An Improved Rounding Method and Semidefinite Programming Relaxation for Graph Partition *

Qiaoming Han †
School of Mathematics and Computer Science
Nanjing Normal University
Nanjing, 210097, P.R.China

Yinyu Ye
Department of Management Sciences
Henry B. Tippie College of Business
The University of Iowa
Iowa City, IA 52242, USA

Jiawei Zhang
Department of Management Sciences
Henry B. Tippie College of Business
The University of Iowa
Iowa City, IA 52242, USA

Revised March 9, 2000

*Research supported in part by NSF grants DMI-9908077 and DMS-9703490.

†The author is currently visiting Computational Optimization Laboratory, Department of Management Sciences, The University of Iowa, Iowa City, IA 52242, USA.

Abstract

Given an undirected graph $G = (V, E)$ with $|V| = n$ and an integer k between 0 and n, the maximization graph partition (MAX-GP) problem is to determine a subset $S \subset V$ of k nodes such that an objective function $w(S)$ is maximized. The MAX-GP problem can be formulated as a binary quadratic program and it is NP-hard. Semidefinite programming (SDP) relaxations of such quadratic programs have been used to design approximation algorithms with guaranteed performance ratios for various MAX-GP problems. Based on several early results, we present a new rounding method using an SDP relaxation, and establish improved approximation ratios for several MAX-GP problems, including Dense-Subgraph, Max-Cut, and Max-Not-Cut.

Key words. Dense-k-Subgraph, Max-Cut, Max-Not-Cut, polynomial approximation algorithm, performance ratio, semidefinite programming.

1 Introduction

Given an undirected graph $G = (V, E)$ with $|V| = n$, non-negative weights w_{ij} on edges $(i, j) \in E$, and an integer k $(0 < k < n)$, the maximization graph partition (MAX-GP) problem is to determine a subset $S \subset V$ of k nodes such that an objective function $w(S)$ is maximized. Some examples of MAX-GP are: *Dense-k-Subgraph (DSP)*, where the total edge weights of the subgraph induced by S is maximized; *Max-Cut with size k (MC)*, where the total edge weights of the edges crossing between S and $V \setminus S$ is maximized; *Max-Not-Cut with size k (MNC)*, where the total edge weights of the non-crossing edges between S and $V \setminus S$ is maximized; *Max-Vertex-Cover with size k (MVC)*, where the total edge weights of the edges covered by S is maximized; etc. We consider DSP, MC and MNC in this paper, and discuss MVC in another paper.

Since these (MAX-GP) problems are NP-hard (e.g., see [7] for DSP, [1] for MC, and [17] for MNC), one should not expect to find polynomial time algorithms for computing their optimal solutions. Therefore, we are interested in how close to optimality one can approach in polynomial time. A (randomized) polynomial time approximation algorithm for a maximization problem has a performance guarantee or worst case ratio $0 < r \leq 1$, if it outputs a feasible solution whose (expected) value is at least r times the maximal value for all instance of the problem. Such an algorithm is often called (randomized) r-approximation algorithm. A key step in designing a good approximation algorithm for such a maximization problem is to establish a good upper bound on the maximal objective value. Linear programming (LP) and semidefinite programming (SDP) have been frequently used to provide such upper bounds for many NP-hard problems.

There are several approximation algorithms for DSP. Kortsarz and Peleg [11] devised an approximation algorithm which has a performance guarantee $O(n^{-0.3885})$. Feige, Kortsarz and Peleg [8] improved it to $O(n^{-1/3+\epsilon})$, for some $\epsilon > 0$. The other approximation algorithms have performance guarantees which are the function of k/n, e.g., a greedy heuristic by Asahiro et.al. [3], and SDP relaxation based algorithms developed by Feige and Langberg [6] and Feige and Seltser [7], and Srivastav and Wolf [14]. The previously best performance guarantee, k/n for general k and $k/n+\epsilon_k$ for $k \sim n/2$, was obtained by Feige and Langberg [6]. Moreover, for the case $k = n/2$, Ye and Zhang [17], using a new SDP relaxation, obtained an improved 0.586 performance guarantee from 0.517 of Feige and Langberg [6].

For approximating the MC problem with size k, both LP and SDP based approximation algorithms have a performance ratio $1/2$ for all k, see Ageev and Sviridenko [1] and Feige and Langberg [6]. For $k = n/2$, i.e., the Max-Bisection, Frieze and Jerrum

[9] obtained a 0.651-approximation algorithm (the same bound was also obtained by Anderson et.al [2] in their paper for max-p-section). Subsequently, this ratio has been improved to 0.699 by Ye [15]. Both of their approximation algorithms are based on SDP relaxations.

For approximating the MNC problem with size k, the LP-based approximation algorithm has a performance ratio $1 - \frac{2k(n-k)}{n(n-1)}$ for all k, and the SDP-based algorithm has a ratio $.5 + \epsilon_k$ for $k \sim n/2$, see Feige and Langberg [6]. Again, Ye and Zhang [17] obtained a 0.602-approximation algorithm for MVC when $k = n/2$, comparing to .541 of Feige and Langberg [6].

Our main contribution of this paper, motivated from the results of Ye [15] and Ye and Zhang [17], is to present a new method to round an optimal solution of the SDP relaxation of the MAX-GP problem for general k. This rounding technique is related to the well-known rounding method introduced by Goemans and Williamson [10] in their seminal work for MAX-CUT, Feige and Goemans [5] for MAX-DICUT and MAX 2-SAT, Zwick [18] for constraint satisfaction problems, and Nesterov[13] and Zwick [19] for MAX-CUT. And this kind of randomized algorithm can be de-randomized by Mahajan and Ramesh[12].

What complicates matters in the MAX-GP problem, comparing to the MAX-CUT problem, is that two objectives are sought—the objective value of $w(S)$ and the size of S. Therefore, in any (randomized) rounding method, we need to balance the (expected) quality of $w(S)$ and the (expected) size of S. We want high $w(S)$; but, at the same time, zero or small difference between $|S|$ and k, since otherwise we have to either add or subtract nodes from S, resulting in a deterioration of $w(S)$ at the end. Our new rounding method is built upon this balance need.

As consequences of the new rounding method, we have yielded improved approximation performance ratios for DSP, MC, MNC and MVC, on a wide range of k. On approximating DSP, for example, our algorithm has guaranteed performance ratios .648 for $k = 3n/5$, 0.586 for $k = n/2$, 0.486 for $k = 2n/5$ and 0.278 for $k = n/4$. For MC and MNC, the performance guarantees are also much better than 0.5 for a wide range of k. On approximating MVC, our algorithm has guaranteed performance ratios .845 for $k = 3n/5$, 0.811 for $k = n/2$, and 0.733 for $k = 2n/5$.

This paper is organized as follows. In the next two sections, we present an SDP relaxation of the MAX-GP problem, the method of rounding an optimal SDP solution, and preliminary analyses of the method. In sections 4, 5 and 6 we prove specific approximation ratios for DSP, MC and MNC, respectively.

2 SDP Relaxation and Rounding Method for MAX-GP

By introducing a binary variable $x_i \in \{-1, 1\}$ to each node $i \in V$, where $i \in S$ if and only if $x_i = 1$, the MAX-GP problem can be formulated as a binary quadratic program with a linear constraint:

$$
\begin{array}{llll}
w^* := & \text{Maximize} & \displaystyle\sum_{1 \le i < j \le n} w_{ij}(c_0 + c_1 \cdot x_i + c_2 \cdot x_j + c_3 \cdot x_i x_j) \\
\text{(MAX-GP)} & & \\
& \text{subject to} & \displaystyle\sum_{i=1}^{n} x_i = 2k - n, \\
& & x_i \in \{-1, 1\}, \quad i = 1, \ldots, n,
\end{array}
$$

where coefficients c_0, c_1, c_2, c_3 are

MAX-GP Problem	c_0	c_1	c_2	c_3
DSP	1/4	1/4	1/4	1/4
MC	1/2	0	0	-1/2
MNC	1/2	0	0	1/2
MVC	3/4	1/4	1/4	-1/4

To construct an SDP relaxation of (MAX-GP), we introduce a "reference" binary variable x_0, where $i \in S$ if and only if $x_i = x_0$. Then, (MAX-GP) can be rewritten as a homogeneous quadratic program:

$$
\begin{array}{llll}
w^* := & \text{Maximize} & \displaystyle\sum_{1 \le i < j \le n} w_{ij}(c_0 + c_1 \cdot x_i x_0 + c_2 \cdot x_j x_0 + c_3 \cdot x_i x_j) \\
\text{(MAX-GP)} & & \\
& \text{subject to} & \displaystyle\sum_{i=1}^{n} x_i x_0 = 2k - n, \\
& & x_i^2 = 1, \quad i = 0, 1, \cdots, n.
\end{array}
$$

Noting further that

$$
\sum_{1 \le i, j \le n} x_i x_j = \left(\sum_{i=1}^{n} x_i x_0 \right)^2 = (2k - n)^2,
$$

we have an SDP relaxation of (MAX-GP) as:

$$
\begin{array}{llll}
w^{SDP} := & \text{Maximize} & \sum\limits_{1\le i<j\le n} w_{ij}(c_0 + c_1 \cdot X_{i0} + c_2 \cdot X_{j0} + c_3 \cdot X_{ij}) & \\
\text{(SDP-GP)} & & & \\
& \text{subject to} & \sum\limits_{1\le i,j\le n} X_{ij} = (2k-n)^2, & \\
& & & (1) \\
& & \sum\limits_{i=1}^{n} X_{i0} = 2k-n, & \\
& & X_{ii} = 1, \quad i = 0, 1, \cdots, n, & \\
& & X \succeq 0. &
\end{array}
$$

where $X \in R^{(n+1)\times(n+1)}$ is the unknown symmetric matrix, and $X \succeq Z$ means that $X - Z$ is positive semidefinite. Obviously, $w^{SDP} \ge w^*$, since, for any feasible solution $x = (x_0, x_1, \cdots, x_n)$ of (MAX-GP), $X = xx^T$ is feasible for (SDP-GP).

Our generic approximation method for (MAX-GP) consists of the following three steps:

Algorithm SDP-GP

1. SDP-GP Solving Solve (SDP-GP) to obtain an optimal semidefinite matrix $\bar{X}$.

Repeat the following two steps many times, and output the best subset.

2. Randomized Rounding Using the multivariate normal distribution with 0 mean and the covariance matrix $\theta\bar{X} + (1-\theta)P$, for a given $0 \le \theta \le 1$, to generate $u, \hat{x}$ and S as:

$$u \in N(0, \theta\bar{X} + (1-\theta)P);$$

$$\hat{x} = \mathrm{sign}(u),$$

i.e.,

$$\hat{x}_i = \begin{cases} 1 & \text{if } u_i \ge 0 \\ -1 & \text{if } u_i < 0; \end{cases}$$

$$S = \{i \ge 1 : \ \hat{x}_i = \hat{x}_0\};$$

where P is the positive semidefinite matrix

$$P = \begin{pmatrix} 1 & \kappa & \kappa & \cdots & \kappa \\ \kappa & 1 & \kappa^2 & \cdots & \kappa^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \kappa & \kappa^2 & \cdots & \kappa^2 & 1 \end{pmatrix},$$

and $\kappa = 2\sigma - 1$ and $\sigma = k/n$.

3. Size Adjusting Using the greedy algorithm, either adding or removing one node at a time while causing the least deterioration in $w(S)$, to resize S to k, and output S. ■

We will specify the detailed size adjusting procedure later for each individual problem.

The rounding method is equivalent to that of Goemans and Williamson [10] if $P = 0$ or $\theta = 1$, and identical to that of Zwick [19] and Ye [15] if $\kappa = 0$. Goemans and Williamson [10] and Bertsimas and Ye [4] have showed that the rounding has the following properties:

$$\begin{aligned} \mathrm{E}[\hat{x}_0\hat{x}_j] &= \frac{2}{\pi}\arcsin(\theta\bar{X}_{0j} + (1-\theta)\kappa), \quad \text{for} \quad j = 1,\ldots,n, \\ \mathrm{E}[\hat{x}_i\hat{x}_j] &= \frac{2}{\pi}\arcsin(\theta\bar{X}_{ij} + (1-\theta)\kappa^2), \quad \text{for} \quad i,j = 1,\ldots,n. \end{aligned}$$

For clarity, we denote the subset after adjusting by $\tilde{S}$, that is, S denotes the selected subset before adjusting; and let $|S| = \delta \cdot n$ $(0 \le \delta \le 1)$. To analyze the quality of the algorithm, we need to establish a bound on

$$\frac{\mathrm{E}[w(\tilde{S})]}{w^*}.$$

Let the greedy algorithm in the size adjusting procedure possess

$$w(\tilde{S}) \ge r(\delta) \cdot w(S), \tag{2}$$

where $r(\delta) \le 1$ is a "deterioration" function ($r(\delta) = 1$ if $\delta = k/n$, i.e., no deterioration occurred), and

$$w(S) = \sum_{1 \le i < j \le n} w_{ij}\left(c_0 + c_1\hat{x}_i\hat{x}_0 + c_2\hat{x}_j\hat{x}_0 + c_3\hat{x}_i\hat{x}_j\right).$$

Again, the overall performance of the algorithm is determined by two factors: the expected quality of $w(S)$ and how much the size of S need to be adjusted. To make sure that $|S|$ is close to k before resizing, we have added two seemly redundant constraints on the top of (SDP-GP) of (1), which will be used to bound the expected value as well as the "variance" of $|S|$. In addition, the convex combination parameter θ used in $\theta\bar{X} + (1-\theta)P$ provides a balance between these two factors. Typically, the more use of $\bar{X}$ in the combination results in higher expected quality $w(S)$ but larger expected difference between $|S|$ and k; and the more use of P results in less $w(S)$ and more accurate $|S|$.

To see this, one can easily verify that if we choose $\theta = 0$ and $\kappa = sin(\frac{\pi}{2} \cdot (2\frac{k}{n} - 1))$, our rounding method would generate an S whose expected size is exactly k.

3 Analysis of the SDP-GP Algorithm

To analyze an expected lower bound for $w(\tilde{S})$, from (2) we now wish to lower bound

$$\frac{\mathrm{E}[r(\delta)\cdot w(S)]}{w^*}.$$

However, since S appeared in each term of the product, it is typically hard to calculate its expected value. The idea is to construct a class of artificial and auxiliary random variables, say z_S, whose expected value would be easily estimated and bounded; moreover, whenever z_S fulfills its expectation, we have

$$r(\delta)\cdot w(S)\geq R(z_S)\cdot w^*.$$

This would establish a performance ratio $R(z_S)$ for the SDP-GP algorithm. Naturally, we like to select the z_S that yields the largest $R(z_S)$.

The artificial random variables we used through out this paper are linear combinations of $w(S)/w^*$, $|S|/n$ and $|S|(n-|S|)/n^2$:

$$z(\gamma,\eta):=\frac{w(S)}{w^*}+\gamma\eta\frac{|S|}{n}+\gamma\frac{|S|(n-|S|)}{n^2}, \tag{3}$$

where $\eta\in[-1,1]$ and $\gamma\geq 0$ are two linear combination parameters. Since

$$0\leq\frac{w(S)}{w^*}\leq\frac{n(n-1)}{2},$$

$$-\gamma|\eta|\leq z(\gamma,\eta)\leq\frac{n(n-1)}{2}+\frac{\gamma(1+\eta)^2}{4},$$

i.e., $z(\gamma,\eta)$ is always bounded.

Thus, if we could show

$$\mathrm{E}[\frac{w(S)}{w^*}]\geq\alpha,\quad \mathrm{E}[\eta\frac{|S|}{n}]\geq\eta\beta_1,\quad\text{and}\quad \mathrm{E}[\gamma\frac{|S|(n-|S|)}{n^2}]\geq\gamma\beta_2,$$

then

$$\mathrm{E}[z(\gamma,\eta)]\geq\mu_{\eta,\gamma}:=\alpha+\gamma(\eta\beta_1+\beta_2).$$

Furthermore, if $z(\gamma,\eta)$ meets its expectation, we must have

$$\frac{w(\tilde{S})}{w^*}\geq r(\delta)\frac{w(S)}{w^*}\geq r(\delta)(\alpha+\gamma\eta\beta_1+\gamma\beta_2-\gamma\eta\delta-\gamma\delta(1-\delta))\geq R(\eta,\gamma,\theta)$$

where

$$R(\eta,\gamma,\theta) := \begin{array}{ll} \text{Minimize} & r(\delta)(\alpha+\gamma\eta\beta_1+\gamma\beta_2-\gamma\eta\delta-\gamma\delta(1-\delta)) \\ \text{subject to} & 0\le\delta\le 1. \end{array} \tag{4}$$

Note that $R(\eta,\gamma,\theta)$ is also a function of θ, since α, β_1 and β_2 depend on the balance combination parameter θ.

Finally, we select

$$R_k^* := \begin{array}{ll} \text{Maximize} & R(\eta,\gamma,\theta) \\ \text{subject to} & 0\le\theta\le 1, \\ & \\ & \gamma\ge 0, \\ & -1\le\eta\le 1. \end{array} \tag{5}$$

and we have our main result

Theorem 1 *The worst-case performance ratio of the SDP-GP algorithm for approximating the MAX-GP problem with size requirement k is at least R_k^*.*

■

We now establish bounds on $\mathrm{E}[\frac{|S|}{n}]$ and $\mathrm{E}[\frac{|S|(n-|S|)}{n^2}]$, because they are universal for solving all MAX-GP problems. In later sections, we will establish an explicit $r(\delta)$ and a bound on $\mathrm{E}[\frac{w(S)}{w^*}]$ for each individual MAX-GP problem.

Generally, for a given $0\le\theta\le 1$, let

$$\begin{aligned}
\tau^+ = \tau^+(\theta) &:= \min_{-1\le y\le 1} \frac{2}{\pi}\frac{\arcsin(\theta-(1-\theta)\kappa)+\arcsin(\theta y+(1-\theta)\kappa)}{1+y} \\
\beta^+ = \beta^+(\theta) &:= 1-\tfrac{2}{\pi}\arcsin(\theta-(1-\theta)\kappa)+2\sigma\tau^+(\theta)
\end{aligned}$$

$$\begin{aligned}
\tau^- = \tau^-(\theta) &:= \min_{-1\le y\le 1} \frac{2}{\pi}\frac{\arcsin(\theta+(1-\theta)\kappa)-\arcsin(\theta y+(1-\theta)\kappa)}{1-y} \\
\beta^- = \beta^-(\theta) &:= 1-\tfrac{2}{\pi}\arcsin(\theta+(1-\theta)\kappa)+2(1-\sigma)\tau^-(\theta)
\end{aligned}$$

$$\begin{aligned}
\tau = \tau(\theta) &:= \min_{-1\le y\le 1} \frac{2}{\pi}\frac{\arcsin(\theta+(1-\theta)\kappa^2)-\arcsin(\theta y+(1-\theta)\kappa^2)}{1-y} \\
\beta = \beta(\theta) &:= 1-\tfrac{2}{\pi}\arcsin(\theta+(1-\theta)\kappa^2)+4\sigma(1-\sigma)\tau(\theta).
\end{aligned}$$

Then, we have the following lemma regarding $|S|$:

Lemma 1 *Let*

$$\beta_1 = \begin{cases} \frac{\beta^+}{2} & \textit{if} \quad \eta \geq 0, \\ \frac{2-\beta^-}{2} & \textit{otherwise;} \end{cases}$$

and

$$\beta_2 = \frac{\beta - (\beta - 4\sigma(1-\sigma)\tau)/n}{4}.$$

Then, the approximation algorithm yields S satisfying the following inequalities:

$$\mathrm{E}[\eta \frac{|S|}{n}] \geq \eta\beta_1,$$
$$\mathrm{E}[\frac{|S|(n-|S|}{n^2}] \geq \beta_2.$$

Proof. We need to prove

$$\beta^+ \frac{n}{2} \leq \mathrm{E}[|S|] \leq \beta^- \frac{n}{2},$$

which can be seen from:

$$\begin{aligned}
&\mathrm{E}[|S|] \\
&= \sum_{i=1}^{n} [Pr(x_i = 1, x_0 = 1) + Pr(x_i = -1, x_0 = -1)] \\
&= \sum_{i=1}^{n} [Pr(u_i \geq 0, u_0 \geq 0) + Pr(u_i < 0, u_0 < 0)] \\
&= \sum_{i=1}^{n} [\frac{1}{2} + \frac{1}{\pi} \arcsin(\theta \bar{X}_{0i} + (1-\theta)\kappa)] \\
&= \sum_{i=1}^{n} [\frac{1}{2} - \frac{1}{\pi} \arcsin(\theta - (1-\theta)\kappa) + \\
&\quad \frac{1}{\pi} \arcsin(\theta - (1-\theta)\kappa) + \frac{1}{\pi} \arcsin(\theta \bar{X}_{0i} + (1-\theta)\kappa)] \\
&\geq \frac{1}{2} \sum_{i=1}^{n} [1 - \frac{2}{\pi} \arcsin(\theta - (1-\theta)\kappa) + \tau^+(1 + \bar{X}_{0i})] \\
&= \frac{1}{2} \left((1 - \frac{2}{\pi} \arcsin(\theta - (1-\theta)\kappa))n + 2k\tau^+ \right) \\
&= \beta^+ \frac{n}{2},
\end{aligned}$$

and

$$\begin{aligned}
&\mathrm{E}[|S|] \\
&= \sum_{i=1}^{n} [Pr(x_i = 1, x_0 = 1) + Pr(x_i = -1, x_0 = -1)]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{n}[Pr(u_i \geq 0, u_0 \geq 0) + Pr(u_i < 0, u_0 < 0)] \\
&= \sum_{i=1}^{n}[\frac{1}{2} + \frac{1}{\pi}\arcsin(\theta \bar{X}_{0i} + (1-\theta)\kappa)] \\
&= n - \frac{1}{2}\sum_{i=1}^{n}[1 - \frac{2}{\pi}\arcsin(\theta \bar{X}_{0i} + (1-\theta)\kappa)] \\
&= n - \frac{1}{2}\sum_{i=1}^{n}[1 - \frac{2}{\pi}\arcsin(\theta + (1-\theta)\kappa) + \\
&\quad \frac{2}{\pi}\arcsin(\theta + (1-\theta)\kappa) - \frac{2}{\pi}\arcsin(\theta \bar{X}_{0i} + (1-\theta)\kappa)] \\
&\leq n - \frac{1}{2}\sum_{i=1}^{n}[1 - \frac{2}{\pi}\arcsin(\theta + (1-\theta)\kappa) + \tau^{-}(1-\bar{X}_{0i})] \\
&= n - \frac{1}{2}\left((1 - \frac{2}{\pi}\arcsin(\theta + (1-\theta)\kappa))n + 2(n-k)\tau^{-}\right) \\
&= (2-\beta^{-})\frac{n}{2}.
\end{aligned}
$$

Noting that

$$\sum_{i \neq j} \bar{X}_{ij} = (2k-n)^2 - n,$$

we derive

$$
\begin{aligned}
&4\mathrm{E}[|S|(n-|S|)] \\
&= n^2 - \mathrm{E}[(e^T \hat{x})^2] \\
&= \sum_{i \neq j}\left(1 - \frac{2}{\pi}\arcsin(\theta \bar{X}_{ij} + (1-\theta)\kappa^2)\right) \\
&= \sum_{i \neq j}\left(1 - \frac{2}{\pi}\arcsin(\theta + (1-\theta)\kappa^2) + \right. \\
&\quad \left. \frac{2}{\pi}\arcsin(\theta + (1-\theta)\kappa^2) - \frac{2}{\pi}\arcsin(\theta \bar{X}_{ij} + (1-\theta)\kappa^2)\right) \\
&\geq \sum_{i \neq j}\left(1 - \frac{2}{\pi}\arcsin(\theta + (1-\theta)\kappa^2) + \tau(1-\bar{X}_{ij})\right) \\
&= (n^2-n)\left(1 - \frac{2}{\pi}\arcsin(\theta + (1-\theta)\kappa^2)\right) + (n^2 - (2k-n)^2)\tau \\
&= \left(\beta - \frac{1 - \frac{2}{\pi}\arcsin(\theta + (1-\theta)\kappa^2)}{n}\right) \cdot n^2
\end{aligned}
$$

■

4 Approximation of DSP

The detailed greedy algorithm used in the size-adjusting procedure for DSP is:

3. Size Adjusting for DSP Let $\tilde{S} = S$, where S is the subset determined in Step 2.

- If $|\tilde{S}| > k$, then for each $i \in \tilde{S}$, let $\zeta(i) = \sum_{j \in \tilde{S}} w_{ij}$ and $\tilde{S} = \{i_1, i_2, ..., i_{|\tilde{S}|}\}$, where $\zeta(i_1) \geq \zeta(i_2) \geq ... \geq \zeta(i_{|\tilde{S}|})$. Then, remove node $i_{|\tilde{S}|}$ from $\tilde{S}$ and reassign $\tilde{S} := \{i_1, i_2, ..., i_{|\tilde{S}|-1}\}$. Repeat this swapping process till $|\tilde{S}| = k$.

- If $|\tilde{S}| < k$, arbitrarily add $k - |\tilde{S}|$ nodes to $\tilde{S}$. ■

The following lemma is due to Srivastav and Wolf [14]:

Lemma 2 *The total weights of edges in the subgraph induced by $\tilde{S}$ resulted from the SDP-GP algorithm satisfies*

$$|\tilde{S}| = k$$

and

$$w(\tilde{S}) \geq \begin{cases} \frac{k(k-1)}{|S|(|S|-1)} w(S) & \text{if} \quad |S| > k; \\ w(S) & \text{if} \quad |S| \leq k. \end{cases} \tag{6}$$

■

Recall

$$w(S) = \frac{1}{4} \sum_{1 \leq i < j \leq n} w_{ij} \left(1 + \hat{x}_0 \hat{x}_i + \hat{x}_0 \hat{x}_j + \hat{x}_i \hat{x}_j\right).$$

For simplicity, in what follows we treat $\frac{k(k-1)}{|S|(|S|-1)}$ as $\frac{k^2}{|S|^2}$, this is because, for n and k sufficiently large, they are virtually identical. Then, based on Lemma 2, the deterioration function $r(\delta)$ of (2) for DSP is

$$r(\delta) = \begin{cases} \frac{\sigma^2}{\delta^2} & \text{if} \quad \delta \geq \sigma, \\ 1 & \text{otherwise.} \end{cases}.$$

Now we analyze $\mathrm{E}[w(S)]/w^*$ for DSP. Let

$$\alpha(\theta) := \max\left\{\alpha : \begin{array}{l} 1+\frac{2}{\pi}(\arcsin(\theta x+(1-\theta)\kappa)+\arcsin(\theta y+(1-\theta)\kappa)+\arcsin(\theta z+(1-\theta)\kappa^2)) \geq \\ \alpha\cdot(1+x+y+z), \\ \\ \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \succeq 0 \end{array}\right\}.$$

Since

$$1+\frac{2}{\pi}(\arcsin(a)+\arcsin(b)+\arcsin(c)) \geq 0$$

for all a, b, c such that

$$\begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix} \succeq 0,$$

$$1+\frac{2}{\pi}(\arcsin(\theta x+(1-\theta)\kappa)+\arcsin(\theta y+(1-\theta)\kappa)+\arcsin(\theta z+(1-\theta)\kappa^2)) \geq 0$$

for all x, y, z such that

$$\begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \succeq 0.$$

Therefore, $\alpha(\theta)$ can be computed from

$$\begin{array}{rll} \alpha = \alpha(\theta) := \min & \frac{1+\frac{2}{\pi}(\arcsin(\theta x+(1-\theta)\kappa)+\arcsin(\theta y+(1-\theta)\kappa)+\arcsin(\theta z+(1-\theta)\kappa^2)}{1+x+y+z} & \\ \\ \text{subject to} & 1+x+y+z > 0, & \\ & \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \succeq 0. & \end{array} \tag{7}$$

One can further verify that the above minimum achieved when

$$\theta x+(1-\theta)\kappa = \theta y+(1-\theta)\kappa = \theta z+(1-\theta)\kappa^2,$$

for a certain range of θ and κ.

Therefore, we have the final piece of the analysis:

Lemma 3 *Let α be computed from (7). Then,*

$$\mathrm{E}[w(S)] \geq \alpha \cdot w^{SDP} \geq \alpha \cdot w^*.$$

Proof.

$$\begin{aligned}
\mathrm{E}[w(S)] &= \frac{1}{4}\sum_{1\leq i<j\leq n} w_{ij}\left(1+\mathrm{E}[\hat{x}_0\hat{x}_i]+\mathrm{E}[\hat{x}_0\hat{x}_j]+\mathrm{E}[\hat{x}_i\hat{x}_j]\right)\\
&= \frac{1}{4}\sum_{i<j} w_{ij}\Big(1+\frac{2}{\pi}(\arcsin(\theta\bar{X}_{0j}+(1-\theta)\kappa)\\
&\quad +\arcsin(\theta\bar{X}_{0i}+(1-\theta)\kappa)+\arcsin(\theta\bar{X}_{ij}+(1-\theta)\kappa^2))\Big)\\
&\geq \frac{\alpha}{4}\sum_{i<j} w_{ij}\left(1+\bar{X}_{0j}+\bar{X}_{0i}+\bar{X}_{ij}\right)\\
&= \alpha w^{SDP} \geq \alpha w^*.
\end{aligned}$$

■

We can now go ahead to compute $R(\eta,\gamma,\theta)$ and R_k^* from (4) and (5) for any given k.

In fact, γ in (5) can be explicit given as follows. Restrict $\eta \in [-1, 2\sigma-1]$ and let

$$\gamma = \begin{cases} \dfrac{\alpha}{\sigma^2-(\eta\beta_1+\beta_2)}, & \text{if } 2\sigma^2-1\leq\eta\leq 2\sigma-1 \\[2ex] \dfrac{\alpha(1-\sigma^2)}{\sigma^2((\eta\beta_1+\beta_2)-\eta)+(1+\eta)^2/4-(\eta\beta_1+\beta_2)}, & \text{if } -1\leq\eta<2\sigma^2-1. \end{cases} \tag{8}$$

Then, we have

Proposition 1 *If the random variable $z(\gamma,\eta)$ fulfills its expectation, i.e., $z(\gamma,\eta)\geq \mu_{\eta,\gamma} := \alpha+\gamma(\eta\beta_1+\beta_2)$, then*

$$w(\tilde{S}) \geq R(\eta,\theta)\cdot w^*,$$

and

$$R(\eta,\theta) = \begin{cases} \dfrac{\alpha(4\sigma^2-(1+\eta)^2)}{4(\sigma^2-(\eta\beta_1+\beta_2))}, & \textit{if} \quad 2\sigma^2-1\leq\eta\leq 2\sigma-1 \\[2ex] \dfrac{\alpha\sigma^2((1+\eta)^2/4-\eta)}{\sigma^2((\eta\beta_1+\beta_2)-\eta)+(1+\eta)^2/4-(\eta\beta_1+\beta_2)}, & \textit{if} \quad -1\leq\eta<2\sigma^2-1. \end{cases}$$

Proof. If $\delta > \sigma$, it is followed from Lemma 2 that

$$\begin{aligned} w(\tilde{S}) &\geq \frac{\sigma^2}{\delta^2} \cdot w(S) \\ &\geq \frac{\sigma^2(\mu_{\eta,\gamma} - \eta\gamma\delta - \gamma\delta(1-\delta))}{\delta^2} \cdot w^* \end{aligned}$$

If $2\sigma^2 - 1 \leq \eta \leq 2\sigma - 1$, then

$$\frac{\sigma^2(\mu_{\eta,\gamma} - \eta\gamma\delta - \gamma\delta(1-\delta))}{\delta^2} \geq \frac{\sigma^2(4\mu_{\eta,\gamma} - \gamma(1+\eta)^2)\gamma}{4\mu_{\eta,\gamma}}$$

This inequality follows from simple calculus and (8) that

$$\delta = \frac{2\mu_{\eta,\gamma}}{(1+\eta)\gamma} = \frac{2}{1+\eta} \tag{9}$$

yields the minimal value for

$$\frac{\mu_{\eta,\gamma} - \eta\gamma\delta - \gamma\delta(1-\delta))}{\delta^2}.$$

If $-1 \leq \eta < 2\sigma^2 - 1$, then

$$\frac{\sigma^2(\mu_{\eta,\gamma} - \eta\gamma\delta - \gamma\delta(1-\delta))}{\delta^2} \geq \sigma^2(\mu_{\eta,\gamma} - \eta\gamma),$$

since from simple calculus, (8) and the fact that

$$\delta = 1$$

yields the minimal value for

$$\frac{\mu_{\eta,\gamma} - \eta\gamma\delta - \gamma\delta(1-\delta)}{\delta^2}.$$

If $\delta \leq \sigma$, the number of nodes in S is no more than k and no vertex have to be removed from S. Then,

$$\begin{aligned} w(\tilde{S}) &= w(S) \\ &\geq (\mu_{\eta,\gamma} - \eta\gamma\delta - \gamma\delta(1-\delta)) \cdot w^* \\ &\geq (\mu_{\eta,\gamma} - \frac{\gamma(1+\eta)^2}{4}) \cdot w^* \end{aligned}$$

The third inequality follows from the simple fact that

$$\delta = \frac{1+\eta}{2} \tag{10}$$

yields the maximal value of

$$\eta\delta + \delta(1-\delta).$$

Using (8) we have, for $2\sigma^2 - 1 \le \eta \le 2\sigma - 1$,

$$\frac{\sigma^2(4\mu_{\eta,\gamma} - \gamma(1+\eta)^2)\gamma}{4\mu_{\eta,\gamma}} = \mu_{\eta,\gamma} - \frac{\gamma(1+\eta)^2}{4} = \frac{\alpha\,(4\sigma^2 - (1+\eta)^2)}{4(\sigma^2 - (\eta\beta_1 + \beta_2))}.$$

and, for $-1 \le \eta < 2\sigma^2 - 1$,

$$\begin{aligned} \sigma^2(\mu_{\eta,\gamma} - \eta\gamma) &= \mu_{\eta,\gamma} - \gamma(1+\eta)^2/4 \\ &= \frac{\alpha\sigma^2((1+\eta)^2/4 - \eta)}{\sigma^2((\eta\beta_1 + \beta_2) - \eta) + (1+\eta)^2/4 - (\eta\beta_1 + \beta_2)}. \end{aligned}$$

This completes the proof. ■

We have calculated Table 1, which shows $R(\eta, \gamma, \theta)$ for $0.2n \le k \le 0.9995n$. Note that the true performance ratio R_k^* for DSP should be greater than $R(\eta, \gamma, \theta)$ shown in the table.

5 Approximation of MC

Because the symmetry of this problem, with out lossing generality, we assume that the required size $k \ge n/2$ or $\sigma \ge 1/2$. The detailed greedy algorithm used in the adjusting procedure for MC is:

3. Size Adjusting for MC Let $\tilde{S} = S$, where S is the subset determined in Step 2.

• If $|S| > k$, then for each $i \in \tilde{S}$, let $\zeta(i) = \sum_{j \in V \setminus \tilde{S}} w_{ij}$ and $\tilde{S} = \{i_1, i_2, ..., i_{|\tilde{S}|}\}$, where $\zeta(i_1) \ge \zeta(i_2) \ge ... \ge \zeta(i_{|\tilde{S}|})$. Then, remove node $i_{|\tilde{S}|}$ from $\tilde{S}$ and reassign $\tilde{S} := \{i_1, i_2, ..., i_{|\tilde{S}|-1}\}$. Repeat this swapping process till $|\tilde{S}| = k$.

• If $n/2 \le |S| < k$, then for each $i \in V \setminus \tilde{S}$, let $\zeta(i) = \sum_{j \in \tilde{S}} w_{ij}$ and $V \setminus \tilde{S} = \{i_1, i_2, ..., i_{|V \setminus \tilde{S}|}\}$, where $\zeta(i_1) \ge \zeta(i_2) \ge ... \ge \zeta(i_{|V \setminus \tilde{S}|})$. Then, add node $i_{|V \setminus \tilde{S}|}$ to $\tilde{S}$ and reassign $V \setminus \tilde{S} := \{i_1, i_2, ..., i_{|V \setminus \tilde{S}|-1}\}$. Repeat this swapping process till $|\tilde{S}| = k$.

$\sigma = \frac{k}{n}$	θ	α	η	γ	$R(\eta, \gamma, \theta)$
0.20	0.94	0.7509	-0.8417	5.9521	0.2008
0.22	0.94	0.7514	-0.8080	5.9211	0.2320
0.24	0.94	0.7520	-0.7719	5.8990	0.2631
0.26	0.92	0.7381	-0.7284	5.9836	0.2942
0.28	0.92	0.7390	-0.6877	6.0083	0.3245
0.30	0.92	0.7399	-0.6455	6.0454	0.3541
0.32	0.92	0.7410	-0.6020	6.0947	0.3827
0.34	0.90	0.7284	-0.5499	6.3204	0.4105
0.36	0.90	0.7298	-0.5039	6.4235	0.4372
0.38	0.90	0.7314	-0.4573	6.5374	0.4626
0.40	0.90	0.7330	-0.4104	6.6593	0.4867
0.42	0.88	0.7220	-0.3544	7.0553	0.5095
0.44	0.88	0.7241	-0.3072	7.2155	0.5310
0.46	0.88	0.7264	-0.2604	7.3644	0.5511
0.48	0.88	0.7287	-0.2142	7.4910	0.5697
0.50	0.88	0.7311	-0.1690	7.5830	0.5866
0.52	0.88	0.7337	-0.1249	7.6282	0.6022
0.54	0.88	0.7364	-0.0820	7.6162	0.6161
0.56	0.88	0.7391	-0.0403	7.5412	0.6287
0.58	0.90	0.7517	-0.0071	7.1198	0.6402
0.60	0.90	0.7542	0.0	5.8985	0.6488
0.62	0.92	0.7652	0.0	4.8650	0.6539
0.64	0.92	0.7674	0.0	4.1124	0.6563
0.65	0.92	0.7685	0.0124	3.9524	0.6571
0.66	0.92	0.7697	0.0248	3.8006	0.6577
0.68	0.92	0.7720	0.0490	3.5173	0.6588
0.70	0.90	0.7683	0.0822	3.3455	0.6597
0.72	0.90	0.7714	0.1045	3.0955	0.6607
0.74	0.90	0.7746	0.1261	2.8686	0.6615
0.76	0.90	0.7780	0.1470	2.6624	0.6621
0.78	0.88	0.7775	0.1730	2.5074	0.6630
0.80	0.88	0.7817	0.2273	2.5010	0.6636
0.82	0.88	0.7861	0.2108	2.0923	0.6687
0.84	0.88	0.7906	0.0	1.1280	0.6759
0.86	0.86	0.7940	0.0	0.9660	0.6859
0.88	0.82	0.7978	0.0	0.8142	0.6987
0.90	0.78	0.8042	0.0	0.6708	0.7149
0.92	0.74	0.8135	0.0	0.5340	0.7356
0.94	0.66	0.8262	0.0	0.4017	0.7624
0.96	0.58	0.8460	0.0	0.2717	0.7985
0.98	0.44	0.8790	0.0	0.1404	0.8515
0.99	0.32	0.9078	0.0	0.0725	0.8926
0.995	0.22	0.9314	0.0	0.0372	0.9232
0.9995	0.08	0.9764	0.0	0.0039	0.9755

Table 1: Worst-case performance ratios of the SDP-DSP algorithm

- If $n-k \le |S| < n/2$, then for each $i \in \tilde{S}$, let $\zeta(i) = \sum_{j\in V\setminus\tilde{S}} w_{ij}$ and $\tilde{S} = \{i_1, i_2, ..., i_{|\tilde{S}|}\}$, where $\zeta(i_1) \ge \zeta(i_2) \ge ... \ge \zeta(i_{|\tilde{S}|})$. Then, remove node $i_{|\tilde{S}|}$ from $\tilde{S}$ and reassign $\tilde{S} := \{i_1, i_2, ..., i_{|\tilde{S}|-1}\}$. Repeat this swapping process till $|\tilde{S}| = n-k$.

- If $|S| < n-k$, then for each $i \in V \setminus \tilde{S}$, let $\zeta(i) = \sum_{j\in \tilde{S}} w_{ij}$ and $V\setminus\tilde{S} = \{i_1, i_2, ..., i_{|V\setminus\tilde{S}|}\}$, where $\zeta(i_1) \ge \zeta(i_2) \ge ... \ge \zeta(i_{|V\setminus\tilde{S}|})$. Then, add node $i_{|V\setminus\tilde{S}|}$ to $\tilde{S}$ and reassign $V\setminus\tilde{S} := \{i_1, i_2, ..., i_{|V\setminus\tilde{S}|-1}\}$. Repeat this swapping process till $|\tilde{S}| = n-k$.

■

The following lemma is straightforward:

Lemma 4 *The total weights of crossing edges between $\tilde{S}$ and $V\setminus\tilde{S}$ resulted from the approximation algorithm satisfy*

$$\textit{either } |\tilde{S}| = k \textit{ or } |V\setminus\tilde{S}| = k$$

and

$$w(\tilde{S}) \ge \begin{cases} \frac{k}{|S|} w(S) & \textit{if} \quad |S| \ge k \\ \frac{n-k}{n-|S|} w(S) & \textit{if} \quad n/2 \le |S| < k \\ \frac{n-k}{|S|} w(S) & \textit{if} \quad n-k \le |S| < n/2 \\ \frac{k}{n-|S|} w(S) & \textit{if} \quad |S| < n-k \end{cases}. \tag{11}$$

Then, based on Lemma 4, the deterioration function $r(\delta)$ of (2) for MC is

$$r(\delta) = \begin{cases} \frac{\sigma}{\delta} & \text{if} \quad \delta \ge \sigma \\ \frac{1-\sigma}{1-\delta} & \text{if} \quad 1/2 \le \delta < \sigma \\ \frac{1-\sigma}{\delta} & \text{if} \quad 1-\sigma \le \delta < 1/2 \\ \frac{\sigma}{1-\delta} & \text{if} \quad \delta < 1-\sigma \end{cases}.$$

Now we analyze $\mathrm{E}[w(S)]/w^*$ for MC. Note

$$w(S) = \frac{1}{2} \sum_{1\le i<j\le n} w_{ij} \left(1 - \hat{x}_i \hat{x}_j\right).$$

Let

$$\begin{aligned} \alpha = \alpha(\theta) := \min \quad & \frac{1-\frac{2}{\pi}\arcsin(\theta z + (1-\theta)\kappa^2)}{1-z} \\ \text{subject to} \quad & -1 \le z < 1. \end{aligned} \tag{12}$$

Then, we have the following inequality whose proof resembles that for Lemma 3:

$$\mathrm{E}[w(S)] \ge \alpha \cdot w^{SDP} \ge \alpha \cdot w^*.$$

Now we can go ahead compute $R(\eta,\gamma,\theta)$ and R_k^* from (4) and (5). In fact, $R(\eta,\gamma,\theta)$ can be constructed from the following four cases.

Let

$$\lambda_{\eta,\gamma}(\delta) = \mu_{\eta,\gamma} - \gamma(\eta\delta + \delta(1-\delta)) = \alpha + \gamma(\eta\beta_1 + \beta_2) - \gamma(\eta\delta + \delta(1-\delta));$$

Case1. $\sigma \le \delta \le 1$:

$$R_1(\eta,\gamma,\theta) = \begin{cases} \sigma\left(2\sqrt{\gamma\mu_{\eta,\gamma}} - \eta\gamma - \gamma\right) & \text{if } \sigma \le \sqrt{\frac{\mu_{\eta,\gamma}}{\gamma}} \le 1, \\ \min\{\lambda_{\eta,\gamma}(\sigma), \sigma\lambda_{\eta,\gamma}(1)\} & \text{otherwise;} \end{cases}$$

Case2. $\frac{1}{2} \le \delta \le \sigma$:

$$R_2(\eta,\gamma,\theta) = \begin{cases} (1-\sigma)\left(2\sqrt{\gamma(\mu_{\eta,\gamma} - \eta\gamma)} + \eta\gamma - \gamma\right) & \text{if } 1-\sigma \le \sqrt{\frac{\mu_{\eta,\gamma}-\eta\gamma}{\gamma}} \le \frac{1}{2}, \\ \min\{\lambda_{\eta,\gamma}(\sigma), 2(1-\sigma)\lambda_{\eta,\gamma}(1/2)\} & \text{otherwise;} \end{cases}$$

Case3. $1-\sigma \le \delta \le \frac{1}{2}$:

$$R_3(\eta,\gamma,\theta) = \begin{cases} (1-\sigma)\left(2\sqrt{\gamma\mu_{\eta,\gamma}} - \eta\gamma - \gamma\right) & \text{if } 1-\sigma \le \sqrt{\frac{\mu_{\eta,\gamma}}{\gamma}} \le \frac{1}{2}, \\ \min\{\lambda_{\eta,\gamma}(1-\sigma), 2(1-\sigma)\lambda_{\eta,\gamma}(1/2)\} & \text{otherwise;} \end{cases}$$

Case4. $0 \le \delta \le 1-\sigma$:

$$R_4(\eta,\gamma,\theta) = \begin{cases} \sigma\left(2\sqrt{\gamma(\mu_{\eta,\gamma} - \eta\gamma)} + \eta\gamma - \gamma\right) & \text{if } \sigma \le \sqrt{\frac{\mu_{\eta,\gamma}-\eta\gamma}{\gamma}} \le 1, \\ \min\{\lambda_{\eta,\gamma}(1-\sigma), \sigma\lambda_{\eta,\gamma}(0)\} & \text{otherwise.} \end{cases}$$

Then, we have

$$R(\eta,\gamma,\theta) = \min\{R_1(\eta,\gamma,\theta), R_2(\eta,\gamma,\theta), R_3(\eta,\gamma,\theta), R_4(\eta,\gamma,\theta)\}.$$

We have generated Table 2, which shows $R(\eta,\gamma,\theta)$ for $0.5n \le k \le 0.98n$. Note that the true performance ratio R_k^* for MC should be greater than $R(\eta,\gamma,\theta)$ of the table, and $R(\eta,\gamma,\theta)$ for $0.5n \ge k \ge 0.02n$ are identical to those shown here.

6 Approximation of MNC

Because again the symmetry of this problem, with out lossing generality, we assume that the required size $\sigma \ge 1/2$. The detailed greedy algorithm used in the adjusting procedure for MNC is:

$\sigma = \frac{k}{n}$	θ	α	η	γ	$R(\eta, \gamma, \theta)$
0.50	0.88	0.831	0.00	7.41	0.699
0.52	0.90	0.839	0.00	6.86	0.725
0.54	0.94	0.855	0.00	3.91	0.734
0.56	0.96	0.863	0.00	2.36	0.721
0.58	0.98	0.871	0.00	1.61	0.698
0.60	0.98	0.870	0.00	1.16	0.671
0.62	0.98	0.870	0.00	0.76	0.642
0.64	1.00	0.878	0.00	0.46	0.616
0.66	1.00	0.878	0.00	0.11	0.593
0.68	1.00	0.878	0.00	0.00	0.562
0.70	1.00	0.878	0.00	0.00	0.527
0.72	1.00	0.878	0.00	0.00	0.492
0.74	1.00	0.878	0.00	0.00	0.456
0.76	1.00	0.878	0.00	0.00	0.421
0.78	1.00	0.878	0.00	0.00	0.386
0.80	1.00	0.878	0.00	0.00	0.351
0.82	1.00	0.878	0.00	0.00	0.316
0.84	1.00	0.878	0.00	0.00	0.281
0.86	1.00	0.878	0.00	0.00	0.246
0.88	1.00	0.878	0.00	0.00	0.210
0.90	1.00	0.878	0.00	0.00	0.175
0.92	1.00	0.878	0.00	0.00	0.140
0.94	1.00	0.878	0.00	0.00	0.105
0.96	1.00	0.878	0.00	0.00	0.070
0.98	1.00	0.878	0.00	0.00	0.035

Table 2: Worst-case performance ratios of the SDP-MC algorithm

3. Size Adjusting for MNC

Let $\tilde{S} = S$ where S is the subset determined in Step 2.

- If $|S| > k$, then for each $i \in \tilde{S}$, let $\zeta(i) = \sum_{j\tilde{S}} w_{ij}$ and $\tilde{S} = \{i_1, i_2, ..., i_{|\tilde{S}|}\}$, where $\zeta(i_1) \geq \zeta(i_2) \geq ... \geq \zeta(i_{|\tilde{S}|})$. Then, remove node $i_{|\tilde{S}|}$ from $\tilde{S}$ and reassign $\tilde{S} := \{i_1, i_2, ..., i_{|\tilde{S}|-1}\}$. Repeat this swapping process till $|\tilde{S}| = k$.

- If $n/2 \leq |S| < k$, then for each $i \in V \setminus \tilde{S}$, let $\zeta(i) = \sum_{j\in V\setminus\tilde{S}} w_{ij}$ and $V \setminus \tilde{S} = \{i_1, i_2, ..., i_{|V\setminus\tilde{S}|}\}$, where $\zeta(i_1) \geq \zeta(i_2) \geq ... \geq \zeta(i_{|V\setminus\tilde{S}|})$. Then, add node $i_{|V\setminus\tilde{S}|}$ to $\tilde{S}$ and reassign $V \setminus \tilde{S} := \{i_1, i_2, ..., i_{|V\setminus\tilde{S}|-1}\}$. Repeat this swapping process till $|\tilde{S}| = k$.

- If $n-k \leq |S| < n/2$, then for each $i \in \tilde{S}$, let $\zeta(i) = \sum_{j\in\tilde{S}} w_{ij}$ and $\tilde{S} = \{i_1, i_2, ..., i_{|\tilde{S}|}\}$, where $\zeta(i_1) \geq \zeta(i_2) \geq ... \geq \zeta(i_{|\tilde{S}|})$. Then, remove node $i_{|\tilde{S}|}$ from $\tilde{S}$ and reassign $\tilde{S} := \{i_1, i_2, ..., i_{|\tilde{S}|-1}\}$. Repeat this swapping process till $|\tilde{S}| = n - k$.

- If $|S| < n - k$, then for each $i \in V \setminus \tilde{S}$, let $\zeta(i) = \sum_{j\in V\setminus\tilde{S}} w_{ij}$ and $V \setminus \tilde{S} = \{i_1, i_2, ..., i_{|V\setminus\tilde{S}|}\}$, where $\zeta(i_1) \geq \zeta(i_2) \geq ... \geq \zeta(i_{|V\setminus\tilde{S}|})$. Then, add node $i_{|V\setminus\tilde{S}|}$ to $\tilde{S}$ and reassign $V \setminus \tilde{S} := \{i_1, i_2, ..., i_{|V\setminus\tilde{S}|-1}\}$. Repeat this swapping process till $|\tilde{S}| = n - k$.

■

The following lemma is also straightforward:

Lemma 5 *The total weights of non-crossing edges between $\tilde{S}$ and $V \setminus \tilde{S}$ resulted from the approximation algorithm satisfy*

$$\textit{either } |\tilde{S}| = k \textit{ or } |V \setminus \tilde{S}| = k$$

and

$$w(\tilde{S}) \geq \begin{cases} \frac{k(k-1)}{|S|(|S|-1)} w(S) & \textit{if} \quad |S| \geq k; \\ \frac{(n-k)(n-k-1)}{(n-|S|)(n-|S|-1)} w(S) & \textit{if} \quad n/2 \leq |S| < k; \\ \frac{(n-k)(n-k-1)}{|S|(|S|-1)} w(S) & \textit{if} \quad n-k \leq |S| < n/2; \\ \frac{k(k-1)}{(n-|S|)(n-|S|-1)} w(S) & \textit{if} \quad |S| < n-k. \end{cases} \tag{13}$$

Then, based on Lemma 5, the deterioration function $r(\delta)$ of (2) for MNC is

$$r(\delta) = \begin{cases} \frac{\sigma^2}{\delta^2} & \text{if} \quad \delta \geq \sigma; \\ \frac{(1-\sigma)^2}{(1-\delta)^2} & \text{if} \quad 1/2 \leq \delta < \sigma; \\ \frac{(1-\sigma)^2}{\delta^2} & \text{if} \quad 1-\sigma \leq \delta < 1/2; \\ \frac{\sigma^2}{(1-\delta)^2} & \text{if} \quad \delta < 1-\sigma. \end{cases}$$

To analyze $\mathrm{E}[w(S)]/w^*$ for MNC, first note

$$w(S) = \frac{1}{2} \sum_{1 \le i < j \le n} w_{ij} \left(1 + \hat{x}_i \hat{x}_j\right).$$

Let

$$\begin{array}{rll} \alpha = \alpha(\theta) := \min & \frac{1+\frac{2}{\pi}\arcsin(\theta z + (1-\theta)\kappa^2)}{1+z} \\ \text{subject to} & -1 < z \le 1. \end{array} \tag{14}$$

Then, we have the following result resembling Lemma 3:

$$\mathrm{E}[w(S)] \ge \alpha \cdot w^{SDP} \ge \alpha \cdot w^*.$$

Now we can go ahead compute $R(\eta, \gamma, \theta)$ and R_k^* from (4) and (5). Again, $R(\eta, \gamma, \theta)$ can be constructed from the four cases similar to MC:

Case1. $\sigma \le \delta \le 1$:

$$R_1(\eta, \gamma, \theta) = \begin{cases} \sigma^2 \left(-\frac{(\eta\gamma+\gamma)^2}{4\mu_{\eta,\gamma}} + \gamma\right) & \text{if } \sigma \le \frac{2\mu_{\eta,\gamma}}{\gamma+\eta\gamma} \le 1; \\ \min\{\lambda_{\eta,\gamma}(\sigma), \sigma^2 \lambda_{\eta,\gamma}(1)\} & \text{otherwise;} \end{cases}$$

Case2. $\frac{1}{2} \le \delta \le \sigma$:

$$R_2(\eta, \gamma, \theta) = \begin{cases} (1-\sigma)^2 \left(-\frac{(-\eta\gamma+\gamma)^2}{4(\mu_{\eta,\gamma}-\eta\gamma)} + \gamma\right) & \text{if } 1-\sigma \le \frac{2(\mu_{\eta,\gamma}-\eta\gamma)}{\gamma-\eta\gamma} \le \frac{1}{2}, \\ \min\{\lambda_{\eta,\gamma}(\sigma), 4(1-\sigma)^2 \lambda_{\eta,\gamma}(1/2)\} & \text{otherwise;} \end{cases}$$

Case3. $1-\sigma \le \delta \le \frac{1}{2}$:

$$R_3(\eta, \gamma, \theta) = \begin{cases} (1-\sigma)^2 \left(-\frac{(\eta\gamma+\gamma)^2}{4\mu_{\eta,\gamma}} + \gamma\right) & \text{if } 1-\sigma \le \frac{2\mu_{\eta,\gamma}}{\gamma+\eta\gamma} \le \frac{1}{2}, \\ \min\{\lambda_{\eta,\gamma}(1-\sigma), 4(1-\sigma)^2 \lambda_{\eta,\gamma}(1/2)\} & \text{otherwise;} \end{cases}$$

Case4. $0 \le \delta \le 1-\sigma$:

$$R_4(\eta, \gamma, \theta) = \begin{cases} \sigma^2 \left(-\frac{(-\eta\gamma+\gamma)^2}{4(\mu_{\eta,\gamma}-\eta\gamma)} + \gamma\right) & \text{if } \sigma \le \frac{2(\mu_{\eta,\gamma}-\eta\gamma)}{\gamma-\eta\gamma} \le 1, \\ \min\{\lambda_{\eta,\gamma}(1-\sigma), \sigma^2 \lambda_{\eta,\gamma}(0)\} & \text{otherwise.} \end{cases}$$

Again,

$$R(\eta, \gamma, \theta) = \min\{R_1(\eta, \gamma, \theta), R_2(\eta, \gamma, \theta), R_3(\eta, \gamma, \theta), R_4(\eta, \gamma, \theta)\}.$$

We have generated Table 3, which shows $R(\eta, \gamma, \theta)$ for $0.5n \le k \le 0.98n$. Note that the true performance ratio R_k^* for MC should be greater than $R(\eta, \gamma, \theta)$ of the table, and $R(\eta, \gamma, \theta)$ for $0.5n \ge k \ge 0.02n$ are identical to those displayed here.

$\sigma = \frac{k}{n}$	θ	α	η	γ	$R(\eta, \gamma, \theta)$
0.50	0.75	0.7699	0	19.91	0.6026
0.52	0.79	0.7899	0	16.11	0.6451
0.54	0.87	0.8274	0	8.01	0.6353
0.56	0.91	0.8446	0	5.01	0.6013
0.58	0.95	0.8606	0	3.51	0.5587
0.60	0.77	0.7844	0.24	13.21	0.5258
0.62	0.69	0.7503	0.26	18.51	0.5049
0.64	0.71	0.7618	0.30	17.71	0.4828
0.66	0.67	0.7485	0.32	19.81	0.4586
0.68	0.71	0.7688	0.38	16.11	0.4320
0.70	0.73	0.7813	0.42	15.41	0.4030
0.72	0.75	0.7943	0.48	13.41	0.3724
0.74	0.75	0.7992	0.52	13.01	0.3397
0.76	0.75	0.8046	0.56	12.61	0.3053
0.78	0.03	0.6085	0.50	18.61	0.2746
0.80	0.02	0.6216	0.51	21.20	0.2547
0.82	0.02	0.6385	0.55	21.00	0.2305
0.84	0.02	0.6569	0.59	21.20	0.2057
0.86	0	0.6735	0.63	21.60	0.1789
0.88	0	0.6960	0.69	20.80	0.1498
0.90	0	0.7211	0.71	23.90	0.1235
0.92	0	0.7493	0.71	30.80	0.0986
0.94	0	0.7820	0.71	42.50	0.0738
0.96	0	0.8213	0.70	71.60	0.0509
0.98	0	0.8731	0.76	88.80	0.0191

Table 3: Worst-case performance ratios of the SDP-MNC algorithm

7 Final Remarks

If the size requirement k is close to n, we have also used $\kappa = sin(\frac{\pi}{2} \cdot (2\frac{k}{n} - 1))$ in matrix P of the the balance rounding procedure, which results in an improvement for the worst-case performance ratio.

In practice, after the semidefinite solution $\bar{X}$ is computed, one can do one dimensional search of θ in the rounding algorithm to yield the best possible $\tilde{S}$.

References

[1] A. A. Ageev and M. I. Sviridenko, "Approximation algorithms for maximum coverage and max cut with given sizes of parts", IPCO, G. Cornuejols, R. E. Burkard, G. J. Woeginger (Eds.), 17-30, 1999. Note

[2] G. Anderson, "An Approximation Algorithm for Max p-Section", Proceedings of *16th International Symposium on Theoretical Aspects of Computer Science*, volume 1563 of Lecture Notes in Computer Science, 237-247. Springer-Verlag, Berlin, 1999.

[3] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, "Greedily finding a dense subgraph," in *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT)*, Lecture Notes in Computers Science, 1097, pages 136-148, Springer-Verlag, 1996.

[4] D. Bertsimas and Y. Ye, "Semidefinite relaxations, multivariate normal distributions, and order statistics," *Handbook of Combinatorial Optimization (Vol. 3)*, D.-Z. Du and P.M. Pardalos (Eds.) pp. 1-19, (1998 Kluwer Academic Publishers).

[5] U. Feige and M. X. Goemans, "Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT," in *Proceedings of the Third Israel Symposium on Theory of Computing and Systems*, (1995) pp. 182-189.

[6] U. Feige and M. Langberg, "Approximation Algorithms for Maximization Problems arising in Graph Partitioning" manuscript and Ph. D. thesis, 1999.

[7] U. Feige and M. Seltser, "On the densest k-subgraph problem," Technical report, Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot, September 1997.

[8] U. Feige, G. Kortsarz and D. Peleg, "The Dense k-Subgraph problem," to appear in *Algorithmica*.

[9] A. Frieze and M. Jerrum, "Improved approximation algorithms for max k-cut and max bisection," *Proc. 4th IPCO Conference* (1995) 1-13.

[10] M. X. Goemans and D. P. Williamson, " Improved approximation algorithms for Maximum Cut and Satisfiability problems using semidefinite programming," *Journal of ACM* 42 (1995) 1115–1145.

[11] G. Kortsarz and D. Peleg, "On choosing a dense subgraph," in *34th FOCS*, (1993) 692-701.

[12] S. Mahajan and H. Ramesh, "Derandomizing semidefinite programming based approximation algorithms," in *36th FOCS*, (1995) pp. 162-169.

[13] Y. E. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optimization Methods and Software* 9 (1998) 141-160.

[14] A. Srivastav and K. Wolf, "Finding dense subgraphs with semidefinite programming," in *Approximation Algorithms for Combinatorial Optimization*, K. Jansen and J. Rolim (Eds.) (1998) pp. 181-191.

[15] Y. Ye, "A .699-approximation algorithm for Max-Bisection," *Manuscript*, March 1999.

[16] Y. Ye, "Approximating quadratic programming with bound and quadratic constraints," *Mathematical Programming* 84 (1999) 219-226.

[17] Y. Ye and J. Zhang, "Approximation of Dense-$\frac{n}{2}$-Subgraph and the complement of Min-Bisection," *Manuscript*, December 1999.

[18] U. Zwick, "Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint," In *9th SODA* (1998) 201-210.

[19] U. Zwick, "Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to Max-Cut and other problems," in *31th STOC* (1999) 679-687.