Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 385 (2007) 78–87
Theoretical Computer Science
www.elsevier.com/locate/tcs

Approximation algorithms for maximum cut with limited unbalance

Giulia Galbiati[a,*], Francesco Maffioli[b]

[a] *Dipartimento di Informatica e Sistemistica, Università degli Studi di Pavia, 27100 Pavia, Italy*
[b] *Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy*

Received 17 May 2006; received in revised form 18 May 2007; accepted 23 May 2007

Communicated by G. Ausiello

Abstract

We consider the problem of partitioning the vertices of a weighted graph into two sets of sizes that differ at most by a given threshold B, so as to maximize the weight of the crossing edges. For B equal to 0 this problem is known as Max Bisection, whereas for B equal to the number n of nodes it is the maximum cut problem. We present polynomial time randomized approximation algorithms with non trivial performance guarantees for its solution. The approximation results are obtained by extending the methodology used by Y. Ye for Max Bisection and by combining this technique with another one that uses the algorithm of Goemans and Williamson for the maximum cut problem. When B is equal to zero the approximation ratio achieved coincides with the one obtained by Y. Ye; otherwise it is always above this value and tends to the value obtained by Goemans and Williamson as B approaches the number n of nodes.

Keywords: Randomized approximation algorithm; Semidefinite programming; Maximum cut; Limited unbalance cut

1. Introduction

Problems addressing optimum cuts are often considered in combinatorial optimization and in theoretical computer science; recently unbalanced graph cuts have received attention [9]. Here we address the following problem: given an undirected graph $G = (V, E)$, with vertex set V of cardinality n and edge set E, where each edge (i, j) has a nonnegative weight w_{ij}, and given a constant B, $0 \leq B < n$, find a cut $(S, V \setminus S)$ of G of maximum weight such that the difference between the cardinalities of the two shores of the cut is not greater than B. We refer to it as the Maximum Cut with Limited Unbalance (MaxCUT-LU for short) problem. When B is equal to zero it is known as the Max Bisection problem and the algorithm in [12] gives the best approximation ratio equal to 0.699. When B is equal to $n - 1$ it is the well-known maximum cut problem and the famous algorithm of [8] gives an approximation ratio equal to 0.87856. It is known that the maximum cut problem is strongly NP-hard and cannot have a PTAS unless $P = \mathrm{NP}$ [4].

* Corresponding author. Tel.: +39 0382 985360; fax: +39 0382 985373.
E-mail addresses: giulia.galbiati@unipv.it (G. Galbiati), maffioli@elet.polimi.it (F. Maffioli).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2007.05.036

In [10] several several applications of Maximum Cut are reported in different fields such as network planning, circuit design, scheduling, cryptanalysis, logic, psychology. For most of them the generalization to MaxCUT-LU makes sense. For instance in circuit design, the problem of dividing the vertex set of the graph underlined by the circuit into two parts of equal cardinalities is of interest, and relaxing the equal cardinality constraint to that of limited unbalance can allow to get better results as far as approximating the optimum weight of the cut obtained, without affecting the suitability of the partition from the point of view of the circuit designer. MaxCUT-LU might also make sense for balancing signed graphs when special constraints arise, with applications e.g. in psychology [2]. The strict relation between Maximum Cut and Maximum 2-Satisfiability problems is well known [8]. Here the extension to MaxCUT-LU can allow to consider constraints on the number of variables set to "true".

In this paper we present polynomial time randomized approximation algorithms with nontrivial performance guarantees for MaxCUT-LU. Our results are obtained by extending to this problem the methodology used in [12] for Max Bisection and by combining this technique with another one that uses the algorithm of [8]. When B is equal to zero the approximation ratio achieved coincides with the one of [12], which is equal to 0.699; otherwise it is always greater than this value, and tends to the 0.87856 value of the algorithm of [8] when B approaches the number n of nodes. Our main results are summarized in Theorem 8, Proposition 9 and Theorem 11. In Table 1 at the end of this work we report the approximation ratios r obtained for some values of η, where $\eta = B/n$ is our unbalance parameter, and θ is another parameter used in the algorithm and described in Section 3.1.

A problem related to MaxCUT-LU and addressed in [1] and [5] is the problem of finding a cut of maximum weight such that the cardinality of one shore of the cut is equal to a given integer k. The approximation ratios that the authors achieve with sophisticated techniques are naturally weaker than ours: for instance in [5], for the case of $k = B = n/3$, they achieve a 0.58 ratio against our 0.797. Other problems addressing optimal cuts with side constraints have recently received attention, as described in the introductory section of [9].

The formulation of MaxCUT-LU is introduced in the next section. The algorithms used to solve the problem are presented in the subsequent sections, the first devoted to the case when η is small, the other when η is large. The last section concludes the work, summarizes the results and presents some of them in Table 1.

2. The formulation

MaxCUT-LU can be formulated by assigning to each vertex i a binary variable $x_i \in \{-1, 1\}$, with vertices on the same shore of the cut receiving the same value, and by setting $w_{ij} = 0$ if $(i, j) \notin E$, as:

$$w^* := \max \left\{ \frac{1}{4} \sum_{i,j} w_{ij}(1 - x_i x_j) : \sum_{i,j} x_i x_j \le B^2;\, x_i \in \{-1, 1\}, i = 1, \dots, n \right\}. \tag{1}$$

The semidefinite relaxation of this binary quadratic programme can be formulated as follows:

$$w^{\mathrm{SDP}} := \max \left\{ \frac{1}{4} \sum_{i,j} w_{ij}(1 - X_{ij}) : \sum_{i,j} X_{ij} \le B^2;\, X_{ii} = 1, i = 1, \dots, n;\, \mathbf{X} \in M_n \right\} \tag{2}$$

where M_n is the set of real, symmetrical, positive semidefinite matrices of order n. It is easy to see that any solution $\mathbf{x}$ of (1) yields a solution $\mathbf{X}$ of (2) with $X_{ij} = x_i x_j$. Hence obviously $w^* \le w^{\mathrm{SDP}}$.

It is known that such a SDP program can be solved to any degree of accuracy in polynomial time, i.e. $\forall \epsilon > 0$, we can find in time polynomial in the length of the instance and in $\log(1/\epsilon)$ a solution of (2) having $\frac{1}{4}\sum_{i,j} w_{ij}(1 - X_{ij}) \ge w^{\mathrm{SDP}} - \epsilon$ (see e.g. [3]). From an almost optimal solution of the SDP program one can then derive a solution of the integer programme using appropriate rounding techniques.

Rounding techniques applied to the solution of SDP relaxations of combinatorial optimization problems in order to get integral solutions of guaranteed degree of approximation have been pioneered by Goemans and Williamson [8] for the Max CUT and Max SAT problems. Frieze and Jerrum [6] have developed such techniques further, addressing the Max Bisection and the Max-k Cut problems. Ye [12] has improved the approximation ratio for Max Bisection using a more sophisticated rounding technique. With respect to Max CUT, problem MaxCUT-LU presents, as Max Bisection, the extra difficulty of having to deal with two objectives: the weight of the cut and the size of its shores.

3. The algorithm for small unbalance

We now present our first algorithm, suitable for solving problem MaxCUT-LU when η is small. In the algorithm, I indicates the identity matrix, N is set equal to $\frac{n^2}{4} - \frac{B^2}{4}$, i.e. to the minimum value of the product of the cardinalities of the shores of a cut with limited unbalance; the parameters θ and k are fixed by the algorithm in an appropriate way, as specified in Section 3.1 entirely devoted to this aspect. Functions $\alpha(\theta)$ and $\beta(\theta, \eta)$ are defined in (3) and (4) and their meaning is made clear in Lemma 2.

The algorithm uses the following technique, introduced in [12], which refines the one in [8]: from a solution $\widetilde{X}$ of the SDP relaxation first it constructs a new matrix X as a convex combination of $\widetilde{X}$ and the identity matrix I; then to matrix X, which is positive definite, it applies the Cholesky decomposition to obtain vectors $(\mathbf{v_1}, \ldots, \mathbf{v_n})$ on the unit n-dimensional sphere S_n. The algorithm then uses the so called random hyperplane technique, i.e. it repeatedly generates a uniformly distributed vector $\mathbf{r}$ on the unit sphere, computes vector $\mathbf{u} = (\mathbf{r} \cdot \mathbf{v_1}, \ldots, \mathbf{r} \cdot \mathbf{v_n})$ and then rounds $\mathbf{u}$ to a vector $\widehat{\mathbf{x}}$ with $\widehat{x}_i \in \{-1, 1\}$, and $\widehat{x}_i = -1$ iff $u_i \geq 0, i = 1, \ldots, n$. Each vector $\widehat{\mathbf{x}}$ hence identifies a cut $(S, V \setminus S)$ of G, where $S = \{i : \widehat{x}_i = 1\}$ or $S = \{i : \widehat{x}_i = -1\}$; in our algorithm we always choose wlog S to be the set of vertices with the larger cardinality.

In the analysis of our algorithm, for the sake of clarity, we assume that $\widetilde{X}$ is an optimum solution of the SDP relaxation and that the vectors of the Cholesky decomposition exactly satisfy the equalities $(\mathbf{v_i} \cdot \mathbf{v_j}) = X_{ij}$. It can be shown that the inaccuracies resulting from using an almost optimal solution $\widetilde{X}$ and an almost exact Cholesky decomposition can be absorbed into the approximation factor presented in Theorem 8 (see Chapter 26 of [11]). This ensures that the algorithm runs in polynomial time.

We now describe function *rebalance*(S), invoked by the algorithm when $|S| > (n + B)/2$. Let $S = \{i_1, \ldots, i_s\}$ and denote by $\delta^c(i)$ the contribution of vertex i to the weight of the cut $(S, V \setminus S)$, i.e. $\delta^c(i) = \sum_{j \notin S} w_{ij}$, and $w(S) = \sum_{i \in S} \delta^c(i)$. Assume wlog that $\delta^c(i_1) \leq \cdots \leq \delta^c(i_s)$; then function *rebalance*(S) reduces the number of nodes in S to $(n + B)/2$ by moving from S to $(S, V \setminus S)$ the first $s - (n + B)/2$ vertices, which less contribute to the weight of the cut.

Throughout this paper, $w(S)$ denotes the weight of the cut $(S, V \setminus S)$.

Algorithm 1.

1 - Solve the SDP problem (2) and let $\widetilde{X}$ be the solution matrix;
2 - fix a value θ with $0 \leq \theta < 1$ and a positive integer k;
3 - let $X = \theta \widetilde{X} + (1 - \theta) I$;
4 - apply Cholesky decomposition to X to obtain vectors $(\mathbf{v_1}, \ldots, \mathbf{v_n})$;
5 - $S_R = \phi$;
6 - repeat for k times the following {
  6.1 - generate a uniformly distributed vector $\mathbf{r}$ on the unit sphere;
  6.2 - compute $\mathbf{u} = (\mathbf{r} \cdot \mathbf{v_1}, \ldots, \mathbf{r} \cdot \mathbf{v_n})$;
  6.3 - round $\mathbf{u}$ to vector $\widehat{\mathbf{x}} \in \{-1, 1\}^n$ identifying a cut $(S, V \setminus S)$;
  6.4 - if $|S| \leq (n + B)/2$ /* the cut is feasible for MaxCUT-LU */
    let $\widetilde{S} = S$ else let $\widetilde{S} =$ *rebalance*(S);
  6.5 - if $w(\widetilde{S}) > w(S_R)$ /* a better cut for MaxCUT-LU is found */
    let $S_R = \widetilde{S}$;
}
7 - return S_R.

In order to analyse the quality of the solution S_R returned by the algorithm, we define

$$\alpha(\theta) := \min_{-1 \leq y < 1} \frac{1 - \frac{2}{\pi} \arcsin(\theta y)}{1 - y} \tag{3}$$

and

$$\beta(\theta, \eta) := \left(1 - \frac{1}{n}\right) \frac{1}{1 - \eta^2} b(\theta) + c(\theta) \tag{4}$$

with

$$b(\theta) = 1 - \frac{2}{\pi}\arcsin(\theta) \text{ and } c(\theta) = \min_{-1\le y<1}\frac{2}{\pi}\frac{\arcsin(\theta) - \arcsin(\theta y)}{1-y}. \tag{5}$$

Notice that the definition of $\alpha(\theta)$ is as in [12] whereas that of $\beta(\theta, \eta)$ is different.

Lemma 2. *If functions $\alpha(\theta)$ and $\beta(\theta, \eta)$ are defined as in (3) and (4), then for the random variable $w(S)$, related to the cut $(S, V \setminus S)$ generated by Algorithm 1 at line 6.3, we have that $E[w(S)] \ge \alpha(\theta)w^*$ and $E[|S|(n - |S|)] \ge \beta(\theta, \eta)N$.*

Proof. In [8,6] it is proved that the probability that vertices i and j are separated in the cut identified by S is equal to $\frac{1}{2}(1 - \frac{2}{\pi}\arcsin(X_{ij}))$. It follows easily that $E[\widehat{x}_i\widehat{x}_j] = \frac{2}{\pi}\arcsin(X_{ij})$ and hence that

$$E[w(S)] = \frac{1}{4}\sum_{i,j} w_{ij}\left(1 - \frac{2}{\pi}\arcsin(X_{ij})\right). \tag{6}$$

Since $\arcsin(X_{ii}) = \pi/2$, for each $i = 1, \dots, n$, and $X_{ij} = \theta\widetilde{X}_{ij}$ when $i \ne j$, we conclude from (3) that the value in (6) is:

$$\ge \frac{1}{4}\sum_{i,j} w_{ij}\alpha(\theta)(1 - \widetilde{X}_{ij}) = \alpha(\theta)w^{\text{SDP}} \ge \alpha(\theta)w^*.$$

We can also derive that

$$\begin{aligned}
E[|S|(n - |S|)] &= \frac{1}{4}\sum_{i,j}\left(1 - \frac{2}{\pi}\arcsin(X_{ij})\right)\\
&= \frac{1}{4}\sum_{i\ne j}\left(1 - \frac{2}{\pi}\arcsin(\theta) + \frac{2}{\pi}\arcsin(\theta) - \frac{2}{\pi}\arcsin(\theta\widetilde{X}_{ij})\right)\\
&\ge \frac{1}{4}\sum_{i\ne j}(b(\theta) + c(\theta)(1 - \widetilde{X}_{ij})).
\end{aligned} \tag{7}$$

Now, noticing that $\sum_{i\ne j}\widetilde{X}_{ij} \le B^2 - n$, from (7) we derive that

$$\begin{aligned}
E[|S|(n - |S|)] &\ge \frac{1}{4}[(n^2 - n)b(\theta) + (n^2 - n)c(\theta) + c(\theta)(n - B^2)]\\
&= \frac{1}{4}[(n^2 - n)b(\theta) + (n^2 - B^2)c(\theta)]\\
&= \left[\left(1 - \frac{1}{n}\right)\frac{1}{1-\eta^2}b(\theta) + c(\theta)\right]\frac{n^2 - B^2}{4} = \beta(\theta, \eta)N. \quad \blacksquare
\end{aligned}$$

Lemma 3. *For every cut $(S, V \setminus S)$ generated by Algorithm 1 at line* 6.3 *we have that $\frac{w(S)}{w^*} \le 2$ and $\frac{|S|(n-|S|)}{N} \le \frac{1}{1-\eta^2}$.*

Proof. Let $S = \{i_1, \dots, i_s\}$. If $s \le (n + B)/2$ then by definition $w(S) \le w^*$. Otherwise apply function *rebalance*, described before Algorithm 1, to set S and let S' be the largest shore of the cut obtained . The weight $w(S)$ has decreased by at most $\frac{w(S)}{s}(s - \frac{n+B}{2})$. By definition $w(S') \le w^*$ but $w(S') \ge w(S) - \frac{w(S)}{s}(s - \frac{n+B}{2})$ and this implies $\frac{w(S)}{w^*} \le 2s/(n + B) \le 2$. The second inequality of the lemma follows immediately from the fact that $\frac{|S|(n-|S|)}{N} \le \frac{n^2}{4}\frac{4}{n^2-B^2}$. $\blacksquare$

Let us now fix a value $\gamma > 0$ and study the random variable

$$Z = \frac{w(S)}{w^*} + \gamma\frac{|S|(n - |S|)}{N}. \tag{8}$$

The two preceding lemmas imply that $Z \leq 2 + \frac{\gamma}{1-\eta^2}$ and that $E[Z] \geq \alpha(\theta) + \gamma\beta(\theta, \eta)$. Hence for small values of η (say $\eta < \sqrt{2/3} \leq 0.8165$) variable Z is bounded above, so that for any $\epsilon > 0$ and for constant k sufficiently large, Algorithm 1 generates a set S for which:

$$Z \geq [\alpha(\theta) + \gamma\beta(\theta, \eta)](1 - \epsilon). \tag{9}$$

We state the following:

Theorem 4. *For any $\gamma > 0$, if random variable Z satisfies (9), then for the corresponding set $\widetilde{S}$ computed by Algorithm 1 at line 6.4, we have:*

$$w(\widetilde{S}) \geq \min(g_1, g_2)w^* \tag{10}$$

with:

$$g_1 = 2\left(\sqrt{\gamma[\alpha(\theta) + \gamma\beta(\theta, \eta)]\frac{(1+\eta)(1-\epsilon)}{1-\eta}} - \frac{\gamma}{1-\eta}\right) \tag{11}$$

$$g_2 = [\alpha(\theta) + \gamma\beta(\theta, \eta)](1-\epsilon) - \frac{\gamma}{1-\eta^2}. \tag{12}$$

Proof. When the algorithm finds a set S satisfying (9) we let $\delta = |S|/n$ and $\lambda = w(S)/w^*$. From (8) and (9) it follows that:

$$\lambda \geq [\alpha(\theta) + \gamma\beta(\theta, \eta)](1-\epsilon) - 4\gamma\delta(1-\delta)\frac{1}{1-\eta^2}. \tag{13}$$

There are two possibilities for $\widetilde{S}$: either $\widetilde{S} = rebalance(S)$ or $\widetilde{S} = S$. In the first case it is easy to see that $w(\widetilde{S}) \geq \frac{n+B}{2}\frac{w(S)}{|S|} = \frac{1+\eta}{2\delta}\lambda w^*$, whereas in the second case we obviously have $w(\widetilde{S}) = \lambda w^*$. Hence $w(\widetilde{S}) \geq \min(\frac{1+\eta}{2\delta}\lambda, \lambda)w^*$ and, using inequality (13) for λ, we get:

$$w(\widetilde{S}) \geq \min(f_1, f_2)w^* \tag{14}$$

with

$$f_1 = [\alpha(\theta) + \gamma\beta(\theta, \eta)]\frac{(1+\eta)(1-\epsilon)}{2\delta} - 2\gamma\frac{1-\delta}{1-\eta} \tag{15}$$

$$f_2 = [\alpha(\theta) + \gamma\beta(\theta, \eta)](1-\epsilon) - 4\gamma\delta(1-\delta)\frac{1}{1-\eta^2}. \tag{16}$$

In order to simplify (15) and (16) and to remove the dependence on δ we study functions f_1 and f_2 for $\delta \geq 0$. Simple calculations show that function f_1 has a minimum at $\delta_1 = \sqrt{\frac{[\alpha(\theta)+\gamma\beta(\theta,\eta)](1-\eta^2)(1-\epsilon)}{4\gamma}}$, where it assumes the value $2(\sqrt{\gamma[\alpha(\theta) + \gamma\beta(\theta, \eta)]\frac{(1+\eta)(1-\epsilon)}{1-\eta}} - \frac{\gamma}{1-\eta})$ which is, by definition, the value of function g_1. Instead function f_2 has a minimum at $\delta_2 = 1/2$, where it takes on the value $[\alpha(\theta) + \gamma\beta(\theta, \eta)](1-\epsilon) - \frac{\gamma}{1-\eta^2}$ which again is, by definition, the value of function g_2. ■

Our next aim is to find the value of γ that maximizes $\min(g_1, g_2)$.

Again with straightforward calculations it can be seen that function g_1 is concave, is equal to zero for $\gamma = 0$ and for $\gamma_R = \frac{\alpha(\theta)(1-\eta^2)(1-\epsilon)}{1-\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}$ and has a maximum at

$$\gamma_M = \frac{\alpha(\theta)}{2\beta(\theta, \eta)}\left(\frac{1}{\sqrt{1-\beta(\theta, \eta)(1-\eta^2)(1-\epsilon)}} - 1\right).$$

Of course $\gamma_M \leq \gamma_R$.

The graph of function g_2, on the other hand, is a line that for $\gamma = 0$ has value $\alpha(\theta)(1-\epsilon)$ and then decreases until it intersects the γ axis, quite surprisingly, again in γ_R.

We have the following result:

Theorem 5. *For each η, $0 \le \eta < 1$, we have that $g_2 \le g_1$ iff $\gamma_L \le \gamma \le \gamma_R$, where we let $\gamma_L = \frac{\alpha(\theta)(1-\eta^2)(1-\epsilon)}{(1+2\eta)^2-\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}$ and $\gamma_R = \frac{\alpha(\theta)(1-\eta^2)(1-\epsilon)}{1-\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}$. Moreover $\gamma_L = \gamma_R$ iff $\eta = 0$.*

Proof. For simplicity of notation we let $\xi = [\alpha(\theta) + \gamma\beta(\theta, \eta)]$. Then by definition we have that $g_2 \le g_1$ iff:

$$\xi(1-\eta^2)(1-\epsilon) - \gamma \le 2(1-\eta^2)\left(\sqrt{\gamma\xi\frac{(1+\eta)(1-\epsilon)}{1-\eta}} - \frac{\gamma}{1-\eta}\right)$$

and hence iff

$$\xi(1-\eta^2)(1-\epsilon) - \gamma + 2\gamma(1+\eta) \le 2\sqrt{\xi\gamma(1-\eta^2)(1-\epsilon)(1+\eta)^2}. \tag{17}$$

Now if we let $x^2 = \xi\gamma(1-\eta^2)(1-\epsilon)$ then inequality (17) becomes

$$x^2 - 2x\gamma(1+\eta) + \gamma^2(1+2\eta) \le 0 \tag{18}$$

which has solutions for $\gamma \le x \le \gamma(1+2\eta)$. It can easily be seen that $\gamma \le \sqrt{\xi\gamma(1-\eta^2)(1-\epsilon)}$ iff $\gamma \le \frac{\alpha(\theta)(1-\eta^2)(1-\epsilon)}{1-\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}$ and that $\sqrt{\xi\gamma(1-\eta^2)(1-\epsilon)} \le \gamma(1+2\eta)$ iff $\frac{\alpha(\theta)(1-\eta^2)(1-\epsilon)}{(1+2\eta)^2-\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)} \le \gamma$. ■

Now from Theorems 5 and 4 we have

Corollary 6. *The value of γ that maximizes $\min(g_1, g_2)$ is γ_M if $\gamma_M \le \gamma_L$ otherwise it is γ_L. Moreover $\max_{\gamma>0}$ $\min(g_1, g_2)$ is equal to the value assumed by function g_1 in γ_M, if $\gamma_M \le \gamma_L$, otherwise to the value assumed in γ_L, if $\gamma_M > \gamma_L$.*

As a consequence of this corollary we can obtain a more explicit evaluation of (10), as expressed by the following lemma:

Lemma 7. *If random variable Z satisfies (9) for $\gamma \in \{\gamma_M, \gamma_L\}$, then for the corresponding set $\widetilde{S}$ computed by Algorithm 1 at line 6.4, we have*

$$w(\widetilde{S}) \ge \rho_1 w^* \quad \text{if } \eta^2 \le \frac{1-\beta(\theta,\eta)(1-\epsilon)}{4-\beta(\theta,\eta)(1-\epsilon)} \text{ and } \gamma = \gamma_M \tag{19}$$

$$w(\widetilde{S}) \ge \rho_2 w^* \quad \text{if } \eta^2 \ge \frac{1-\beta(\theta,\eta)(1-\epsilon)}{4-\beta(\theta,\eta)(1-\epsilon)} \text{ and } \gamma = \gamma_L \tag{20}$$

with

$$\rho_1 = \frac{\alpha(\theta)}{\beta(\theta,\eta)(1-\eta)}(1 - \sqrt{1-\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}) \tag{21}$$

$$\rho_2 = \frac{4\eta\alpha(\theta)(1+\eta)(1-\epsilon)}{(1+2\eta)^2 - \beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}. \tag{22}$$

Proof. We show first that $\eta^2 \le \frac{1-\beta(\theta,\eta)(1-\epsilon)}{4-\beta(\theta,\eta)(1-\epsilon)}$ iff $\gamma_M \le \gamma_L$, i.e. by definition, iff:

$$\frac{\alpha(\theta)}{2\beta(\theta,\eta)}\left(\frac{1}{\sqrt{1-\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}} - 1\right) \le \frac{\alpha(\theta)(1-\eta^2)(1-\epsilon)}{(1+2\eta)^2 - \beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}. \tag{23}$$

In fact if we let $a = (1+2\eta)^2$ and $b = \beta(\theta,\eta)(1-\eta^2)(1-\epsilon)$ then (23) becomes equivalent to $\frac{1}{\sqrt{1-b}} \le \frac{a+b}{a-b}$ which is true iff $2\sqrt{a} \ge a + b$. But this inequality is equivalent to $1 - 4\eta^2 \ge \beta(\theta,\eta)(1-\eta^2)(1-\epsilon)$ which is true iff $\eta^2 \le \frac{1-\beta(\theta,\eta)(1-\epsilon)}{4-\beta(\theta,\eta)(1-\epsilon)}$.

Now from Theorem 4 and Corollary 6 it remains only to prove that $\rho_1 = g_1(\gamma_M)$ and that $\rho_2 = g_1(\gamma_L)$. By definition we have that

$$g_1(\gamma_M) = 2\left(\sqrt{[\gamma_M\alpha(\theta) + \gamma_M^2\beta(\theta,\eta)]\frac{(1+\eta)(1-\epsilon)}{1-\eta}} - \frac{\gamma_M}{1-\eta}\right). \tag{24}$$

For simplicity of notation we let $\xi_M = \sqrt{1-\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}$ in the definition of γ_M . Then (24) becomes :

$$= 2\left(\sqrt{\left[\frac{\alpha^2(\theta)(1-\xi_M)}{2\beta(\theta,\eta)\xi_M} + \frac{\alpha^2(\theta)}{4\beta(\theta,\eta)}\left(\frac{1-\xi_M}{\xi_M}\right)^2\right]\frac{(1+\eta)(1-\epsilon)}{1-\eta}} - \frac{\gamma_M}{1-\eta}\right)$$

$$= 2\left(\sqrt{\frac{\alpha^2(\theta)}{4\beta(\theta,\eta)}\left(\frac{1}{\xi_M}-1\right)\left(\frac{1}{\xi_M}+1\right)\frac{(1+\eta)(1-\epsilon)}{1-\eta}} - \frac{\gamma_M}{1-\eta}\right)$$

$$= \frac{\alpha(\theta)(1+\eta)(1-\epsilon)}{\xi_M} - \frac{\alpha(\theta)(1-\xi_M)}{\xi_M\beta(\theta,\eta)(1-\eta)}$$

$$= \alpha(\theta)\frac{\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)-1+\xi_M}{\beta(\theta,\eta)(1-\eta)\xi_M} = \alpha(\theta)\frac{\xi_M-\xi_M^2}{\beta(\theta,\eta)(1-\eta)\xi_M} = \rho_1.$$

We also have that

$$g_1(\gamma_L) = 2\left(\sqrt{\gamma_L(\alpha(\theta)+\gamma_L\beta(\theta,\eta))\frac{(1+\eta)(1-\epsilon)}{1-\eta}} - \frac{\gamma_L}{1-\eta}\right) \tag{25}$$

and if we let $\xi_L = (1+2\eta)^2 - \beta(\theta,\eta)(1-\eta^2)(1-\epsilon)$, then (25) becomes:

$$= 2\left(\sqrt{\frac{\alpha(\theta)^2(1-\eta^2)(1-\epsilon)}{\xi_L}\left(1+\frac{\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}{\xi_L}\right)\frac{(1+\eta)(1-\epsilon)}{1-\eta}} - \frac{\gamma_L}{1-\eta}\right)$$

$$= 2\left(\sqrt{\frac{\alpha^2(\theta)(1+\eta)^2(1-\epsilon)^2}{\xi_L}\left(1+\frac{\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)}{\xi_L}\right)} - \frac{\alpha(\theta)(1+\eta)(1-\epsilon)}{\xi_L}\right)$$

$$= 2\left(\frac{\alpha(\theta)(1+\eta)(1-\epsilon)}{\xi_L}\sqrt{\xi_L+\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)} - \frac{\alpha(\theta)(1+\eta)(1-\epsilon)}{\xi_L}\right)$$

$$= \frac{4\alpha(\theta)(1+\eta)(1-\epsilon)\eta}{(1+2\eta)^2-\beta(\theta,\eta)(1-\eta^2)(1-\epsilon)} = \rho_2. \quad \blacksquare$$

Notice that functions ρ_1 and ρ_2, for each $\epsilon > 0$, depend on η and also on the value fixed in the algorithm for θ . For $\eta = 0$ and $\epsilon = 0$, ρ_1 coincide with the bound given in [12]; function ρ_2 has instead no counterpart in [12].

3.1. The appropriate choice of θ and k

In this subsection we discuss the choices that Algorithm 1 makes at line 2. Let us first consider the choice of θ. Since ρ_1 (ρ_2), for fixed $\epsilon > 0$, is a function of η and θ, then, for any given η, it is possible to compute, and to use in the algorithm, the value of θ that maximizes ρ_1 (ρ_2). The ratios reported in the first three groups of lines of Table 1 have been computed with this strategy, for n sufficiently large ($n \geq 10^4$) and $\epsilon = 0$. Figs. 1 and 2 show the behavior of the two functions; function ρ_1 has been plotted for $\theta \in [0.8..1]$ and $\eta \in [0..0.2]$, function ρ_2 for $\theta \in [0.8..1]$ and $\eta \in [0.2..0.8]$. In both cases we set $\epsilon = 0$. It is evident that the value of θ that maximizes ρ_1 (ρ_2) is a value in [0.88, 1).

For what concerns the choice of k we make the following considerations.

If we let $x = [\alpha(\theta) + \gamma\beta(\theta,\eta)](1-\epsilon)$ and $p = Pr\{Z < x\}$ then we have, for bounded values of η and hence of Z, that $E[Z] \leq px + (1-p)\max(Z)$. This inequality, together with the fact that from Lemmas 2 and 3 it follows that $Z \leq 2 + \frac{\gamma}{1-\eta^2}$ and $E[Z] \geq \alpha(\theta) + \gamma\beta(\theta,\eta)$, implies that:

$$p \leq \frac{2+\frac{\gamma}{1-\eta^2} - (\alpha(\theta)+\gamma\beta(\theta,\eta))}{2+\frac{\gamma}{1-\eta^2} - (\alpha(\theta)+\gamma\beta(\theta,\eta))(1-\epsilon)}. \tag{26}$$

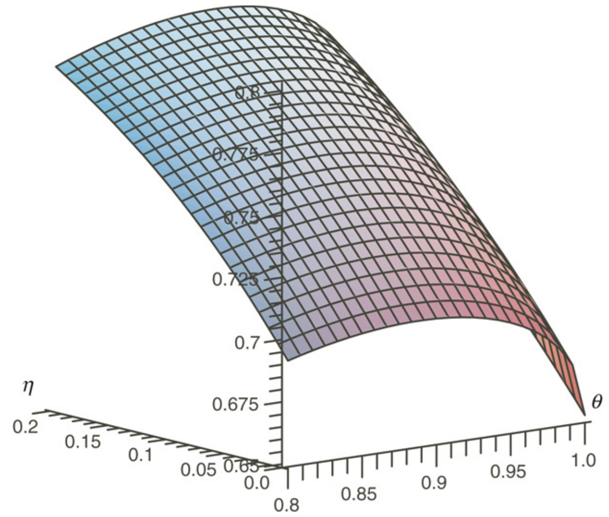

Fig. 1. Function ρ_1, with $\theta \in [0.8..1]$, $\eta \in [0..0.2]$.

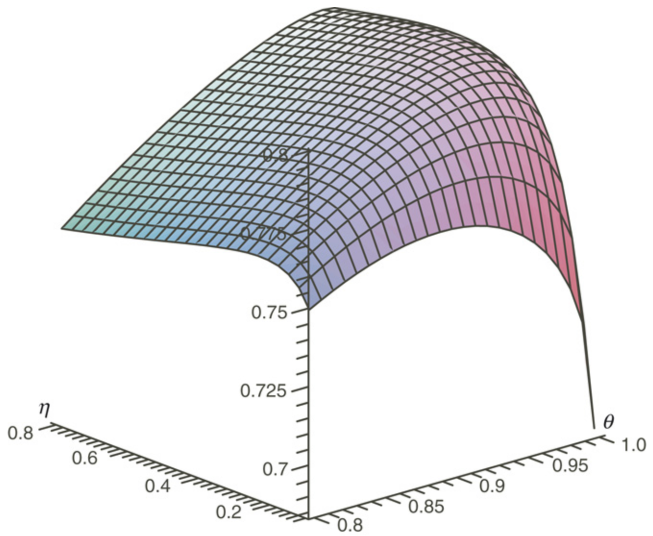

Fig. 2. Function ρ_2, with $\theta \in [0.8..1]$, $\eta \in [0.2..0.8]$.

It can be verified that $\left(\frac{2+\frac{\gamma}{1-\eta^2}-(\alpha(\theta)+\gamma\beta(\theta,\eta))}{2+\frac{\gamma}{1-\eta^2}-(\alpha(\theta)+\gamma\beta(\theta,\eta))(1-\epsilon)}\right)^k \le \epsilon$ if we choose $k = \frac{1}{\epsilon}\log\frac{1}{\epsilon}$ for small value of η and $\gamma = \gamma_M$ or $k = \frac{1}{\epsilon}\log\frac{1}{\epsilon^2}$ for large value of η and $\gamma = \gamma_L$.

In Algorithm 1 we therefore fix the values of θ and k according to these considerations. The overall performance of the algorithm then may finally be specified by the following theorem:

Theorem 8. *Let ϵ be a small positive constant. Then Algorithm 1 returns a solution S_R having $E[w(S_R)] \ge \rho w^*$ with:*

$$\rho = \rho_1(1-\epsilon) \quad \text{if } \eta^2 \le \frac{1-\beta(\theta,\eta)(1-\epsilon)}{4-\beta(\theta,\eta)(1-\epsilon)} \tag{27}$$

$$\rho = \rho_2(1-\epsilon) \quad \text{if } \eta^2 \ge \frac{1-\beta(\theta,\eta)(1-\epsilon)}{4-\beta(\theta,\eta)(1-\epsilon)} \tag{28}$$

where ρ_1 and ρ_2 are defined in (21) and (22).

Proof. Let $\gamma_1 = \gamma_M, \gamma_2 = \gamma_L$. For each $i \in \{1, 2\}$, let $x_i = [\alpha(\theta) + \gamma_i \beta(\theta, \eta)](1 - \epsilon)$, Z_i be the random variable defined in (8) with $\gamma = \gamma_i$, and Z_{M_i} be the random variable assuming the maximum value for Z_i in the loop of Algorithm 1, with $(\widetilde{S}_i, V \setminus \widetilde{S}_i))$ being the corresponding cut. It is straightforward that $E[w(S_R)] \geq E[w(\widetilde{S}_i)]$ and that $E[w(\widetilde{S}_i)] \geq \rho_i w^* Pr\{w(\widetilde{S}_i) \geq \rho_i w^*\}$, $i \in \{1, 2\}$. Now from Lemma 7 it follows that $Pr\{w(\widetilde{S}_1) \geq \rho_1 w^*\} \geq Pr\{Z_1 \geq x_1\}$ when $\eta^2 \leq \frac{1-\beta(\theta,\eta)(1-\epsilon)}{4-\beta(\theta,\eta)(1-\epsilon)}$ and also that $Pr\{w(\widetilde{S}_2) \geq \rho_2 w^*\} \geq Pr\{Z_2 \geq x_2\}$ when $\eta^2 \geq \frac{1-\beta(\theta,\eta)(1-\epsilon)}{4-\beta(\theta,\eta)(1-\epsilon)}$. From the considerations made on the appropriate choice of k, we derive that $Pr\{Z_{M_i} \geq x_i) \geq 1 - \epsilon$, for each $i \in \{1, 2\}$ and then the conclusion follows. ■

Hence the following proposition can be stated and easily proved.

Proposition 9. *For each of the values of η reported in the first three groups of lines of Table 1, the value of ρ guaranteed by Theorem 8 is larger than the value of r reported in the table, for n sufficiently large.*

Proof. The values reported in the table have been computed, by truncation at the third decimal, for each η, using the value of θ that maximizes ρ_1 (ρ_2) for n sufficiently large ($n \geq 10^4$) and $\epsilon = 0$. Since $\rho_i(1 - \epsilon)$ tends, for $\epsilon \to 0$, to a value greater or equal to the one reported in the table, the result follows. ■

4. The algorithm for large unbalance

In this section we use the following very simple algorithm, that uses function *rebalance*(S), introduced in Section 3. Here with w^M we indicate the weight of a maximum cut.

Algorithm 10.
- use the algorithm in [8] to obtain a cut $(S, V \setminus S)$ having $w(S) \geq 0.87856\, w^M$;
- denote by S the set of vertices with larger cardinality;
- if $|S| \leq (n + B)/2$ /* the cut is feasible for MaxCUT-LU */
  let $\widetilde{S} = S$ else let $\widetilde{S} = rebalance(S)$;
- return $\widetilde{S}$.

We have the following:

Theorem 11. *Algorithm 10 returns a set $\widetilde{S}$ having:*

$$w(\widetilde{S}) \geq 0.87856 \frac{1+\eta}{2} w^*. \tag{29}$$

Proof. If $\widetilde{S} = S$ the result follows easily since $w^M \geq w^*$ and $\eta \leq 1$. Otherwise, as in the proof of Lemma 3, the removal from S of the $|S| - (n + B)/2$ vertices that contribute less to the weight $w(S)$ of the cut reduces the weight by at most $\frac{w(S)}{|S|}(|s| - \frac{n+B}{2})$. Hence $w(\widetilde{S}) \geq w(S) - \frac{w(S)}{|S|}(|S| - \frac{n+B}{2}) = w(S)\frac{n+B}{2|S|}$. Since $w(S) \geq 0.87856\, w^M$ the result follows. ■

5. Conclusions

We have presented two polynomial time randomized approximation algorithms giving nontrivial performance guarantees for the MaxCUT-LU problem. The approximation ratios have been obtained by extending to this problem the methodology used in [12] for Max Bisection and by combining this technique with another one that uses the algorithm of [8]. Depending on the value of η, (27) or (28) or (29) give our best approximation result. In Table 1 we report the ratios r obtained for some values of the parameter η, subdivided in 4 groups. The values in the first group are given by (27), those in the last group are given by (29), the others by (28). Moreover the values reported in the first three groups have been computed, for n sufficiently large ($n \geq 10^4$) and $\epsilon = 0$, with truncation at the third decimal, using the values of θ, also reported in the table, that maximize the ratios.

For smaller n, e.g. $n = 10^3$, some of the approximation ratios in the table decrease only by 10^{-3}. Note that the breaking point between Algorithms 1 and 10 occurs for $\eta < \sqrt{2/3}$, as we have assumed.

In [7] an extensive computational experience with the algorithms we have analyzed is performed on several types of graphs. It turns out that the approximation ratios obtained on these graphs are always much better than the theoretical guarantees reported in Table 1.

Table 1
Best r for different values of η (and optimal choices of θ)

η	0.0000	0.0500	0.1000	0.1050	0.1065
θ	0.888	0.890	0.894	0.895	0.895
r	0.699	0.731	0.759	0.761	0.762
η	0.1065	0.2000	0.3333	0.4000	0.4500
θ	0.893	0.941	0.966	0.972	0.975
r	0.762	0.788	0.797	0.798	0.798
η	0.4930	0.5000	0.6000	0.7000	0.8000
θ	0.977	0.977	0.980	0.982	0.984
r	0.797	0.797	0.795	0.793	0.790
η	0.8000	0.8500	0.9000	0.9500	0.9999
r	0.790	0.8126	0.834	0.856	0.878

Acknowledgement

The second author was partially supported by MiUR (Ministero dell'Università e della Ricerca).

References

[1] A.A. Ageev, M.I. Sviridenko, Approximation algorithms for maximum coverage and max cut with given sizes of parts, in: IPCO 1999, in: LNCS, vol. 1610, Springer, Berlin, 1999, pp. 17–30.
[2] J. Akiyama, D. Avis, V. Chvatal, H. Era, Balancing signed graphs, Discrete Appl. Math. 3 (1981) 227–233.
[3] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, SIAM J. Optim. 5 (1995) 13–51.
[4] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, Proof verification and hardness of approximation problems, in: Proc. of the 33rd FOCS, 1992, pp. 14–23.
[5] U. Feige, M. Langberg, Approximation algorithms for maximization problems in graph partitioning, J. Algorithms 41 (2001) 174–211.
[6] A. Frieze, M. Jerrum, Improved approximation algorithms for MAX k-CUT and MAX BISECTION, Algorithmica 18 (1997) 67–81.
[7] G. Galbiati, S. Gualandi, F. Maffioli, Computational experince with a SDP-based algorithm for maximum cut with limited unbalance, in: Proc. of the 3rd International Network Optimization Conference, INOC 2007, 2007. File n. 24.
[8] M.X. Goemans, D.P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, J. ACM 42 (1995) 1115–1145.
[9] A. Hayrapetyan, D. Kempe, M. Pal, Z. Svitkina, Unbalanced graph cuts, in: ESA 2005, in: LNCS, vol. 3669, Springer, Berlin, 2005, pp. 191–202.
[10] S. Poljak, Z. Tuza, Maximum cuts and large bipartite subgraphs, in: W. Cook, L. Lovasz, P. Seymour (Eds.), Combinatorial Optimization, in: AMS-DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 20, American Mathematical Society, Providence, RI, 1995, pp. 181–244.
[11] V.V. Vazirani, Approximation Algorithms, Springer-Verlag, Berlin, 2001.
[12] Y. Ye, A. 699-approximation algorithm for max-bisection, Math. Program. Ser. A 90 (2001) 101–111.