yannakakis [5] to be NP-hard for $k \geq 3$. (Of course, it is solvable in polynomial time if $k = 2$.) They also gave a simple polynomial-time algorithm having performance guarantee $\frac{2(k-1)}{k}$, that is, it is guaranteed to deliver a k-cut of weight at most $\frac{2(k-1)}{k}$ times the minimum weight of a k-cut. Later, in [6], the same authors showed that for $k \geq 3$ the problem is max-SNP-hard, which implies that, assuming P$\neq$NP, there exists a positive ε such that the problem has no polynomial-time approximation algorithm with performance guarantee $1 + \varepsilon$.

The present paper concentrates on the optimal 3-cut problem. From the above remarks, it follows that this problem is max-SNP-hard, and the approximation algorithm of [6] has a performance guarantee of $\frac{4}{3}$. Recently, Călinescu, Karloff, and Rabani [1] gave an approximation algorithm having a performance guarantee of $\frac{7}{6}$. We give a further improvement that is based on their approach.

Chopra and Rao [3] and Cunningham [4] investigated linear programming relaxations of the 3-cut problem, showing results on classes of facets and separation algorithms. Here are the two simplest relaxations. (By a *T-path* we mean the edge-set of a path joining two of the terminals. By a *wye* we mean the edge-set of a tree having exactly three nodes of degree one, each of which is a terminal. For a set A, a subset B of A, and a vector $z \in \mathbf{R}^A$, $z(B)$ denotes $\sum_{j \in B} z_j$.)

$$(LP1) \qquad \begin{array}{rl} \text{minimize} & \sum_{e \in E} c_e x_e \\ \text{subject to} & \\ & x(P) \geq 1,\ P \text{ a } T\text{-path} \\ & x_e \geq 0,\ e \in E. \end{array}$$

$$(LP2) \qquad \begin{array}{rl} \text{minimize} & \sum_{e \in E} c_e x_e \\ \text{subject to} & \\ & x(P) \geq 1,\ P \text{ a } T\text{-path} \\ & x(Y) \geq 2,\ Y \text{ a wye} \\ & x_e \geq 0,\ e \in E. \end{array}$$

It follows from some simple observations about shortest paths, and the equivalence of optimization and separation, that both problems can be solved in polynomial time. It was proved in [4] that the approximation algorithm of [5] delivers a 3-cut of value at most $\frac{4}{3}$ times the optimal value of $(LP1)$. (In particular, the minimum weight of a 3-cut is at most $\frac{4}{3}$ times the optimal value of $(LP1)$.) It was conjectured that the minimum weight of a 3-cut is at most $\frac{16}{15}$ times the optimal value $(LP2)$. The examples in Figure 1 (from [4]) show that this conjecture, if true, is best possible. In both examples, the values of a feasible solution x of $(LP2)$ are shown in the figure. The weights c_e are all 2 for the example on the left. For the one on the right they are 1 for the edges of the interior triangle, and 2 for the other edges. In both cases the minimum 3-cut value is 8, but the given feasible solution of $(LP2)$ has value 7.5.

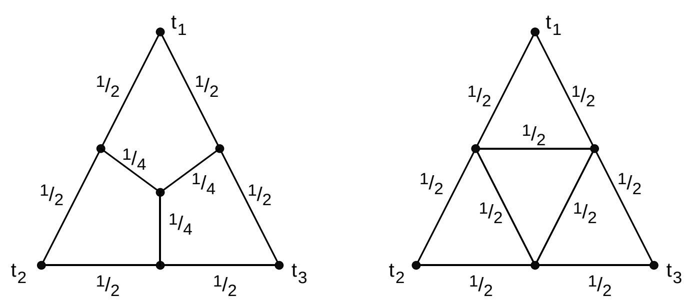

Fig. 1. Bad examples for $(LP2)$

Recently, Călinescu, Karloff, and Rabani [1] gave a new linear programming relaxation. Although their approach applies to any number k of terminals, we continue to restrict attention to the case when $k = 3$. They need to assume that G be a complete graph. (Of course, if any missing edges are added with weight zero, the resulting 3-cut problem is equivalent to the given one, so this assumption is not limiting.) The relaxation is based on the following observations. First, every minimal 3-cut is of the form $\beta(R_1, R_2, R_3)$, where $t_i \in R_i$ for all i. Here, where $\mathcal{R}$ is a family of disjoint subsets of $\mathcal{R}$, $\beta(\mathcal{R})$ denotes the set of all edges of G joining nodes in different members of the family. Since $c \geq 0$, there is an optimal 3-cut of this form. Second, the incidence vector x of a minimal 3-cut is a kind of distance function, that is, it defines a function $d(v, w) = x_{vw}$ on pairs of nodes of G which is non-negative, symmetric, and satisfies the triangle inequality. Finally, with respect to d the distance between any two terminals is 1, and the sum of the distances from any node v to the terminals is 2. The resulting linear-programming relaxation is:

$$
(LP3) \qquad \begin{array}{ll} \text{minimize} & \sum_{e \in E} c_e x_e \\ & \text{subject to} \\ & \qquad x_{vw} = 1,\ v, w \in T,\ v \neq w \\ & \sum_{v \in T} x_{vw} = 2,\ w \in V \\ & x_{uv} + x_{vw} - x_{uw} \geq 0,\ u, v, w \in V \\ & \qquad x_e \geq 0,\ e \in E. \end{array}
$$

This relaxation is at least as tight as $(LP2)$. To see this, suppose that (after adding missing edges to make G complete), we have a feasible solution x to $(LP3)$. Then for any path P of G joining u to v, $x(P) \geq x_{uv}$, by applying the triangle inequality. It follows that $x(P) \geq 1$ for any T-path P. Moreover, any wye Y is the disjoint union of paths P_1, P_2, P_3 from some node v to the terminals. It follows that $x(Y) \geq \sum_{w \in T} x_{vw} = 2$. Thus every feasible solution of $(LP3)$ gives a feasible solution of $(LP2)$ having the same objective value. The first example of Figure 1 shows that the optimal value of $(LP3)$ can be strictly greater than the optimal value of $(LP2)$. On the other hand, the second example shows that there is no hope to prove in general that the the minimum weight of a 3-cut is less than $\frac{16}{15}$ times the optimal value of $(LP3)$.

It was proved in [1] that the minimum weight of a 3-cut is at most $\frac{7}{6}$ times the optimal value of $(LP3)$. As a consequence, an approximation algorithm for the optimal 3-cut problem having a performance guarantee of $\frac{7}{6}$ was derived. (It is clear that $(LP3)$ can be solved in polynomial time, since it is of polynomial size.) However, it was left open whether this result could be strengthened; the second example of Figure 1 shows an example for which the minimum weight of a 3-cut can be as large as $16/15$ times the optimal value of $(LP3)$, and this is the worst example given in [1]. (To see that x of that example does extend to a feasible solution of $(LP3)$, we simply define x on each missing edge uv to be the minimum length, with respect to lengths x_e, of a path from u to v.)

In this paper it is shown that the minimum weight of a 3-cut is at most $\frac{12}{11}$ times the optimal value of $(LP3)$, and that this is best possible. (This result has been obtained independently by Karger, Klein, Stein, Thorrup, and Young [7].) As a consequence we obtain an approximation algorithm for the optimal 3-cut problem having a performance guarantee of $\frac{12}{11}$.

2 Triangle embeddings

Călinescu, Karloff, and Rabani [1] introduced an extremely useful geometric relaxation, which they showed was equivalent to the linear-programming relaxation $(LP3)$. Let $\triangle$ denote the convex hull of the three elementary vectors $e^1 = (1,0,0)$, $e^2 = (0,1,0)$, and $e^3 = (0,0,1)$ in $\mathbf{R}^3$. By a *triangle embedding* of G we mean a mapping y from V into $\triangle$ such that $y(t_i) = e^i$ for $i = 1,2,3$. A triangle embedding y determines a vector $x \in \mathbf{R}^E$ as follows. For each edge uv, let x_{uv} be one-half the L_1 distance from $y(u)$ to $y(v)$. It is easy to see that this x is a feasible solution to $(LP3)$. Conversely, a feasible solution x of $(LP3)$ determines a triangle embedding y as follows. For each node v, let $y(v) = (1 - x_{t_1 v}, 1 - x_{t_2 v}, 1 - x_{t_3 v})$.

Given a triangle embedding y we can obtain x as above, and then use x to obtain a triangle embedding y'. It is easy to see that $y = y'$. It is not true, on the other hand, that every feasible solution of $(LP3)$ arises in this way from a triangle-embedding. However, it is "almost true". The following result is implicit in [1], and we include a proof for completeness.

Theorem 1. *Let x be a feasible solution of $(LP3)$, let y be the triangle embedding determined by x and let x' be the feasible solution of $(LP3)$ determined by y. Then $x' \le x$, and if x is an optimal solution of $(LP3)$, so is x'.*

Proof. First, observe that the second statement is a consequence of the first and the non-negativity of c. Now let $uv \in E$. Both $y(u)$ and $y(v)$ have component-sum 1. Therefore, $y(u) - y(v)$ has component-sum zero, and so one-half of the L_1 distance between $y(u)$ and $y(v)$ is the sum of the non-negative components of $y(u) - y(v)$. Hence we may assume, perhaps by interchanging u with v and relabelling the terminals, that one-half of the L_1 distance between $y(u)$ and $y(v)$ is the sum of the first two components of $y(u) - y(v)$. Therefore,

$$\begin{aligned}\frac{1}{2}\|y(u) - y(v)\|_1 &= y_1(u) - y_1(v) + y_2(u) - y_2(v)\\ &= 1 - x_{ut_1} - (1 - x_{vt_1}) + 1 - x_{ut_2} - (1 - x_{vt_2})\\ &= (2 - x_{ut_3}) - (2 - x_{vt_3})\\ &\le x_{uv},\end{aligned}$$

as required.

□

The approximation algorithm of Călinescu, Karloff, and Rabani uses the following ideas. Suppose that $(LP3)$ is solved, and an optimal solution x^* that arises from a triangle embedding is found. For a number α between 0 and 1 that is different from x^*_{rv} for every $v \in V$ and $r \in T$, and an ordering r, s, t of T, define $R_r = \{v \in V : x^*_{rv} < \alpha\}$, $R_s = \{v \in V \backslash R_r : x^*_{sv} < \alpha\}$, $R_t = V \backslash (R_r \cup R_s)$. We call the 3-cut $\beta(R_r, R_s, R_t)$ *uniform* (with respect to this x^*). It is easy to see that there are $O(n)$ uniform 3-cuts. The algorithm of [1] simply chooses the uniform 3-cut having minimum weight. It is proved to have weight at most $\frac{7}{6}$ times the minimum weight of a 3-cut.

We consider a slight generalization of the notion of uniform 3-cut. Let α, α' be two numbers chosen as α was above, and let r, s, t be an ordering of T. Define $R_r = \{v \in V : x^*_{rv} < \alpha\}$, $R_s = \{v \in V \backslash R_r : x^*_{sv} < \alpha'\}$, $R_t = V \backslash (R_r \cup R_s)$. We call the 3-cut $\beta(R_r, R_s, R_t)$ *flat* (with respect to this x^*). Clearly, every uniform 3-cut is flat. It is easy to see that there are $O(n^2)$ flat 3-cuts. Our approximation algorithm simply chooses the flat 3-cut having minimum weight. We will show that it has weight at most $\frac{12}{11}$ times the weight of an optimal 3-cut. This result is based on a tight analysis of the bound for the optimal 3-cut problem given by $(LP3)$.

3 Linear programming again

It is easy to check that if the optimal value of $(LP3)$ is zero, then there is a 3-cut of weight zero. Therefore, we may assume that the optimal value is positive. So our problem may be restated as finding the best upper bound, over all choices of G and c, for the minimum weight of a 3-cut divided by the optimal value of $(LP3)$. By multiplying c by an appropriate positive number, we may assume that the minimum weight of a 3-cut is 1. It is now more convenient to prove the best lower bound on the value of $(LP3)$. Surprisingly, we can use a different linear programming problem to do this.

Assume that G is fixed, and that an optimal solution x^* of $(LP3)$ is also fixed. Then the problem of finding the worst optimal value can be stated as:

$$(P) \qquad \begin{array}{c} \text{minimize } \sum_{e \in E} c_e x^*_e \\ \text{subject to} \\ c(S) \geq 1, \;\; S \text{ a 3-cut} \\ c_e \geq 0, \;\; e \in E. \end{array}$$

Note that the variables are the weights c_e! It may seem that the hypothesis that G and x^* are known is very strong, but it turns out that we can assume that there are not many choices for them. First, we may assume that x^* is rational, since it is an optimal solution of a linear-programming problem having rational data. Therefore, there exists a positive integer q such that qx^* is integer-valued. Second, we may assume that x^* arises from a triangle-embedding y^*, and it is easy to see that qy^* is integral, as well. Therefore, we can think of y^* as embedding the nodes of G into a finite subset $\triangle_q$ of $\triangle$, consisting of those points $y \in \triangle$ for which qy is integral. We define the planar graph $G_q = (\triangle_q, E_q)$ by $uv \in E_q$ if and only if the L_1 distance between u and v is $\frac{2}{q}$. Figure 3 shows G_9; the numbers there are explained later. For nodes u, v of G_q, we denote by $d_q(u, v)$ the least number of edges of a path in G_q from u to v. (It is easy to see that $d_q(u, v) = \frac{q}{2}$ times the L_1 distance from u to v.)

Theorem 2. *Let G, c be a 3-cut instance, let x^* be a rational-valued optimal solution of $(LP3)$, with corresponding triangle-embedding y^*, and let q be a positive integer such that qx^* is integral. Then there is a 3-cut instance on graph $\hat{G}$ with nodeset $\triangle_q$ and edge-weights $\hat{c}$ such that:*

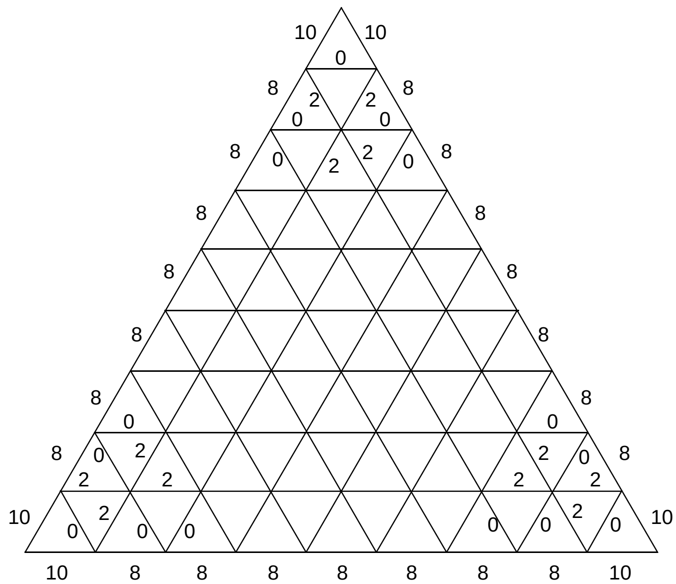

Fig. 2. G_9

(a) $\hat{x}$ *defined by* $q\hat{x}_{uv} = d_q(u, v)$ *for all* $uv \in E$ *is a feasible solution of* $(LP3)$ *(for* $\hat{G}, \hat{c}$*), and* $\hat{c}\hat{x} \leq cx^*$*;*

(b) The optimal 3-cut value for $\hat{G}, \hat{c}$ *is at least that for* G, c*;*

(c) $\hat{c}_e = 0$ *for all* $e \notin E_q$*;*

(d) For every flat 3-cut of $\hat{G}$ *with respect to* $\hat{x}$*, there is a flat 3-cut of* G *with respect to* x^* *having no larger weight.*

Proof. We use the mapping y^* from V to $\triangle_q$, and we assume that x^* arises from y^*. Suppose that two nodes u, v of G are mapped to the same point of $\triangle_q$ by y^*. Form G' by identifying u with v and, where multiple edges are formed, replacing the pair by a single edge whose weight is their sum. Then every 3-cut of G' determines a 3-cut of G having the same weight, so the minimum weight of a 3-cut of G' is at least the minimum weight of a 3-cut of G. Moreover, x^* also determines a triangle-embedding of G', so there is a feasible solution of $(LP3)$ for G' having value cx^*. Finally, every flat cut of G' gives a flat cut of G of the same weight. Thus the theorem is true for G if it is true for G', and so we may assume that y^* is one-to-one.

Now suppose that y^* is not onto, that is, that there is an element z of $\triangle_q$ such that $y^*(v) \neq z$ for all $v \in V$. We can form a graph G' from G by adding a node v and an edge uv of weight zero for every $u \in V$. It is easy to see that the minimum weight of a 3-cut of G' is the same as that of G. Also, if we map the new node to z, we get a triangle embedding of G', and it corresponds to a feasible solution of $(LP3)$ on G' having value equal to cx^*. Finally, every flat cut of G' corresponds to a flat cut of G of the same weight. So the theorem is true for G if it is true for G'. It follows that we may assume that y^* is onto. Therefore, we may assume that $V = \triangle_q$, and that y^* is the identity mapping.

Now suppose that there exists $uv \in E \backslash E_q$, such that $c_{uv} = \varepsilon > 0$. Let P be the edge-set of a path in G_q from u to v such that $|P| = d_q(u, v)$. Decrease c_{uv} to zero, and increase c_e by ε for all $e \in P$. We denote the new c by c'. Then, since every 3-cut using e uses an edge from P, the minimum weight of a 3-cut with respect to c' is not less than that with respect to c. (Similarly, every flat 3-cut has value with respect to c' not less than that with respect to c.) Now $c'x^* = cx^* - \varepsilon d_q(u, v) + \varepsilon d_q(u, v) = cx^*$. This argument can be repeated as long as there is such an edge uv.

□

It is a consequence of the above theorem that it is enough to study the 3-cut problem on graphs G_q with $x_e^* = \frac{1}{q}$ for all $e \in E_q$. (That is, to obtain the best bound on the ratio of the optimal weight of a 3-cut to the optimal value of $(LP3)$, it suffices to analyze such graphs and weights.) In particular, for each positive integer q, we are interested in the optimal value of the following linear programming problem.

$$
(P_q) \qquad \begin{array}{rl} \text{minimize} & \frac{1}{q} \sum_{e \in E} c_e \\ & \text{subject to} \\ & c(S) \geq 1, \;\; S \text{ a 3-cut of } G_q \\ & c_e \geq 0, \;\; e \in E_q \end{array}
$$

The dual problem is

$$
(D_q) \qquad \begin{array}{rl} \text{maximize} & \sum z_S \\ & \text{subject to} \\ & \sum_{e \in S} \leq \frac{1}{q}, \;\; e \in E_q \\ & z_S \geq 0, \;\; S \text{ a 3-cut of } G_q. \end{array}
$$

We actually solved these problems numerically for several values of q, and then were able to find solutions for general q.

Theorem 3. *For $q \geq 4$ the optimal value of (P_q) and of (D_q) is equal to*

$$
f(q) = \begin{cases} \frac{11}{12} + \frac{1}{12(q+1)}, & \text{if } q \equiv 0 \text{ mod } 3 \\ \frac{11}{12} + \frac{1}{12q}, & \text{if } q \equiv 1 \text{ mod } 3 \\ \frac{11}{12} + \frac{1}{12q} - \frac{1}{12q^2}, & \text{if } q \equiv 2 \text{ mod } 3 \end{cases}
$$

Moreover, there is an optimal solution of (D_q) for which z_S is positive only if S is a flat 3-cut.

It is easy to see that Theorems 2 and 3 have the following consequence. This result has been proved independently by Karger *et al.* [7], whose approach is somewhat different, but also uses a linear programming analysis of triangle-embedding.

Theorem 4. *For any 3-cut instance, the minimum weight of a 3-cut is at most $\frac{12}{11}$ times the optimal value of $(LP3)$, and the constant $\frac{12}{11}$ is best possible.* □

4 An improved approximation algorithm

Algorithm

1. Find a rational-valued optimal solution x^* of $(LP3)$.
2. Find the triangle embedding y^* determined by x^*.
3. Return the flat 3-cut of minimum weight.

As pointed out before, the first step can be performed in polynomial time. The polynomial-time algorithms for linear programming can be modified to return a rational-valued optimal solution, and one of polynomial size. The second is easy. So is the third step, using the observation made earlier that there are only $O(n^2)$ flat 3-cuts of G.

Theorem 5. *The above algorithm returns a 3-cut of weight at most* $\frac{12}{11}$ *times the minimum weight of a 3-cut.*

Proof. We may assume that the optimal value of a 3-cut is 1, so it is enough to prove that the algorithm delivers a 3-cut of weight at most $\frac{12}{11}$. Let x^* be a rational-valued optimal solution for $(LP3)$, and let q be a common denominator for the components of x^*, such that q is a multiple of 3. Consider an optimal solution z^* of (D_q) as given by Theorem 3. Then

$$\sum_S \frac{12}{11} z_S^* \geq 1,$$

and $z_S^* > 0$ only if S is a flat 3-cut. Therefore

$$\begin{aligned} \min_{z_S^*>0} c(S) &\leq \sum_S \frac{12}{11} z_S^* c(S) \\ &= \frac{12}{11} \sum_S z_S^* c(S) \\ &= \frac{12}{11} \sum_{e \in E} c_e \sum_{e \in S} z_S^* \\ &\leq \frac{12}{11} \sum_{e \in E} c_e x_e^* \\ &\leq \frac{12}{11}. \end{aligned}$$

□

5 Proof of Theorem 3

To prove Theorem 3, it is enough to give feasible solutions of (P_q) and of (D_q) having objective value $f(q)$. For simplicity, we will actually do something weaker. For the case when $q \equiv 0 \bmod 3$, we give a feasible solution of (P_q) having objective value $f(q)$, and a feasible solution to (D_q) using only variables corresponding to flat 3-cuts having objective value $\frac{11}{12}$. Although this does not quite prove Theorem 3, it is enough to prove Theorems 4 and 5, since a common denominator for the components of x^* can always be chosen to be a multiple of 3.

First, we describe our feasible solution to (P_q). Consider Figure 2 which shows G_9. Let c'_e be the number next to edge e, or 1 if no number appears. It is easy to see that the minimum value of a 3-cut is 40, so $c = c'/40$ is a feasible solution to (P_9). Its objective value is the sum of the components of c' divided by 9, which is $\frac{37}{40}$.

Here is the general construction (when q is a multiple of 3) for an optimal solution of (P_q). If $q = 3m$, divide $\triangle_q$ into three "corner triangles" of side m together with the "middle hexagon". Put $c'_e = 3m+1$ for all edges incident with the terminals. Put $c'_e = 2m+2$ for all other edges on the boundary of $\triangle_q$. Put $c'_e = m-1$ for each edge e in a corner triangle that is parallel to an outside edge and distance 1 from it. Put $c'_e = 1$ for all other edges in the middle hexagon (including its boundary). Put $c'_e = 0$ for all other edges.

It is easy to convince oneself that the minimum weight of a 3-cut with respect to c' is $4(3m+1)$, and hence that $c = c'/4(3m+1)$ is a feasible solution to (P_q). Here is a sketch of a proof. (The ideas come, essentially, from the result of Dahlhaus, *et al.* [5], showing that there is a polynomial-time algorithm to solve the optimal multiterminal cut problem when G is planar and the number of terminals is fixed.) Any minimal 3-cut of G_q has the form $\beta(R_1, R_2, R_3)$. There are two kinds of such 3-cuts, corresponding to the case in which there is a pair i, j for which there is no edge joining a node in R_i to a node in R_j, and the one where this is not true. The minimum value of a 3-cut of the first type is simply the sum of the weights of two cuts, each separating a terminal from the other two. In the case of G_q with c' described above, to show that any such cut has weight at least $4(3m+1)$, it is enough to show (due to the symmetry of c') that any cut separating one terminal from the other two has weight at least $2(3m+1)$. This is done by exhibiting an appropriate flow of this value from one terminal to the other two.

The second type of 3-cut corresponds to the union of three paths in the planar dual of G_q, such that the three paths begin at the same face triangle and end with edges that are on different sides of the outside face. Finding a minimum-weight such 3-cut can be accomplished by, for each choice of the face triangle, solving a shortest path problem. Therefore, to show that any 3-cut of the second type has c'-weight at least $4(3m+1)$, one shows that, for each choice of face triangle, there is an appropriate "potential" on the faces of G_q.

To compute the objective value of this feasible solution (P_q), note that there are 6 edges e having $c'_e = 3m+1$, $3(3m-2)$ edges e having $c'_e = 2m+2$, $6(m-1)$ edges e having $c'_e = m-1$, and $9m^2$ edges e having $c'_e = 1$. From this we get that the total c'-weight of all the edges is $3m(11m+12)$. To obtain the objective value of the resulting c in (P_q), we divide by $4(3m+1)(3m)$, and this gives $f(q)$ for $q = 3m$.

Now we need to show a feasible solution of (D_q) having objective value $\frac{11}{12}$. This requires a weighting of the flat 3-cuts of G_q. We assign positive dual variables to two kinds of 3-cuts. For each integer j, $1 \leq j < m$ and each choice of two terminals r, s we consider the (uniform) 3-cut $\beta(R_r(j), R_s(j), V \backslash (R_r(j) \cup R_s(j)))$ where, for $t = r, s$, $R_r(j) = \{v \in V_q : d_q(t, v) < j\}$. There are $3m$ such 3-cuts S, and for each of them we set $z_S = \frac{1}{4q}$. Notice that these variables contribute to the left-hand side of the main constraint of (D_q) only for certain edges, namely, those that are contained in the corner triangles and are parallel to one of the two sides of $\triangle$ that meet at that corner. For each of these edges, the total contribution is exactly $1/2q$.

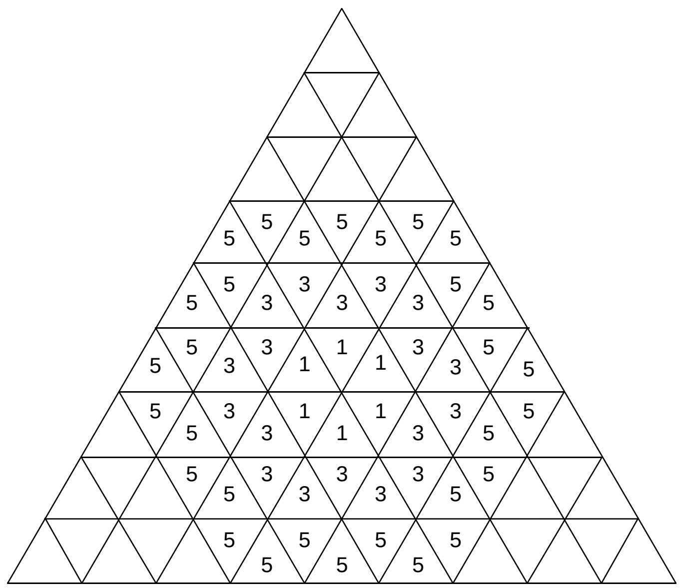

Fig. 3. Feasible solution of (D_9)

The weights assigned to the second type of flat cut are determined by a weighting of the face triangles of G_q that are contained in the middle hexagon. See Figure 3, where such a weighting of the face triangles is indicated for G_9. Let us use the term *row* in the following technical sense. It is defined by a straight line through the centre of a face triangle and parallel to one of its three sides. When we speak of the face triangles *in* the row, we mean all of the face triangles that are intersected by the line. When we speak of the edges *in* the row, we mean all of the edges that are intersected by the line. Notice that in the figure, the sum of the weights of the face triangles in each row is the same, namely 35. It is obvious how to extend this pattern to find a weighting with this property for any $q = 3m$. Then the sum of the weights of the face triangles in any row is $4m^2 - 1$.

Given a face triangle, consider the set of all edges in the three rows containing the triangle. It is possible to choose two flat 3-cuts of G_q whose union is this set, and whose intersection is a single edge, or is the set of edges of the face triangle. (There is more than one way to do this.) For each of these two 3-cuts, assign a weight equal to the weight of the triangle divided by $2q(4m^2-1)$. (Note that a 3-cut S may be assigned weight by two different face triangles; these weights are added to form the variable z_S.) Now consider the constraint of (D_q) corresponding to an edge e. The contribution of the variables just defined to the left-hand side of the constraint, is at most the sum of the weights of the face triangles in rows containing the edge. If the edge is in the middle hexagon, or is in a corner triangle and is not parallel to one of the edges incident with the corner, then it gets contributions from triangles in two different rows, and otherwise, it gets contributions from triangles in one row. Therefore, the contribution for the first type of edge is at most $(4m^2-1)/(4m^2-1)q = \frac{1}{q}$. For the second type of edge the total contribution is at most half this, that is, at most $\frac{1}{2q}$. But the second group of edges consists precisely of the ones that get a contribution from the dual variables assigned to the uniform 3-cuts, and that contribution is $\frac{1}{2q}$. So the total contribution of all of the dual variables to the left-hand side of the constraint of (D_q) corresponding to any edge e is at most $\frac{1}{q}$, so we have defined a feasible solution of (D_q).

Now the objective value of this solution can be computed as follows. There are $3m$ variables corresponding to uniform 3-cuts, each given value $\frac{1}{4q}$. Therefore, the contribution to the objective function of variables of this type is $\frac{3m}{12m} = \frac{1}{4}$. The contribution of the other variables is the sum of over the $2m$ horizontal rows in the middle hexagon, of the total weight of a row divided by $q(4m^2-1)$. Therefore, it is

$$2m(4m^2-1)/q(4m^2-1) = \frac{2}{3}.$$

Therefore, the objective value of our feasible solution to (D_q) is

$$\frac{1}{4} + \frac{2}{3} = \frac{11}{12}.$$

6 Remarks

Since the constant $\frac{12}{11}$ is best possible in Theorem 4, it is natural to ask whether it is best possible in Theorem 5. Note, however, that the family of examples that we use to show the tightness of the LP bound, all have the property that there is a flat 3-cut that is optimal. Therefore, these examples are not at all bad for the approximation algorithm. However, it seems likely that $\frac{12}{11}$ is indeed best possible in Theorem 5. For several values of q Kevin Cheung [2] has constructed examples in which the optimal solution of $(LP3)$ has denominator q, and the approximation algorithm returns a 3-cut of value at least $\frac{1}{f(q)}$ times the optimal value of $(LP3)$. Actually, his examples seem to be the first that show that our approximation algorithm does not always return an optimal solution. In fact, no such example seems to have been known even for the simpler algorithm of Călinescu *et al.* [1].

All of the results of Călinescu *et al.* [1] quoted above for $k = 3$ are special cases of their results for general k. They give a linear-programming relaxation that generalizes $(LP3)$, and a corresponding generalization of the notion of triangle-embedding, an embedding into a $(k-1)$-dimensional simplex in which the terminals are mapped to the extreme points. They show that the optimal value of a k-cut is at most $\frac{3k-2}{2k}$ times the optimal value of this linear-programming problem. As a result, they obtain an approximation algorithm for the optimal k-cut problem having performance guarantee $\frac{3k-2}{2k}$. The recent paper [7], which has most of our results for $k = 3$, also has results for $k > 3$, improving the bounds given by [1]. For example, [7] gives bounds of 1.1539 for $k = 4$ and 1.3438 for all $k > 6$. The problem of giving a tight analysis for $k > 3$, as we now have for $k = 3$, remains open.

Acknowledgment. We are grateful to Gruia Călinescu, Joseph Cheriyan, Kevin Cheung, and Levent Tunçel for conversations about this work.

References

1. G. Călinescu, H. Karloff, and Y. Rabani: An improved approximation algorithm for MULTIWAY CUT Proceedings of Symposium on Theory of Computing, ACM, 1998.
2. Kevin Cheung, private communication, 1999.
3. S. Chopra and M.R. Rao, "On the multiway cut polyhedron", Networks 21(1991), 51–89.
4. W.H. Cunningham, "The optimal multiterminal cut problem", in: C. Monma and F. Hwang (eds.), *Reliability of Computer and Communications Networks*, American Math. Soc., 1991, pp. 105–120.
5. E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis, "The Complexity of multiway cuts", extended abstract, 1983.
6. E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis, "The Complexity of multiterminal cuts", SIAM J. Computing, 23(1994), 864–894.
7. D. Karger, P. Klein, C. Stein, M. Thorrup, and N. Young, "Rounding algorithms for a geometric embedding of minimum multiway cut," Proceedings of Symposium on Theory of Computing, ACM, 1999, to appear.