Algorithmica (2000) 27: 198–207
DOI: 10.1007/s004530010013

Algorithmica
© 2000 Springer-Verlag New York Inc.

Approximation Algorithms for Minimum K-Cut

N. Guttmann-Beck[1] and R. Hassin[1]

Abstract. Let $G = (V, E)$ be a complete undirected graph, with node set $V = \{v_1, \dots, v_n\}$ and edge set E. The edges $(v_i, v_j) \in E$ have nonnegative weights that satisfy the triangle inequality. Given a set of integers $K = \{k_i\}_{i=1}^p$ ($\sum_{i=1}^p k_i \le |V|$), the *minimum K-cut problem* is to compute disjoint subsets with sizes $\{k_i\}_{i=1}^p$, minimizing the total weight of edges whose two ends are in different subsets. We demonstrate that for any fixed p it is possible to obtain in polynomial time an approximation of at most three times the optimal value. We also prove bounds on the ratio between the weights of maximum and minimum cuts.

Key Words. Approximation algorithms, Minimum cuts.

1. Introduction. Let $G = (V, E)$ be a complete undirected graph with nonnegative edge weights $l(e)$, $e \in E$, and let K be a set of p positive integers $\{k_i\}_{i=1}^p$ such that $\sum_{i=1}^p k_i \le |V|$. A *K-cut* is a collection of disjoint node sets $\{P_i\}_{i=1}^p$ such that $\forall i \in \{1, \dots, p\}$ $|P_i| = k_i$. The *minimum K-cut problem* is to find a K-cut such that $\sum_{i=1}^{p-1} \sum_{j=i+1}^p l(P_i, P_j)$ (where $l(P_i, P_j) = \sum_{v\in P_i} \sum_{u \in P_j} l(v, u)$) is minimized.

The version of this problem in which $\sum_{i=1}^p k_i = |V|$ and the sizes of the sets are not given but their number is required to be p, and p is fixed, is polynomially solvable [2]. When p is part of the input, this version is NP-hard [1] and Saran and Vazirani [3] gave a $2 - 2/p$ approximation algorithm. If, in addition, the sets in the partition are restricted to be of equal size, the problem is only known to be approximable within $|V|(p-1)/p$ by Saran and Vazirani [3].

We consider the minimum K-cut problem under the assumption that the weights satisfy the triangle inequality. The problem is still NP-hard under this assumption because each graph can be made to satisfy the triangle inequality while not changing the problem by adding a high enough constant (for example, $\sum_{e\in E} l(e)$) to all the edge lengths.

We demonstrate that for any fixed p it is possible to obtain in polynomial time an approximation of at most three times the optimal value.

Suppose now $\sum_{i=1}^p k_i = |V|$ (partitioning all the nodes in the graph). Let $l_{\max}$, $l_{\min}$ be the weights of the maximum and minimum cuts in the graph, respectively, and let $r = l_{\max}/l_{\min}$. We prove bounds on r for several interesting cases, assuming the triangle inequality. For example, for the special case of partitioning the graph into two equal-sized sets we prove that $r \le 2$. Therefore, in this case, every cut can serve as an approximation, with weight at most twice the optimal. When the graph must be partitioned into two unequal-sized sets r can be arbitrarily large and we prove a bound which depends on the sizes of the two sides of the cut.

[1] Department of Statistics and Operations Research, Tel Aviv University, Tel Aviv 69978, Israel. {nili,hassin}@math.tau.ac.il.

Received September 4, 1997; revised July 15, 1998. Communicated by H. N. Gabow.

2. The Approximation Algorithm.

We will use the following definitions:

For a set of edges E, $l(E) = \sum_{e \in E} l(e)$.
For a set of nodes U, and a node v, $l(v, U) = \sum_{u \in U} l(v, u)$.
Denote by *opt* the value of a minimum K-cut.

We start by defining a new problem, the min-star problem, which we solve optimally for a constant p. We then use its solution to approximate the minimum K-cut problem. The *min-star problem* is to find vertices $v_1, \ldots, v_p$ and a K-cut such that $v_i \in P_i$, $i = 1, \ldots, p$, and $\sum_{i=1}^{p} k_i l(v_i, \bigcup_{j \neq i} P_j) + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} k_i k_j l(v_i, v_j)$ is minimized. The problem is called min-star because for fixed $v_1, \ldots, v_p$ the contribution to the objective function which is associated with v_i is proportional to the sum of edge weights of a star subgraph whose center is v_i.

THEOREM 2.1. *Algorithm Find_Partition (see Figure 1) solves the min-star problem. It can be executed in time $O(n^{(p+1)})$.*

```
Find_Partition
  input
  1. A graph $G = (V, E)$, $|V| = n$, with weights $l(e)$ $e \in E$ .
  2. Constants $k_1, \ldots, k_p$ such that $\sum_{i=1}^{p} k_i \leq n$.
  returns
  1. $\{v_1, \ldots, v_p\} \subset V$.
  2. A partition $P_1, \ldots, P_p$ of $V$, such that $v_i \in P_i, |P_i| = k_i, i = 1, \ldots, p$.
  begin
  for every subset $\{v_1, \ldots, v_p\} \subset V$ :
        $\{a_1, \ldots, a_{n-p}\} := V \setminus \{v_1, \ldots, v_p\}$.
        Compute $\bar{x}$, an optimal solution to the following transportation problem:
        minimize $\sum_{i=1}^{p} \sum_{j=1}^{n-p} \Big( \sum_{r=1, r \neq i}^{p} k_r l(v_r, a_j) \Big) x_{ij}$
        subject to
                $\sum_{i=1}^{p} x_{ij} \leq 1, \quad j = 1, \ldots, n-p$ ,
                $\sum_{j=1}^{n-p} x_{ij} = k_i - 1, \quad i = 1, \ldots, p$ ,
                $x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, p, \; j = 1, \ldots, n-p$ .

        $P_i^{\{v_1, \ldots, v_p\}} := \{v_i\} \cup \{a_j | 1 \leq j \leq n-p \; \bar{x}_{ij} = 1\}, i = 1, \ldots, p.$
        $d^{\{v_1, \ldots, v_p\}} := \sum_{i=1}^{p} k_i l(v_i, \cup_{j \neq i} P_j^{\{v_1, \ldots, v_p\}}) + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} k_i k_j l(v_i, v_j).$
        end for
  Find $\{v_1^*, \ldots, v_p^*\} \subset V$ for which $d^{\{v_1^*, \ldots, v_p^*\}}$ is minimal.
  return $(v_1^*, \ldots, v_p^*, P_1^{\{v_1^*, \ldots, v_p^*\}}, \ldots, P_p^{\{v_1^*, \ldots, v_p^*\}})$.
  end Find_Partition
```

Fig. 1. Algorithm *Find_Partition*.

PROOF. Let $\bar{v_1}, \dots, \bar{v_p}, \bar{V_1}, \dots, \bar{V_p}$ be an optimal solution for the *min-star problem*. Since the algorithm checks all the subsets of V of size p it also checks the subset $\{\bar{v_1}, \dots, \bar{v_p}\}$. For this subset the sum $\sum_{i=1}^{p-1} \sum_{j=i+1}^{p} k_i k_j l(\bar{v_i}, \bar{v_j})$ is constant, so we need to find a partition $(P_1, \dots, P_p)$ which minimizes $\sum_{i=1}^{p} k_i l(\bar{v_i}, \bigcup_{j \neq i} P_j)$, this is achieved by finding an optimal solution to the transportation problem (where $x_{ij} = 1$ if and only if vertex a_j is assigned to the subset P_i).

For a fixed value of p we can solve the transportation problem in time $O(n)$, using the algorithms given by Tokuyama and Nakano [4]. There are $O(n^p)$ subsets $\{v_1, \dots, v_p\} \subset V$, so altogether the time complexity is $O(n^{p+1})$. □

We now show that the weight of the partition found as an optimal solution for the min-star problem is no more than $3opt$.

THEOREM 2.2. *Let $(v_1, \dots, v_p, P_1, \dots, P_p)$ be the output of Find_Partition. Let $O_1, \dots, O_p$ be a minimum K-cut. Then*

$$apx = \sum_{i<j} l(P_i, P_j) \le 3 \sum_{i<j} l(O_i, O_j) = 3\, opt.$$

PROOF. Denote by S^* the optimal solution value for the min-star problem. By the triangle inequality

$$\begin{aligned} apx &= \sum_{i<j} l(P_i, P_j) = \sum_{i<j} \sum_{\substack{u_i \in P_i \\ u_j \in P_j}} l(u_i, u_j) \\ &\le \sum_{i<j} \sum_{\substack{u_i \in P_i \\ u_j \in P_j}} (l(u_i, v_j) + l(v_i, v_j) + l(v_i, u_j)) \\ &= \sum_{i<j} (k_j l(v_j, P_i) + k_i k_j l(v_i, v_j) + k_i l(v_i, P_j)) \\ &= \sum_{i=1}^{p} k_i l(v_i, \cup_{j \neq i} P_j) + \sum_{i<j} k_i k_j l(v_i, v_j) = S^*. \end{aligned}$$

On the other hand, according to Theorem 2.1 *Find_Partition* solves the min-star problem so that, for every $u_1 \in O_1, \dots, u_p \in O_p$,

$$S^* = \sum_{i=1}^{p} k_i l\left(v_i, \bigcup_{j \neq i} P_j\right) + \sum_{i<j} k_i k_j l(v_i, v_j) \le \sum_{i=1}^{p} k_i l\left(u_i, \bigcup_{j \neq i} O_j\right) + \sum_{i<j} k_i k_j l(u_i, u_j).$$

Summing over all $(u_1, \dots, u_p)$ such that $u_1 \in O_1, \dots, u_p \in O_p$ we get that

$$\begin{aligned} S^* \prod_{l=1}^{p} k_l &\le \sum_{i=1}^{p} \left(\prod_{l=1}^{p} k_l\right) l\left(O_i, \bigcup_{j \neq i} O_j\right) + \sum_{i<j} \left(\prod_{l=1}^{p} k_l\right) l(O_i, O_j) \\ &= \left(\prod_{l=1}^{p} k_l\right) \left(\sum_{i \neq j} l(O_i, O_j) + \sum_{i<j} l(O_i, O_j)\right) \end{aligned}$$

$$= \left(\prod_{l=1}^{p} k_l\right)\left(3\sum_{i<j} l(O_i, O_j)\right) = 3\left(\prod_{l=1}^{p} k_l\right) opt .$$

Hence $S^* \leq 3\,opt$, giving that $apx \leq S^* \leq 3\,opt$. □

We now show that the bound of Theorem 2.2 is the best possible. Consider the graph shown in Figure 2. It has $2n + n^2 + 2$ nodes classified into sets A and B each containing n nodes, a set C containing n^2 nodes, and two distinguished nodes x and y. The edge weights are set according to the following rules:

- An edge connecting two nodes inside each set is of weight 1.
- $\forall a \in A, b \in B, l(a, b) = 2.$
- $\forall a \in A, c \in C, l(a, c) = 1.$
- $\forall b \in B, c \in C, l(b, c) = 3.$
- $\forall a \in A, l(a, x) = l(a, y) = 1.$
- $\forall b \in B, l(b, x) = 2, l(b, y) = 1.$
- $\forall c \in C, l(c, x) = 1, l(c, y) = 2.$
- $l(x, y) = 1.$

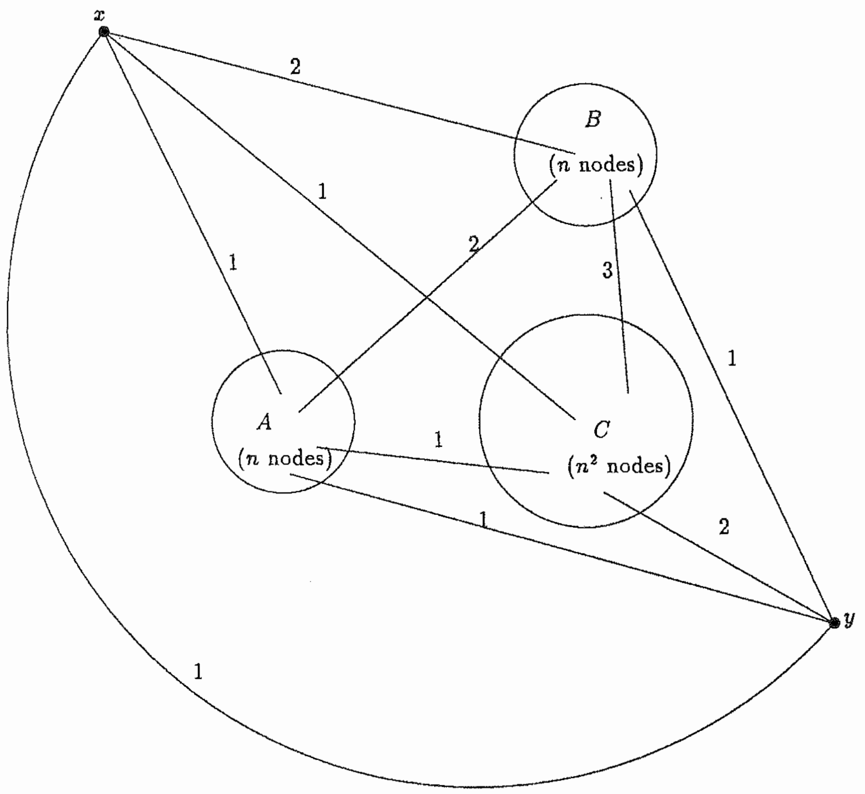

Fig. 2. An example where $apx \approx 3\,opt$.

We wish to partition this graph into two sets, one containing $n+1$ nodes and the other containing n^2+n+1 nodes. When we choose the nodes x, y, as centers of stars, then the partition $\{x\} \cup B, \{y\} \cup A \cup C$ defines an optimal solution for the minimum star problem since the stars only use edges of weight 1. Hence *Find_Partition* may output the cut $\{x\} \cup B, \{y\} \cup A \cup C$. The weight of this cut is dominated by $l(B, C) = 3n^3$.

On the other hand, the partition $\{x\} \cup A, \{y\} \cup B \cup C$ gives a cut whose value is dominated by $l(A, C) = n^3$. So in this case (when n goes to infinity) $apx = 3\,opt$.

3. Bounds on $l_{\max}/l_{\min}$.

In this section we assume $\sum_{i=1}^{p} k_i = |V|$ and show that we can bound the ratio between the maximum and minimum cuts of a graph. For example, we show that the triangle inequality assumption implies that any cut can be used as an approximation in the case of two equal-sized sets, with weight at most $2opt$.

3.1. *Partitioning into Two Equal-Sized Sets.*

Given a graph $G = (V, E)$ with edge weights satisfying the triangle inequality, suppose we wish to partition the graph into two equal-sized sets. Let (P_1, P_2) be a minimum cut, and let (R_1, R_2) be a maximum cut. Denote $A = P_1 \cap R_1$, $B = P_2 \cap R_1$, $C = P_1 \cap R_2$, and $D = P_2 \cap R_2$ (see Figure 3). For these definitions $l_{\min} = l(P_1, P_2)$ and $l_{\max} = l(R_1, R_2)$. Then

$$l_{\min} = l(P_1, P_2) = l(A, B) + l(B, C) + l(A, D) + l(C, D), \tag{1}$$

while

$$l_{\max} = l(R_1, R_2) = l(A, C) + l(B, C) + l(A, D) + l(B, D). \tag{2}$$

LEMMA 3.1. *If $V_1, V_2, V_3 \subset V$ are disjoint, then*

$$|V_3| l(V_1, V_2) \le |V_2| l(V_1, V_3) + |V_1| l(V_3, V_2).$$

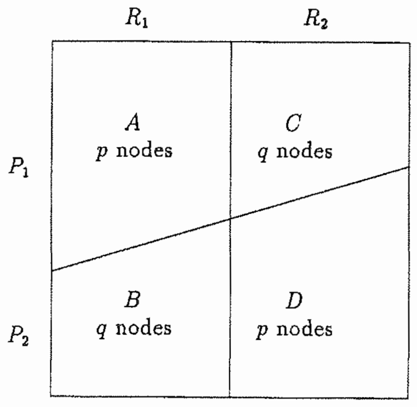

Fig. 3. The sets A, B, C, and D.

PROOF. By the triangle inequality, for every $w \in V_3$,

$$\begin{aligned} l(V_1, V_2) &= \sum_{v \in V_1} \sum_{u \in V_2} l(v, u) \le \sum_{v \in V_1} \sum_{u \in V_2} (l(v, w) + l(w, u)) \\ &= |V_2| l(w, V_1) + |V_1| l(w, V_2). \end{aligned}$$

The claimed inequality is obtained by summation over all $w \in V_3$. □

We denote $|A| = |D| = p$ and $|B| = |C| = q$.

LEMMA 3.2.

$$\min \left\{ \frac{2q}{p} l(A, D), \frac{2p}{q} l(B, C) \right\} \le l(A, D) + l(B, C).$$

PROOF. Suppose otherwise, then

$$\frac{2q}{p} l(A, D) > l(A, D) + l(B, C),$$

and

$$\frac{2p}{q} l(B, C) > l(A, D) + l(B, C).$$

This gives

$$\frac{2q - p}{p} l(A, D) > l(B, C)$$

and

$$\frac{2p - q}{q} l(B, C) > l(A, D),$$

so that

$$\frac{2q - p}{p} l(A, D) > \frac{1}{(2p - q)/q} l(A, D).$$

Since $l(A, D) \ge 0$ this implies

$$\frac{2q - p}{p} \frac{2p - q}{q} > 1,$$

or

$$-2(p - q)^2 > 0.$$

A contradiction. □

THEOREM 3.3.

$$l_{\max} \le 2 l_{\min}.$$

PROOF. The following inequalities can be concluded from Lemma 3.1:

$$\begin{aligned} l(A,C) &\le \frac{q}{p} l(A,D) + l(C,D), \\ l(B,D) &\le \frac{q}{p} l(A,D) + l(A,B), \\ l(A,C) &\le \frac{p}{q} l(B,C) + l(A,B), \\ l(B,D) &\le \frac{p}{q} l(B,C) + l(C,D). \end{aligned}$$

Hence

$$l(A,C) + l(B,D) \le l(C,D) + l(A,B) + \min\left\{ \frac{2q}{p} l(A,D), \frac{2p}{q} l(B,C) \right\}.$$

Using Lemma 3.2 and (1),

$$l(A,C) + l(B,D) \le l_{\min}.$$

By (1) also

$$l(B,C) + l(A,D) \le l_{\min},$$

and we get from (2) that $l_{\max} \le 2l_{\min}$, as claimed. □

To see that the bound of Theorem 3.3 is tight consider a graph with $4n$ nodes: $v_1, \dots, v_{2n}, u_1, \dots, u_{2n}$. The weights are set according to the following rules:

- $\forall i, j, l(v_i, v_j) = 0$.
- $\forall i, j, l(u_i, u_j) = 0$.
- $\forall i, j, l(v_i, u_j) = 1$.

We wish to partition this graph into two equal-sized sets each containing $2n$ nodes. The cut $\{v_1, \dots, v_{2n}\}, \{u_1, \dots, u_{2n}\}$ has weight $4n^2$. On the other hand, the cut $\{v_1, \dots, v_n\} \cup \{u_1, \dots, u_n\}, \{u_{n+1}, \dots, u_{2n}\} \cup \{v_{n+1}, \dots, v_{2n}\}$ has weight of $2n^2$. So in this case $l_{\max} = 2l_{\min}$.

3.2. *Partitioning into k Equal-Sized Sets*

LEMMA 3.4. *If (A, B) is a partition of V into two equal-sized sets, then $l(A) + l(B) \le 2l(A, B)$.*

PROOF. By the triangle inequality, for every four nodes $a_1, a_2 \in A$, $b_1, b_2 \in B$,

$$\begin{aligned} l(a_1,a_2) &\le l(a_1,b_1) + l(a_2,b_1), \\ l(a_1,a_2) &\le l(a_1,b_2) + l(a_2,b_2), \\ l(b_1,b_2) &\le l(a_1,b_1) + l(a_1,b_2), \\ l(b_2,b_2) &\le l(a_2,b_1) + l(a_2,b_2). \end{aligned}$$

Summing these four inequalities and dividing by 2 we get

$$l(a_1, a_2) + l(b_1, b_2) \leq l(a_1, b_1) + l(a_2, b_1) + l(a_1, b_2) + l(a_2, b_2).$$

Summing this over all a_1, a_2, b_1, b_2 we get

$$2|V|^2(l(A) + l(B)) \leq 4|V|^2 l(A, B),$$

hence

$$l(A) + l(B) \leq 2l(A, B). \qquad \square$$

LEMMA 3.5. *Let $G = (V, E)$ be a graph with $|V| = kn$. If $\{P_i\}_{i=1}^k$ is a partitioning of V into k equal-sized sets then the value of the cut, lc, satisfies $lc = \sum_{i<j} l(P_i, P_j) \geq ((k-1)/(k+1))l(E)$.*

PROOF. By Lemma 3.4, for $i \neq j$,

$$l(P_i) + l(P_j) \leq 2l(P_i, P_j).$$

Summing over $j = 1, \ldots, k,\ j \neq i$, we get

$$(k-1)l(P_i) + \sum_{j=1}^{k} l(P_j) - l(P_i) \leq 2 \sum_{\substack{j=1 \\ j \neq i}}^{k} l(P_i, P_j).$$

Summing over $i = 1, \ldots, k$ we get

$$(k-1)\sum_{i=1}^{k} l(P_i) + k\sum_{j=1}^{k} l(P_j) - \sum_{i=1}^{k} l(P_i) \leq 2\sum_{i \neq j} l(P_i, P_j),$$

or

$$2(k-1)\sum_{i=1}^{k} l(P_i) \leq 4lc.$$

Since $\sum_{i=1}^{k} l(P_i) = l(E) - lc$ we get

$$(k-1)(l(E) - lc) \leq 2lc,$$

giving the required result:

$$lc \geq \frac{k-1}{k+1} l(E). \qquad \square$$

REMARK 3.6. This bound is tight. For example, let $G = (V_1 \cup V_2, E)$ where $|V_1| = |V_2| = n$. Let all the edges connecting nodes inside V_1 or V_2 have a weight of two units and all the edges connecting one node from V_1 and one node from V_2 have a unit weight. For the cut which separates V_1 from $V_2, l(V_1, V_2) = n^2$. Since $l(V_1) = l(V_2) = n(n-1)$ we get that $l(E) = 3n^2 - 2n$, giving that asymptotically $l(V_1, V_2) = L(E)/3$.

THEOREM 3.7. *Let $l_{\max}$ and $l_{\min}$ denote the weight of maximum and minimum (respectively) cuts of a given graph into k equal-sized sets. Then*

$$(k-1)l_{\max} \le (k+1)l_{\min}.$$

PROOF. Since the edges in $l_{\max}$ are a subset of E then by Lemma 3.5

$$l_{\max} \le l(E) \le \frac{k+1}{k-1} l_{\min}. \qquad \square$$

3.3. *Partitioning into Two Unequal-Sized Sets.* First we show that when partitioning the graph into two unequal-sized sets, the weight of the maximum cut may be much bigger than the weight of the minimum cut. Then we show that we can still bound the maximum cut, but the bound depends on of the sizes of the two node sets.

Let $V = \{u_1, \dots, u_{n-1}\} \cup \{v\}$. For each $i, j \in \{1, \dots, n-1\}$ set $l(u_i, u_j)$ to 0. For each $i \in \{1, \dots, n-1\}$ set $l(v, u_i)$ to 1. Suppose we wish to partition this graph into two sets, one containing $n-1$ nodes, and the other containing 1 node. In this case, a best partition would be $\{u_1\}, \{u_2, \dots, u_{n-1}\} \cup \{v\}$ giving a cut of weight 1. However, the partition $\{v\}, \{u_1, \dots, u_{n-1}\}$ gives a cut of weight $n-1$.

LEMMA 3.8. *Let $A = \{a_1, \dots, a_t\}$, $B = \{b_1, \dots, b_s\}$, $s \le t$, be a partitioning of V. Then $l(E) \le (2 + (t-2)/s) l(A, B)$.*

PROOF. By the triangle inequality, for every i, j, k,

$$l(a_i, a_j) \le l(a_i, b_k) + l(a_j, b_k).$$

Summing this inequality over $j \in \{1, \dots, t\} \backslash \{i\}$ (for a fixed k) we get

$$\sum_{\substack{j=1 \\ j \ne i}}^{t} l(a_i, a_j) \le (t-1) l(a_i, b_k) + \sum_{j=1}^{t} l(a_j, b_k) - l(a_i, b_k).$$

Summing over $i \in \{1, \dots, t\}$ we get that, for every $k \in \{1, \dots, s\}$,

$$\begin{aligned} 2\sum_{i<j} l(a_i, a_j) &\le (t-1)\sum_{i=1}^{t} l(a_i, b_k) + t\sum_{j=1}^{t} l(a_j, b_k) - \sum_{i=1}^{t} l(a_i, b_k) \\ &= 2(t-1)\sum_{i=1}^{t} l(a_i, b_k) \\ &= 2(t-1) l(A, b_k). \end{aligned}$$

Summing over $k = 1, \dots, s$ and dividing by $2s$ we get

$$l(A) = \sum_{i<j} l(a_i, a_j) \le \frac{t-1}{s} l(A, B).$$

Similarly,

$$l(B) = \sum_{i<j} l(b_i, b_j) \le \frac{s-1}{t} l(A, B).$$

By the last two inequalities and the assumption $s \le t$,

$$\begin{aligned} s(l(A) + l(B)) &= s\left(\sum_{i<j} l(a_i, a_j) + \sum_{i<j} l(b_i, b_j)\right) \\ &\le \left(s\frac{t-1}{s} + s\frac{s-1}{t}\right) l(A, B) \\ &\le (t - 1 + s - 1) l(A, B) = (s + t - 2) l(A, B). \end{aligned}$$

Since $l(A) + l(B) = l(E) - l(A, B)$,

$$s(l(E) - l(A, B)) \le (s + t - 2) l(A, B),$$

proving

$$l(E) \le \left(2 + \frac{t-2}{s}\right) l(A, B). \qquad \square$$

REMARK 3.9. This bound is tight. Suppose that $s = t = n$. Substituting in the lemma's bound we get $l(E) \le (3 - 2/n) l(A, B)$. The example in Remark 3.6 has $s = t = n$ and $l(E) = 3n^2 - 2n$. It presents a cut with $l(A, B) = n^2$ so that the bound is achieved.

THEOREM 3.10. *Let $l_{\max}$ and $l_{\min}$ denote the weight of maximum and minimum cuts, respectively, of a given graph into two sets S,T with $|S|/|T| = s/t$ $(s \le t)$, then $l_{\max} \le (2 + (t-2)/s) l_{\min}$.*

PROOF. Since the edges in the maximum cut are a subset of E it follows from Lemma 3.8 that

$$l_{\max} \le l(E) \le \left(2 + \frac{t-2}{s}\right) l_{\min}. \qquad \square$$

References

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, CA, 1979.

[2] O. Goldschmidt and D. S. Hochbaum, A polynomial algorithm for the k-cut problem for fixed k, *Mathematics of Operations Research*, **19** (1994), 24–37.

[3] H. Saran and V. Vazirani, Finding k cuts within twice the optimal, *SIAM Journal on Computing*, **24** (1995), 101–108.

[4] T. Tokuyama and J. Nakano, Geometric algorithms for the minimum cost assignment problem, *Random Structures and Algorithms*, **6** (1995), 393–405.