

Approximation Bounds for Max-Cut Problem with Semidefinite Programming Relaxation

Dachuan Xu* and Shuzhong Zhang[†]

June 2005

Abstract

In this paper, we consider the max-cut problem as studied by Goemans and Williamson [8]. Since the problem is NP-hard in general, following Goemans and Williamson, we apply the approximation method based on the semidefinite programming (SDP) relaxation. In fact, the estimated worst-case performance ratio is dependent on the data of the problem with α being a uniform lower bound. In light of this new bound, we show that the actual worst-case performance ratio of the SDP relaxation approach (with the triangle inequalities added) is at least $\alpha + \delta_d$ if every weight is strictly positive, where $\delta_d > 0$ is a constant depending on the problem dimension and data. Alon, Sudakov and Zwick [2] show that the local analyses of the Goemans and Williamson algorithm and its extension by Zwick [25] are tight even if arbitrary collections of valid linear constraints are added to the SDP relaxation. Hence the improvement is in this sense best possible for the Goemans and Williamson type approach to the max-cut problem.

Key words: max-cut problem, semidefinite programming relaxation, approximation algorithm, performance ratio.

AMS subject classification: 49M27, 90C20, 90C22.

*College of Applied Sciences, Beijing University of Technology, 100 Pingleyuan, Chaoyang District, Beijing 100022, P.R. China. Email: xudc@bjut.edu.cn. Research supported in part by NSF of China (Grant Nos. 10401038 and 10171108) and Startup Grant for doctoral research of Beijing University of Technology. The manuscript was done while the author was a Postdoctoral Fellow at Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong.

[†]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: zhang@se.cuhk.edu.hk. Research supported by Hong Kong RGC Earmarked Grants CUHK4174/03E and CUHK4242/04E.

1. Introduction

One of the recent ground-breaking works in approximation algorithms for combinatorial optimization is the seminal paper by Goemans and Williamson [8] (1995), in which a semidefinite programming relaxation is proposed for the quadratic maximization formulation of the classical NP-hard max-cut problem, followed by a novel randomized rounding scheme. Goemans and Williamson managed to show that this procedure gives an approximate solution for the max-cut problem with the worst-case performance ratio guaranteed to be no less than $0.87856\dots$, denoted by α in this paper. Soon it turned out that the bound α is essentially tight for the Goemans-Williamson approach. In particular, Alon, Sudakov and Zwick [2] show that the local analysis of the Goemans and Williamson algorithm and its extension by Zwick [25] are tight even if arbitrary collections of valid linear constraints are added to the SDP relaxation (see also [1, 6, 12]). Up to date, no known polynomial-time approximation algorithm solves the max-cut problem with a guaranteed worst-case performance ratio substantially better than α . On the negative side, Håstad [10] shows that it is NP-hard to approximate the max-cut problem with a guaranteed worst case bound more than $16/17 + \epsilon = 0.94117\dots + \epsilon$, where $\epsilon > 0$ is any fixed positive constant.

The technique introduced by Goemans and Williamson has found applications in many other combinatorial optimization problems. For surveys on the subject, see [7, 9, 11, 14, 18, 21]. The power of the method has also been extended to more general non-convex quadratic programming problems; see [11, 15, 16, 17, 20, 22, 23, 25]. As a consequence of their rank-1 matrix decomposition technique, Sturm and Zhang [19] show that the SDP relaxation can be exact for some interesting quadratic optimization problems. Tseng [20], and Ye and Zhang [24] further extend the results for quadratic optimization problems. Zhang [25] studies the quality of the SDP relaxation method under various assumptions on the structure of the quadratic maximization problem. In particular, it turns out that the worst-case performance ratio α holds whenever a certain off-diagonal nonpositivity condition in the cost matrix is satisfied. Moreover, it is shown that the SDP relaxation is actually exact whenever the cost matrix is off-diagonally non-negative. The last result was extended by Kim and Kojima [13], where they show that a simpler second-order cone relaxation is in fact already exact.

For the max-cut problem itself, Goemans and Williamson [8] show that their rounding algorithm behaves even better if the percentage of edges in the cut is relatively high, or more precisely, if the cut contains more than 85% of the total weight of the edges. Using outward rotations, Zwick [26] obtains an approximation algorithm for max-cut and the performance guarantee is strictly greater than α , unless the maximum cut of the input graph contains about 84.458% of the total weight of the edges of the graph. Feige and Langberg [5] present a procedure called RPR² (Random Projection followed by Randomized Rounding) for rounding the solution of semidefinite programs and improve the tradeoff curve (presented by Zwick) for max-cut. In general, however, the approximation ratio remains α .

In this paper we mainly present two new results for the max-cut problem. In Section 2 we show that the SDP relaxation bound can be estimated using the SDP optimal solution itself, with α as a uniform lower bound of the new estimated bound. This analysis relies on the convexity of a related value function. Making further use of the convexity we prove in Section 3 that the true approximation ratio for the max-cut problem is $\alpha + \delta$ where $\delta > 0$ is in the order of $\Omega(\frac{1}{n^2L})$ with L denoting the ratio between the maximum weight and the minimum weight, if we add the triangle inequalities in the SDP relaxation and assume that every weight is strictly positive. We remark here that Feige, Karpinski and Langberg [4] show that if the degree of the graph is at most Δ , then the SDP relaxation approach yields an approximation bound $\alpha + \Omega(\frac{1}{\Delta^4})$, a result with an accent similar to ours.

Throughout, \mathfrak{R}^n denotes the space of n -dimensional Euclidean space; \mathcal{S}^n denotes the space of $n \times n$ real symmetric matrices; T denotes the transpose operation for a matrix. We denote e as the all-one vector. For any $x \in \mathfrak{R}^n$, we denote x_i as the i -th component of x . For $A \in \mathfrak{R}^{n \times n}$, A_{ij} denotes the (i, j) -th entry of A and $\text{diag}(A)$ denotes an n -dimensional vector formed by the diagonal elements of A . For $A, B \in \mathcal{S}^n$, we denote $\langle A, B \rangle = \sum_{i,j} A_{ij}B_{ij}$ as the inner-product between A and B . We denote $A \succeq B$ (respectively $A \succ B$) by the fact that $A - B$ is positive semidefinite (respectively positive definite). For a given one-dimensional function f , we denote $f(A)$ to be $[f(A_{ij})]_{n \times n}$.

2 An analysis of the max-cut problem

In this section, we consider the original max-cut problem. The goal is to get an estimation on the worst-case performance ratio in terms of the problem/solution data. In most cases, this ratio will turn out to be better than α , which plays the role as a uniform lower bound.

Let us start by introducing the max-cut problem. Consider an undirected graph $G(V, E)$ with $|V| = n$ and nonnegative (but not all zeros) weights $w_{ij} = w_{ji}$ on each edge $(i, j) \in E$. Set $w_{ij} = 0$, if $(i, j) \notin E$. The max-cut problem aims at finding the set of vertices S that maximizes the total weight of the edges in the *cut* (S, \bar{S}) ; that is, the total weight of the edges with one endpoint in S and the other endpoint in \bar{S} . The max-cut problem is a classical NP-hard problem. It is provable that it is even NP-hard to get an approximate algorithm with a worst-case performance ratio higher than $16/17$; see [10].

The Goemans and Williamson approach to the max-cut problem begins with a mathematical programming reformulation of the problem:

$$(MC) \quad \begin{aligned} & \text{maximize} && \frac{1}{2} \sum_{i,j} w_{ij}(1 - x_i x_j) \\ & \text{subject to} && x_i^2 = 1, \quad i = 1, \dots, n. \end{aligned}$$

We denote the cost matrix $Q = [q_{ij}]_{n \times n}$ as follows

$$q_{ij} = \begin{cases} -w_{ij}, & \text{if } i \neq j \\ \sum_{k=1}^n w_{ik}, & \text{if } i = j. \end{cases}$$

This Q is known as the Laplacian matrix of the graph. In the case that the weights are all nonnegative then $Q \succeq 0$.

The equivalent nonlinear semidefinite formulation is

$$(MCSP) \quad \begin{aligned} & \text{maximize} && \frac{2}{\pi} \langle Q, \arcsin(X) \rangle \\ & \text{subject to} && X \succeq 0, \text{diag}(X) = e, \end{aligned}$$

and the convex semidefinite programming relaxation for $(MCSP)$ is (see e.g. [3])

$$(MCR) \quad \begin{aligned} & \text{maximize} && \langle Q, X \rangle \\ & \text{subject to} && X \succeq 0, \text{diag}(X) = e. \end{aligned} \tag{2.1}$$

Let $v(MC)$ denote the optimal value of (MC) , $v(MCSP)$ denote the optimal value of $(MCSP)$, and $v(MCR)$ denote the optimal value of (MCR) . Let X^* be an optimal solution of (MCR) , i.e., $v(MCR) = \langle Q, X^* \rangle$. Obviously, $X = ee^T$ (the all-one matrix) is in any case a feasible solution for (MCR) . In case ee^T is indeed optimal for (MCR) , then it is also optimal for $(MCSP)$. Hence the relaxation is exact. For the interesting case we may assume that ee^T is not optimal for (MCR) , i.e.

$$\sum_{i,j} q_{ij} X_{ij}^* > \sum_{i,j} q_{ij}$$

or equivalently,

$$\sum_{i \neq j} q_{ij} X_{ij}^* > \sum_{i \neq j} q_{ij}.$$

Denote

$$W := \sum_{i \neq j} q_{ij} (1 - X_{ij}^*) < 0. \tag{2.2}$$

Noting $q_{ij}(1 - X_{ij}^*) \leq 0$ for all $i \neq j$, let us further define

$$\lambda_{ij} := \frac{q_{ij}(1 - X_{ij}^*)}{W} \geq 0$$

for all $i \neq j$. Naturally we have $\sum_{i \neq j} \lambda_{ij} = 1$, by definition of W . In other words, λ_{ij} 's can be viewed as coefficients of a convex combination.

We further note that $\text{diag}((X^*)^2) = e$ and $(X^*)^2 \succeq 0$ and so $(X^*)^2$ forms a feasible solution for (MCR) . Therefore, by optimality of X^* we have

$$\sum_{i \neq j} q_{ij} X_{ij}^* \geq \sum_{i \neq j} q_{ij} (X_{ij}^*)^2. \tag{2.3}$$

Now denote

$$A := \frac{\sum_{i \neq j} q_{ij}(1 - X_{ij}^*)X_{ij}^*}{W} = \sum_{i \neq j} \lambda_{ij} X_{ij}^*.$$

Thus,

$$-1 \leq A \leq 0 \tag{2.4}$$

where the second inequality is due to (2.3) and the first inequality is due to the fact that $X^* \succeq 0$ and so $|X_{ij}^*| \leq 1$ for all i, j .

Now let us consider a function that plays an important role in our analysis:

$$h(t) = \frac{1 - \frac{2}{\pi} \arcsin(t)}{1 - t}, \quad t \in [-1, 1). \tag{2.5}$$

Lemma 2.1 *The function $h(t)$ defined in (2.5) is strongly convex in the interval $[-1, 1)$, and it attains its minimum at $t = -0.6892 \dots$ with the minimum value equal to $\alpha = 0.87856 \dots$.*

Proof:

Simple computation shows that

$$h'(t) = -\frac{2}{\pi} \frac{1}{(1 - t^2)^{1/2}(1 - t)} + \frac{1 - \frac{2}{\pi} \arcsin t}{(1 - t)^2}$$

and

$$h''(t) = -\frac{2}{\pi} \frac{1}{(1 - t^2)^{3/2}} \frac{t}{1 - t} - \frac{4}{\pi} \frac{1}{(1 - t^2)^{1/2}(1 - t)^2} + 2 \frac{1 - \frac{2}{\pi} \arcsin t}{(1 - t)^3}.$$

One further computes that

$$h''(t) \geq 0.4472 =: \mu, \tag{2.6}$$

for all $t \in [-1, 1)$. The second part of the lemma is proved by numerically solving the root of $h'(t)$.

□

To visualize the picture, see Figure 1.

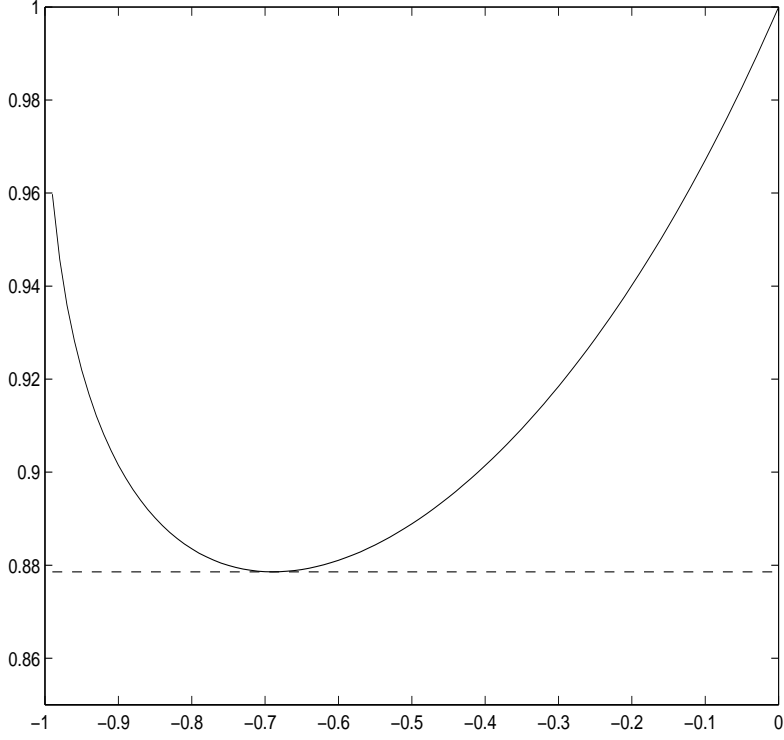


Figure 1: Function h

In Figure 1, the solid line corresponds to the function $h(t)$, and the dashed line corresponds to the minimum value α .

Theorem 2.2 $v(MC) = v(MCSP) \geq h(A)v(MCR)$.

Proof:

Let $\mathcal{I} = \{(i, j) \mid X_{ij}^* \neq 1\}$.

Obviously, $\lambda_{ij} = 0$ whenever $(i, j) \notin \mathcal{I}$.

By the convexity of h it follows that

$$\begin{aligned}
 h(A) &= h\left(\sum_{(i,j) \in \mathcal{I}} \lambda_{ij} X_{ij}^*\right) \\
 &\leq \sum_{(i,j) \in \mathcal{I}} \lambda_{ij} h(X_{ij}^*) \\
 &= \sum_{(i,j) \in \mathcal{I}} \frac{q_{ij}(1 - X_{ij}^*)}{W} \times \frac{1 - \frac{2}{\pi} \arcsin(X_{ij}^*)}{1 - X_{ij}^*} \\
 &= \frac{1}{W} \sum_{(i,j) \in \mathcal{I}} q_{ij} \left(1 - \frac{2}{\pi} \arcsin(X_{ij}^*)\right)
 \end{aligned}$$

$$= \frac{1}{W} \sum_{i \neq j} q_{ij} \left(1 - \frac{2}{\pi} \arcsin(X_{ij}^*)\right).$$

Noting (2.2), the above inequality can be rewritten as

$$\sum_{i \neq j} q_{ij} \left(\frac{2}{\pi} \arcsin(X_{ij}^*)\right) \geq \sum_{i \neq j} q_{ij} (1 - h(A) + h(A)X_{ij}^*). \quad (2.7)$$

Since X^* forms a feasible solution to $(MCSP)$, therefore we have

$$\begin{aligned} v(MCSP) &\geq \sum_{i,j} q_{ij} \left(\frac{2}{\pi} \arcsin(X^*)\right) \\ &= \sum_{i \neq j} q_{ij} \left(\frac{2}{\pi} \arcsin(X_{ij}^*)\right) + \sum_{i=1}^n q_{ii} \\ &\geq \sum_{i \neq j} q_{ij} (1 - h(A) + h(A)X_{ij}^*) + \sum_{i=1}^n q_{ii} \quad (\text{by (2.7)}) \\ &= (1 - h(A)) \sum_{i,j} q_{ij} + h(A) \sum_{i,j} q_{ij} X_{ij}^* \\ &= (1 - h(A)) e^T Q e + h(A) v(MCR) \\ &= h(A) v(MCR). \end{aligned}$$

This concludes the proof. □

Combining Theorem 2.2 and Lemma 2.1, we have

Corollary 2.3 $v(MC) = v(MCSP) \geq \alpha v(MCR)$.

3 Further analysis of the max-cut problem

In this section, we shall revisit the original max-cut problem in light of the strong convexity property of $h(\cdot)$ (Lemma 2.1).

Let us re-denote

$$\lambda_{ij} = w_{ij} (1 - X_{ij}^*) \geq 0$$

for all $i \neq j$, and

$$\lambda = \sum_{i,j} \lambda_{ij} = \langle Q, X^* \rangle > 0.$$

Furthermore, denote

$$A = \sum_{i \neq j} \frac{\lambda_{ij}}{\lambda} X_{ij}^*. \quad (3.8)$$

As in Section 2 (see (2.4)), we have $-1 \leq A \leq 0$.

Also along the same line as in Section 2, denote $\mathcal{I} = \{(i, j) \mid X_{ij}^* \neq 1, i, j = 1, 2, \dots, n\}$.

By the strong convexity of h as established in Lemma 2.1 we have

$$h(X_{ij}^*) \geq h(A) + h'(A)(X_{ij}^* - A) + \frac{\mu}{2}(X_{ij}^* - A)^2, \quad (3.9)$$

for all i, j , where $\mu = 0.4472$ (cf. (2.6)) is the lower bound on the second order derivative of h in the domain $[-1, 1]$.

Multiplying $\frac{\lambda_{ij}}{\lambda}$ to (3.9) and sum up for all $(i, j) \in \mathcal{I}$, we have

$$\sum_{(i,j) \in \mathcal{I}} \frac{\lambda_{ij}}{\lambda} h(X_{ij}^*) \geq h(A) + \frac{\mu}{2} \sum_{(i,j) \in \mathcal{I}} \frac{\lambda_{ij}}{\lambda} (X_{ij}^* - A)^2 = h(A) + \frac{\mu}{2} \sum_{i,j} \frac{\lambda_{ij}}{\lambda} (X_{ij}^* - A)^2. \quad (3.10)$$

Now the left side of the above inequality is

$$\begin{aligned} & \sum_{(i,j) \in \mathcal{I}} \frac{\lambda_{ij}}{\lambda} h(X_{ij}^*) \\ &= \sum_{i,j} w_{ij} \left(1 - \frac{2}{\pi} \arcsin X_{ij}^*\right) \langle Q, X^* \rangle \\ &= \langle Q, \frac{2}{\pi} \arcsin(X_{ij}^*) \rangle \langle Q, X^* \rangle. \end{aligned}$$

In combination of (3.10) we have

$$\begin{aligned} v(MC) &= v(MCSP) \\ &\geq \langle Q, \frac{2}{\pi} \arcsin(X_{ij}^*) \rangle \\ &\geq h(A) \langle Q, X^* \rangle + \frac{\mu}{2} \sum_{i,j} \lambda_{ij} (X_{ij}^* - A)^2 \\ &= h(A) v(MCR) + \frac{\mu}{2} \sum_{i,j} w_{ij} (1 - X_{ij}^*) (X_{ij}^* - A)^2. \end{aligned} \quad (3.11)$$

Goemans and Williamson [8] use the random hyperplane rounding technique to convert a solution of (MCR) to a feasible cut in the graph. In that sense, any ratio between $v(MC)$ and $v(MCR)$ will serve as a worst-case performance bound.

Whenever $A \geq -0.5$ or $A \leq -0.8$, we shall have $h(A) \geq 0.8835$, hence a bound better than α based on (3.11). Let us now consider the case

$$A \in [-0.8, -0.5]. \quad (3.12)$$

In the subsequent discussion, we assume that the graph is complete and that

$$w_{ij} > 0 \quad \text{for all } i \neq j. \quad (3.13)$$

Let

$$I_\varepsilon = \{(i, j) \mid |X_{ij}^* - A| \leq \varepsilon\}$$

and

$$J_\varepsilon = \{(i, j) \mid X_{ij}^* \geq 1 - \varepsilon, i \neq j\},$$

with $\varepsilon = 0.03$.

Let us consider an enhanced (but still polynomially sized) version of the SDP relaxation for the max-cut problem. That is, we add the following so-called triangle inequalities into the constraint set of (2.1):

$$X_{ij} + X_{jk} + X_{ki} \geq -1, \quad (3.14)$$

$$-X_{ij} - X_{jk} + X_{ki} \geq -1, \quad (3.15)$$

$$-X_{ij} + X_{jk} - X_{ki} \geq -1, \quad (3.16)$$

$$X_{ij} - X_{jk} - X_{ki} \geq -1, \quad (3.17)$$

for all possible triplets $1 \leq i < j < k \leq n$.

Lemma 3.1 *Let X^* be an optimal solution to the problem (MCR) with (3.14), (3.15), (3.16) and (3.17) added to the constraint set. Then, $I_\varepsilon \cap J_\varepsilon = \emptyset$ and $\{(i, j) \mid 1 \leq i \neq j \leq n\} \setminus (I_\varepsilon \cup J_\varepsilon) \neq \emptyset$.*

Proof:

That $I_\varepsilon \cap J_\varepsilon = \emptyset$ is immediately clear by the fact that $A \in [-0.8, -0.5]$ and $\varepsilon = 0.03$.

We wish to prove the second assertion by contradiction. Namely, for the sake of contradiction, we assume that $I_\varepsilon \cup J_\varepsilon = \{(i, j) \mid 1 \leq i \neq j \leq n\}$.

Let $m = \text{rank}(X^*)$ and let $X^* = [v_1, \dots, v_n]^T [v_1, \dots, v_n]$. Therefore, $v_i \in \Re^m$, $\|v_i\| = 1$, $i = 1, \dots, n$, and $X_{ij}^* = v_i^T v_j$ for all $1 \leq i, j \leq n$. Let

$$\mathcal{V}_1 = \{v_i \mid v_i^T v_1 \geq 1 - \varepsilon, i = 1, \dots, n\}$$

and

$$\mathcal{V}_2 = \{v_j \mid |v_j^T v_1 - A| \leq \varepsilon, j = 1, \dots, n\}.$$

Obviously, $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ and $\mathcal{V}_1 \cup \mathcal{V}_2 = \{v_1, \dots, v_n\}$ by the contradiction assumption.

We shall first prove the following three facts regarding \mathcal{V}_1 and \mathcal{V}_2 .

Fact 1. For any $v_i, v_j \in \mathcal{V}_1$, it must follow that $v_i^T v_j \geq 1 - \varepsilon$.

This is because if the above statement is incorrect then we must have $|v_i^T v_j - A| \leq \varepsilon$ instead (note the contradiction assumption). However,

$$v_i^T v_j = X_{ij}^* \geq -1 + X_{i1}^* + X_{1j}^* \geq -1 + 2 \times (1 - \varepsilon) = 1 - 2\varepsilon$$

due to (3.15). This is impossible as

$$|v_i^T v_j - A| \geq 1 - 2\varepsilon - A > \varepsilon.$$

Fact 2. For any $v_i, v_j \in \mathcal{V}_2$, we have $v_i^T v_j \geq 1 - \varepsilon$.

To see this, we again prove by contradiction. Suppose there are two vectors $v_i, v_j \in \mathcal{V}_2$ such that $|v_i^T v_j - A| \leq \varepsilon$. Then the triplet $\{v_1, v_i, v_j\}$ would violate the constraint (3.14).

Fact 3. For any $v_i \in \mathcal{V}_1, v_j \in \mathcal{V}_2$, we have $|v_i^T v_j - A| \leq \varepsilon$.

Once again, this can be shown by contradiction. Suppose that the above is not true. Then we would have $v_i \in \mathcal{V}_1, v_j \in \mathcal{V}_2$, and $v_i^T v_j - 1 \geq -\varepsilon$. Then, the triplet $\{v_1, v_i, v_j\}$ would contradict the constraints (3.15)-(3.17).

Next we shall perform an operation on the vectors v_1, \dots, v_n described by the following steps. First of all, let

$$v_k^T v_l = \max\{v_i^T v_j \mid v_i \in \mathcal{V}_1 \text{ and } v_j \in \mathcal{V}_2\}.$$

Using **Fact 3** we have $-0.83 \leq v_k^T v_l \leq -0.47$.

By **Fact 1** and **Fact 2** we know that all the vectors in \mathcal{V}_1 reside in the second order cone

$$\text{SOC}(v_k) := \{d \mid \angle(d, v_k) \leq \arccos(1 - \varepsilon)\},$$

and all the vectors in \mathcal{V}_2 reside in the second order cone

$$\text{SOC}(v_l) := \{d \mid \angle(d, v_l) \leq \arccos(1 - \varepsilon)\}.$$

By **Fact 3**, we know that $\text{SOC}(v_k) \cap \text{SOC}(v_l) = \{0\}$.

Let $h = (v_k + v_l)/2$. Then,

$$\angle(v_k, h) = \angle(v_l, h) \leq \arccos(A - \varepsilon)/2 \leq \arccos(-0.83)/2 \leq 1.28.$$

Moreover, for any $d \in \text{SOC}(v_k)$ we have

$$\angle(d, h) \leq \angle(d, v_k) + \angle(v_k, h) \leq \arccos(1 - \varepsilon) + \arccos(-0.83)/2 \leq 1.53 < \pi/2.$$

By symmetry, we also have $\angle(d, h) \leq 1.53 < \pi/2$ for all $d \in \text{SOC}(v_l)$.

This implies that there exists a solid and pointed convex cone containing the sum of the two second order cones, namely there is a solid, pointed convex cone \mathcal{K} such that $\mathcal{K} \supseteq \text{SOC}(v_k) + \text{SOC}(v_l)$. In particular, we may, for instance, let

$$\mathcal{K} := \text{SOC}(h) := \{d \mid \angle(d, h) \leq 1.53 < \pi/2\}.$$

Therefore $\text{int } \mathcal{K}^* \neq \emptyset$.

Before proceeding let us consider a geometric property. Suppose that there is a given direction ι and a vector x . Suppose $\angle(\iota, x) > \pi/2$, i.e. $\iota^T x < 0$. Consider a movement from x to $x + \Delta x$, where Δx is a small displacement. Then,

$$\begin{aligned} \cos \angle(\iota, x + \Delta x) &= \frac{\langle \iota, x + \Delta x \rangle}{\|\iota\| \cdot \|x + \Delta x\|} \\ &= \frac{\langle \iota, x \rangle + \langle \iota, \Delta x \rangle}{\|\iota\| \cdot \|x\|} \left[1 - \frac{x^T \Delta x}{\|x\|^2} + o(\|\Delta x\|) \right] \\ &= \cos \angle(\iota, x) + \left\langle \frac{\iota}{\|\iota\|} - \cos \angle(\iota, x) \frac{x}{\|x\|}, \frac{\Delta x}{\|x\|} \right\rangle + o(\|\Delta x\|). \end{aligned}$$

Therefore the angle between ι and $x + \Delta x$ actually increases from the angle between ι and x for small displacement Δx if and only if

$$\left\langle \frac{\iota}{\|\iota\|} - \cos \angle(\iota, x) \frac{x}{\|x\|}, \Delta x \right\rangle < 0. \quad (3.18)$$

Now let us take any $\Delta x \in -\text{int } \mathcal{K}^*$. So we have

$$\langle \theta u + \xi v, \Delta x \rangle < 0 \quad (3.19)$$

for all $\theta \geq 0$, $\xi \geq 0$, $u \in \text{SOC}(v_k)$ and $v \in \text{SOC}(v_l)$ with $0 \neq \theta u + \xi v \in \mathcal{K}$. This implies that if we shift the whole cone $\text{SOC}(v_l)$ along the direction Δx , then for *each* fixed pair of vectors in $\text{SOC}(v_k)$ and $\text{SOC}(v_l)$, the (obtuse) angle between them actually increases, due to condition (3.18) and the actual choice of the moving direction (3.19), which satisfies condition (3.18).

Let us now perform such a shifting operation. First, we keep $\text{SOC}(v_k)$ unchanged. In this case, all the vectors in \mathcal{V}_1 remain unchanged. Then, we shift the entire cone $\text{SOC}(v_l)$ slightly along the direction Δx . In this case, all the vectors in \mathcal{V}_2 are also shifted along the same direction Δx , while their relative positions remain unchanged.

It is clear that if we do so, then

$$\begin{aligned} \angle(v_i, v_j) &\text{ is invariant if } v_i, v_j \in \mathcal{V}_1, \\ \angle(v_i, v_j) &\text{ is invariant if } v_i, v_j \in \mathcal{V}_2, \\ \angle(v_i, v_j) &\text{ increases if } v_i \in \mathcal{V}_1 \text{ and } v_j \in \mathcal{V}_2, \end{aligned}$$

where the last assertion is due to (3.18), (3.19) and the above observations.

Let $\hat{X}_{ij} = v_i^T v_j$ where v_i 's, v_j 's are the new vectors after the shifting operation. Since the shifting does not change the norm, so $\hat{X} \succeq 0$ and $\text{diag}(\hat{X}) = e$, implying that \hat{X} remains feasible for (MCR). Moreover, \hat{X}_{ij} remains unchanged for either $i, j \in \mathcal{V}_1$ or $i, j \in \mathcal{V}_2$. However, \hat{X}_{ij} decreases for $i \in \mathcal{V}_1$ and $j \in \mathcal{V}_2$.

It can be easily verified that the corresponding objective value for \hat{X} is strictly greater than $v(R)$. Moreover, \hat{X} also satisfies the triangle inequalities due to the three facts established earlier. The contradiction proves the lemma. \square

Let

$$L = \frac{\max_{i \neq j} w_{ij}}{\min_{i \neq j} w_{ij}} > 0. \quad (3.20)$$

Now, we apply Lemma 3.1 and use (3.11) to obtain

$$\begin{aligned} v(MC) &\geq h(A)v(MCR) + \frac{\mu}{2} \sum_{i,j} w_{ij}(1 - X_{ij}^*)(X_{ij}^* - A)^2 \\ &\geq h(A)v(MCR) + \frac{\mu}{2} \sum_{(i,j) \in \{(i,j) | 1 \leq i \neq j \leq n\} \setminus \{I_\varepsilon \cup J_\varepsilon\}} w_{ij}(1 - X_{ij}^*)(X_{ij}^* - A)^2 \\ &\geq h(A)v(MCR) + \frac{\mu\varepsilon^3}{2} \min_{i \neq j} w_{ij} \\ &= h(A)v(MCR) + \frac{\mu\varepsilon^3}{2L} \max_{i \neq j} w_{ij} \\ &\geq h(A)v(MCR) + \frac{\mu\varepsilon^3}{4Ln^2} \sum_{i,j} w_{ij}(1 - X_{ij}^*) \\ &= (h(A) + \frac{\mu\varepsilon^3}{4Ln^2})v(MCR). \end{aligned}$$

Finally we conclude the following theorem:

Theorem 3.2 *Consider the Goemans and Williamson algorithm for the max-cut problem where all the weights are positive. Suppose that the triangle inequalities (3.14), (3.15), (3.16), (3.17) are added in the SDP relaxation. Then, the worst-case performance ratio is bounded below by*

$$\alpha(A) = \begin{cases} h(A) (\geq 0.8835), & \text{if } A \in [-1, -0.8] \cup [-0.5, 0], \\ h(A) + \frac{\mu\varepsilon^3}{4Ln^2} (\geq \alpha + \frac{\mu\varepsilon^3}{4Ln^2}), & \text{otherwise,} \end{cases}$$

where $\mu > 0$ and $\varepsilon > 0$ are some universal constants, and A is as defined in (3.8) and L is as defined in (3.20).

4 Discussions

In Section 3, we showed that the worst-case performance ratio for the Goemans and Williamson method is at least $\alpha + \Omega(\frac{1}{n^2L})$ if we add the triangle inequalities and assume that $w_{ij} > 0$ for all $i \neq j$, where L is the ratio between the maximum weight and the minimum weight and n is the number of nodes. One may wonder whether we could still prove similar result if we relax the assumption $w_{ij} > 0$ to $w_{ij} \geq 0$. Unfortunately, this is not possible. To put the picture into perspective, we draw our attention to a result of Alon, Sudakov and Zwick [2]. In that paper (Theorem 3.1), these authors show that for any rational number $-1 < a < 0$, there exist infinitely many values of n for which there is a graph $G = (V, E)$, $V = \{1, \dots, n\}$ and a positive semidefinite matrix U such that either $U_{ij} = a$ or $U_{ij} = 1$ for all $\{i, j\} \in E$, and the size of maximum cut in G is equal to

$$\max_{\text{diag}(X)=e, X \succeq 0} \sum_{\{i,j\} \in E} \frac{1 - X_{ij}}{2} = \sum_{\{i,j\} \in E} \frac{1 - U_{ij}}{2}.$$

Even if valid linear constraints (such as the triangle inequalities) are added to the SDP relaxation, the above result remains true. By the discussion in the previous section, it follows that the performance ratio $h(a)$ is tight. This means that the worst case performance ratio of the Goemans-Williamson type method is exactly $h(A)$. Therefore, the condition on the positive weights cannot be removed, and in this sense, the result in the previous section is best possible within the framework of the Goemans-Williamson approach to the max-cut problem.

References

- [1] N. Alon and B. Sudakov, Bipartite subgraphs and the smallest eigenvalue, *Combinatorics, Probability and Computing*, 2000 (9): 1-12.
- [2] N. Alon, B. Sudakov and U. Zwick, Constructing worst case instances for semidefinite programming based approximation algorithms, *SIAM Journal on Discrete Mathematics*, 2002 (15): 58-72.
- [3] D. Bertsimas and Y. Ye, Semidefinite relaxation, multivariate normal distributions, and other statistics, In: Du, D.-Z., Pardalos, P.M., eds., *Handbook of Combinatorial Optimization 3*, pp. 1-19. Kluwer Academic Publishers.
- [4] U. Feige, M. Karpinski and M. Langberg, Improved approximation of max-cut on graphs of bounded degree, *Journal of Algorithms*, 2002 (43): 201-219.

- [5] U. Feige and M. Langberg, The RPR² rounding technique for semidefinite programs, In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming, Crete, Greece*, 2001, 213-224.
- [6] U. Feige and G. Schechtman, On the optimality of the random hyperplane rounding technique for MAX CUT, to appear in *Random Structures and Algorithms*.
- [7] M.X. Goemans, Semidefinite programming in combinatorial optimization, *Mathematical Programming*, 1997 (79): 143-161.
- [8] M.X. Goemans and D.P. Williamson, Improved approximation algorithms for Maximum Cut and Satisfiability problems using semidefinite programming, *Journal of the ACM*, 1995 (42): 1115-1145.
- [9] Q. Han, Y. Ye and J. Zhang, An improved rounding method and semidefinite programming relaxation for graph partition, *Mathematical Programming*, 2002 (92): 509-535.
- [10] J. Håstad, Some optimal inapproximability results, In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 1997, 1-10.
- [11] C. Helmberg, Semidefinite programming for combinatorial optimization, *ZIB-Report 00-34*, 2000.
- [12] H. Karloff, How good is the Goemans-Williamson MAX CUT algorithm? *SIAM Journal on Computing*, 1999 (29): 336-350.
- [13] S. Kim and M. Kojima, Exact solutions of some nonconvex quartic optimization problems via SDP and SOCP relaxations, *Working Paper*, Department of Mathematical and Information Sciences, The Tokyo Institute of Technology, 2001.
- [14] L. Lovász, Semidefinite programs and combinatorial optimization, *Technical Report*, Microsoft Research, Redmond, WA 98052, 2000.
- [15] A. Nemirovski, C. Roos and T. Terlaky, On the maximization of quadratic form over intersection of ellipsoids with common center, *Mathematical Programming*, 1999 (86): 463-474.
- [16] Yu. Nesterov, Semidefinite relaxation and nonconvex quadratic optimization, *Optimization Methods and Software*, 1998 (9): 141-160.
- [17] Yu. Nesterov, Global quadratic optimization via conic relaxation, *CORE Discussion Paper*, Louvain-la-Neuve, Belgium, 1998.
- [18] F. Rendl, Semidefinite programming and combinatorial optimization, *Tech. Report*, Universität Klagenfurt, Institut für Mathematik, Austria, 1998.

- [19] J.F. Sturm and S. Zhang, On cones of nonnegative quadratic functions, *Mathematics of Operations Research*, 2003 (28): 246-267.
- [20] P. Tseng, Further Results on approximating nonconvex quadratic optimization by semidefinite programming relaxation, *SIAM Journal on Optimization*, 2003 (14): 268-283.
- [21] D. Xu, Y. Ye and J. Zhang, Approximate the 2-catalog segmentation problem using semidefinite programming relaxation, *Optimization Method and Software*, 2003 (18): 705-719.
- [22] Y. Ye, Approximating quadratic programming with bound and quadratic constraints, *Mathematical Programming*, 1999 (84): 219-226.
- [23] Y. Ye, Approximating global quadratic optimization with convex quadratic constraints, *Journal of Global Optimization*, 1999 (15): 1-17.
- [24] Y. Ye and S. Zhang, New results on quadratic minimization, *SIAM Journal on Optimization*, 2003 (14): 245-267.
- [25] S. Zhang, Quadratic maximization and semidefinite relaxation, *Mathematical Programming*, 2000 (87): 453-465.
- [26] U. Zwick, Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems, In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, 1999, 679-687.