Improved Approximation Algorithms for Maximum Graph Partitioning Problems

Gerold Jäger and Anand Srivastav

Mathematisches Seminar,
Christian-Albrechts-Universität Kiel,
Christian-Albrechts-Platz 4, D-24118 Kiel, Germany
gej@numerik.uni-kiel.de, asr@numerik.uni-kiel.de

Extended Abstract

Abstract. In this paper we improve the analysis of approximation algorithms based on semidefinite programming for the maximum graph partitioning problems MAX-k-CUT, MAX-k-UNCUT, MAX-k-DIRECTED-CUT, MAX-k-DIRECTED-UNCUT, MAX-k-DENSE-SUBGRAPH, and MAX-k-VERTEX-COVER. It was observed by Han, Ye, Zhang (2002) and Halperin, Zwick (2002) that a parameter-driven random hyperplane can lead to better approximation factors than obtained by Goemans and Williamson (1994). Halperin and Zwick could describe the approximation factors by a mathematical optimization problem for the above problems for $k = \frac{n}{2}$ and found a choice of parameters in a heuristic way. The innovation of this paper is twofold. First, we generalize the algorithm of Halperin and Zwick to cover all cases of k, adding some algorithmic features. The hard work is to show that this leads to a mathematical optimization problem for an optimal choice of parameters. Secondly, as a key-step of this paper we prove that a sub-optimal set of parameters is determined by a *linear program*. Its optimal solution computed by CPLEX leads to the desired improvements. In this fashion a more systematic analysis of the semidefinite relaxation scheme is obtained which leaves room for further improvements.

1 Introduction

For a directed graph $G = (V, E)$ with $|V| = n$ and a non-negative weight $\omega_{i,j}$ on each edge $(i, j) \in E$, such that $\omega_{i,j}$ is not identically zero on all edges, and for $0 < \sigma := \frac{k}{n} < 1$ we consider the following problems:

a) MAX-k-CUT: determine a subset $S \subseteq V$ of k vertices such that the total weight ω^* of the edges connecting S and $V \setminus S$ or connecting $V \setminus S$ and S is maximized. [1]

[1] In some literature MAX-k-CUT denotes the problem of partitioning the set of vertices into subsets $S_1, \cdots, S_k$, so that the total weight of the edges connecting S_i and S_j for $1 \leq i \neq j \leq k$ is maximized.

b) MAX-k-UNCUT: determine a subset $S \subseteq V$ of k vertices such that the total weight ω^* of the edges of the subgraphs induced by S and induced by $V \setminus S$ is maximized.

c) MAX-k-DIRECTED-CUT: determine a subset $S \subseteq V$ of k vertices such that the total weight ω^* of the edges connecting S and $V \setminus S$ is maximized.

d) MAX-k-DIRECTED-UNCUT: determine a subset $S \subseteq V$ of k vertices such that the total weight ω^* of the edges of the subgraphs induced by S and induced by $V \setminus S$ plus the edge weights connecting $V \setminus S$ and S is maximized.

e) MAX-k-DENSE-SUBGRAPH: determine a subset $S \subseteq V$ of k vertices such that the total weight ω^* of the edges of the subgraph induced by S is maximized.

f) MAX-k-VERTEX-COVER: determine a subset $S \subseteq V$ of k vertices such that the total weight ω^* of the edges touching S is maximized.

As all these problems are NP-hard, we are interested in approximating the optimal solution to these problems within a factor of $0 \leq \varrho \leq 1$. Goemans and Williamson [10] showed in their pioneer paper that via the semidefinite programming (SDP) relaxation an approximation factor of 0.878 can be proved for the MAX-CUT problem. Stimulated by their work, many authors have considered only one or two of the six problems above (see [2] for MAX-k-CUT, [9] and [15] for MAX-$\frac{n}{2}$-CUT, [1] for MAX-k-DIRECTED-CUT, [16] for MAX-$\frac{n}{2}$-UNCUT and MAX-$\frac{n}{2}$-DENSE-SUBGRAPH, [3], [7], [8] and [14] for MAX-k-DENSE-SUBGRAPH and [2] for MAX-k-VERTEX-COVER). Feige and Langberg [5] improved on known special and global approximation factors for the four undirected problems with some new techniques based on semidefinite programming. Their paper contains also a nice summary of known results. Han, Ye and Zhang [12] also applied semidefinite programming to these four problems and in most cases they managed to obtain better approximation factors than previously known. Halperin and Zwick [11] used more general methods for the balanced version ($\sigma = \frac{1}{2}$) and in this case achieved substantially improved approximation factors for all six problems above.

In this paper we give an algorithm for the problems a) - f), generalizing the approach of Halperin and Zwick, resp. of Han, Ye and Zhang by introducing new parameters which enlarge the region of the semidefinite programming relaxation. This gives a new version of the semidefinite relaxation scheme (Algorithm *Graph Partitioning*, section 2, page 4). In Theorem 1, we show that the expectation of the approximation factors depend on a set of parameters, which are used in the algorithm. The key observation is that a sub-optimal choice of these parameters can be determined by a finite *linear program* (section 4). By discretizing the other parameters, we finally obtain a choice leading to improvements over known approximation guarantees. Here are some examples for which the improvement is significant (comprehensive tables can be found in section 5).

Problem	σ	Prev.	Our Method
MAX-k-CUT	0.3	0.527	0.567
MAX-k-UNCUT	0.4	0.5258	0.5973
MAX-k-DIRECTED-CUT	0.5	0.644	0.6507
MAX-k-DIRECTED-UNCUT	0.5	0.811	0.8164
MAX-k-DENSE-SUBGRAPH	0.2	0.2008	0.2664
MAX-k-VERTEX-COVER	0.6	0.8453	0.8784

In summary, we see that our technique of combining the analysis of the random hyperplane with mathematical programming leads to improvements over many previously known approximation factors for the maximization problems considered in this paper. This shows that a more systematic analysis of the semidefinite relaxation scheme gives better approximation guarantees and opens room for further improvements, if better methods for choosing an optimal parameter set can be designed.

2 The Algorithm

For $S \subseteq V$ the set of edges E can be divided by $E = S_1 \dot\cup S_2 \dot\cup S_3 \dot\cup S_4$, where $S_1 = \{(i,j) \,|\, i,j \in S\}, S_2 = \{(i,j) \,|\, i \in S, j \in V \setminus S\}, S_3 = \{(i,j) \,|\, i \in V \setminus S, j \in S\}, S_4 = \{(i,j) \,|\, i,j \in V \setminus S\}$. As we will see, we distinguish the six problems MAX-k-CUT, MAX-k-UNCUT, MAX-k-DIRECTED-CUT, MAX-k-DIRECTED-UNCUT, MAX-k-DENSE- SUBGRAPH, MAX-k-VERTEX-COVER by four $\{0,1\}$ parameters a_1, a_2, a_3, a_4. All these problems maximize some of the four edge classes S_1, S_2, S_3, S_4.
For $i = 1,2,3,4$ we define a_i as 1, if the problem maximizes the edge weights of S_i, and 0 otherwise. The following values a_1, a_2, a_3, a_4 lead to the specific problems:

Problem	a_1	a_2	a_3	a_4
MAX-k-CUT	0	1	1	0
MAX-k-UNCUT	1	0	0	1
MAX-k-DIRECTED-CUT	0	1	0	0
MAX-k-DIRECTED-UNCUT	1	0	1	1
MAX-k-DENSE-SUBGRAPH	1	0	0	0
MAX-k-VERTEX-COVER	1	1	1	0

For $F \subseteq E$ we define $\omega(F) = \sum_{(i,j)\in F} \omega_{ij}$ and for $S \subseteq V$:

$$\omega_{a_1,a_2,a_3,a_4}(S) := a_1\omega(S_1) + a_2\omega(S_2) + a_3\omega(S_3) + a_4\omega(S_4).$$

The optimization problem considered in this paper is the following.

General Maximization Problem:

$$\max_{S\subseteq V,|S|=k} \omega_{a_1,a_2,a_3,a_4}(S) \tag{1}$$

Let $\mathrm{OPT}(a_1,a_2,a_3,a_4,\sigma)$ be the value of an optimal solution of (1). Our aim is to design a randomized polynomial-time algorithm which returns a solution of value at least $\varrho\cdot\mathrm{OPT}(a_1,a_2,a_3,a_4,\sigma)$, where $\varrho=\varrho(a_1,a_2,a_3,a_4,\sigma)$ is the so-called approximation factor with $0\le\varrho\le1$. In fact, we will show that the expected value of ϱ is large.
In the algorithm we give a formulation of the general maximization problem (1) as a semidefinite program, generalizing Halperin and Zwick [11].

Algorithm *Graph Partitioning*
Input: A weighted directed graph $G=(V,E)$ with $|V|=n$, $0<\sigma<1$ and parameters $0\le\theta,\vartheta,\nu\le1$ and $-1\le\kappa\le1$, a maximum graph partitioning problem with parameters a_1,a_2,a_3,a_4.
Output: A set S of k vertices with large $\omega_{a_1,a_2,a_3,a_4}(S)$.

1. *Relaxation* We solve the following semidefinite program:
   Maximize $\sum_{1\le i\ne j\le n}\frac{1}{4}\omega_{ij}\big[(a_1+a_2+a_3+a_4)+(a_1+a_2-a_3-a_4)X_{i0}$
   $+(a_1-a_2+a_3-a_4)X_{j0}+(a_1-a_2-a_3+a_4)X_{ij}\big]$
   with the optimal value ω^* subject to the constraints
   (a) $\sum_{i=1}^n X_{i0}=2k-n$
   (b) $\sum_{1\le i,j\le n}X_{ij}=(2k-n)^2$
   (c) $X_{ii}=1$ for $i=0,1,\cdots,n$
   (d) $X\in\mathbb{R}^{n+1,n+1}$ is positive semidefinite and symmetric
   (e) $X_{ij}+X_{il}+X_{jl}\ge-1$ for $0\le i,j,l\le n$
   (f) $X_{ij}-X_{il}-X_{jl}\ge-1$ for $0\le i,j,l\le n$
   From b), c) and d) it follows:
   (g) $\sum_{1\le i<j\le n}X_{ij}=\frac{1}{2}\left((2k-n)^2-n\right)$

   (This is the same semidefinite program like in [11] with new constraint (a) and generalized constraint (b).)

We repeat the following four steps polynomially often and output the best subset.

2. *Randomized Rounding*
   - Choose parameters $0\le\theta,\vartheta\le1$ and $-1\le\kappa\le1$ (note that for every problem and for each σ we choose different parameters).
   - Choose a positive semidefinite symmetric matrix $Y=Z^TZ\in\mathbb{R}^{n+1,n+1}$, depending on θ,ϑ,κ as follows:

Put $Y := \theta L + (1-\theta)P$, where we define $L = (l_{ij})_{0\le i,j\le n}$ and $P = (p_{ij})_{0\le i,j\le n}$ by

$$(l_{ij})_{0\le i,j\le n} = \begin{cases} 1 & \text{for } i=j \\ \vartheta X_{0i} & \text{for } i\ne 0, j=0 \\ \vartheta X_{0j} & \text{for } i=0, j\ne 0 \\ \vartheta X_{ij} \quad \text{or} \quad X_{ij} \quad \text{or} \quad \vartheta^2 X_{ij} & \text{for } 1\le i\ne j\le n \end{cases}$$

$$(p_{ij})_{0\le i,j\le n} = \begin{cases} 1 & \text{for } i=j \\ \kappa & \text{for } i=0, j\ne 0 \vee i\ne 0, j=0 \\ \kappa, \text{ if } \kappa\ge 0 \quad \text{or} \quad 1 \quad \text{or} \quad \kappa^2 & \text{for } 1\le i\ne j\le n \end{cases}$$

We can write the non diagonal elements of Y for $0\le i\ne j\le n$ as

$$Y_{ij} = \begin{cases} d_1 X_{ij} + e_1, & \text{if } i=0 \vee j=0 \\ d_2 X_{ij} + e_2, & \text{otherwise} \end{cases}$$

with $d_1 = \theta\vartheta$; $e_1 = (1-\theta)\kappa$; $d_2 = \theta\vartheta, \theta, \theta\vartheta^2$; $e_2 = (1-\theta)\kappa$, (if $\kappa\ge 0$), $1-\theta, (1-\theta)\kappa^2$. Hence: $-1\le e_1\le 1$; $0\le d_1, d_2, e_2\le 1$.
(It is easy to show that Y is a positive semidefinite symmetric matrix.)

- We choose $\bar{u}$ with $\bar{u}_i \in N(0,1)$ for $i = 0,1,\cdots,n$ and let $u = Z\bar{u}$.
- For $i = 1,\cdots,n$ let $\hat{x}_i = 1$, if $u_i\ge 0$ and -1 otherwise and let $S = \{i\ge 1 \mid \hat{x}_i = 1\}$ (see [4]).
  (As special cases we get previously used positive semidefinite matrices: $\vartheta = 1$ and case 3 for P in [12]; $\theta = 1$ and case 1 for L in [11], which is called *outward rotation*.)

3. *Linear Randomized Rounding*
   - Choose a parameter $0\le\nu\le 1$ (again for every problem and for each σ we choose a different parameter).
   - With probability $0\le\nu\le 1$ we overrule the choice of S made above, and for each $i\in V$, put i into S, independently, with probability $(1+X_{i0})/2$ and into $V\setminus S$ otherwise.
4. *Size Adjusting*
   a) If the problem is symmetric (MAX-k-CUT or MAX-k-UNCUT):
   - If $k\le |S| < \frac{n}{2}$, we remove uniformly at random $|S|-k$ vertices from S.
   - If $|S| < k$, we add uniformly at random $k-|S|$ vertices to S.
   - If $\frac{n}{2}\le |S| < n-k$, we add uniformly at random $n-k-|S|$ vertices to S.
   - If $|S|\ge n-k$, we remove uniformly at random $|S|-n+k$ vertices from S.

   b) If the problem is not symmetric:
   - If $|S|\ge k$, we remove uniformly at random $|S|-k$ vertices from S.
   - If $|S| < k$, we add uniformly at random $k-|S|$ vertices to S.
5. *Flipping* (only for MAX-$\frac{n}{2}$-DIRECTED-CUT, MAX-$\frac{n}{2}$-DIRECTED-UNCUT, MAX-$\frac{n}{2}$-DENSE-SUBGRAPH, and MAX-$\frac{n}{2}$-VERTEX-COVER)
   If $\omega_{a_1,a_2,a_3,a_4}(V\setminus S) > \omega_{a_1,a_2,a_3,a_4}(S)$, we output $V\setminus S$, otherwise S.

3 Computation of the Approximation Factors

3.1 Main Result

The main results are shown in the tables containing the approximation factors for the different problems. Nevertheless, let us state them also in a formal way:

Theorem 1. *(**Main Theorem**) The expected ratio $\omega_{a_1,a_2,a_3,a_4}(S)/OPT$ of the approximation factors for the problems MAX-k-CUT, MAX-k-UNCUT, MAX-k-DIRECTED-CUT, MAX-k-DIRECTED-UNCUT, MAX-k-DENSE-SUBGRAPH and MAX-k-VERTEX-COVER is bounded from below by the minimum of the solutions of some linear programs. Solving these linear programs lead to the approximation factors shown in the tables of section 5.*

We denote the sets S after the steps 2,3,4,5 by $S', S'', S''', S''''(= S)$ and define $\delta := \frac{|S''|}{n}$. We want to compute $\rho := \frac{\omega_{a_1,a_2,a_3,a_4}(S)}{\omega^*}$. For $x_1 \in \mathbb{R}$, $x_2 \in \mathbb{R}_0^+$ we consider the function of Han, Ye, Zhang [12]:

$$y(x_1, x_2) = \frac{\omega_{a_1,a_2,a_3,a_4}(S'')}{\omega^*} + x_1 \frac{|S''|}{n} + x_2 \frac{|S''|(n - |S''|)}{n^2}$$

(Halperin, Zwick [11] consider this function only for $x_1 \in \mathbb{R}_0^-$. The case $x_2 < 0$ could also be considered, but as it does not lead to any progress, we omit it.) The first term $\frac{\omega_{a_1,a_2,a_3,a_4}(S'')}{\omega^*}$ gives the value of the partition S'' after step 3 in relation to the optimal value, $\frac{|S''|}{n}$ and $\frac{|S''|(n-|S''|)}{n^2}$ measure the closeness of the partition to the required size. x_1, x_2 are parameters which depend on the specific problem and σ. They are chosen so that the analysis leads to good approximation factors. For analyzing this function, we have to estimate the expected values of the three terms. This is done in the main lemma. Its proof is given in the full paper.

Lemma 1. *(**Main Lemma**) For $n \to \infty$ there are constants $\alpha, \beta^+, \beta^-, \gamma$ with:*

a) $E\left[\frac{\omega_{a_1,a_2,a_3,a_4}(S'')}{\omega^*}\right] \geq \alpha(\theta, \vartheta, \kappa, \nu)$

b) $E\left[\frac{|S''|}{n}\right] \geq \beta^+(\sigma, \theta, \vartheta, \kappa, \nu)$

c) $E\left[\frac{|S''|}{n}\right] \leq \beta^-(\sigma, \theta, \vartheta, \kappa, \nu)$

d) $E\left[\frac{|S''|(n-|S''|)}{n^2}\right] \geq \gamma(\sigma, \theta, \vartheta, \kappa, \nu)$

3.2 Proof of Theorem 1

For $x_1 \geq 0$ define $\beta(\sigma, \theta, \vartheta, \kappa, \nu)$ as $\beta^+(\sigma, \theta, \vartheta, \kappa, \nu)$ and otherwise as $\beta^-(\sigma, \theta, \vartheta, \kappa, \nu)$. As we repeat the steps 2 and 3 of the algorithm polynomially often, the function $z(x_1, x_2)$ is its expectation value, up to a factor of $1 - \epsilon$, which can be neglected. By Lemma 1 we get:

$$
\begin{aligned}
&\frac{\omega_{a_1,a_2,a_3,a_4}(S'')}{\omega^*} + x_1\frac{|S''|}{n} + x_2\frac{|S''|(n-|S''|)}{n^2} \\
&\geq E\left[\frac{\omega_{a_1,a_2,a_3,a_4}(S'')}{\omega^*}\right] + E\left[x_1\frac{|S''|}{n}\right] + E\left[x_2\frac{|S''|(n-|S''|)}{n^2}\right] \\
&\geq \alpha(\theta,\vartheta,\kappa,\nu) + x_1\beta(\sigma,\theta,\vartheta,\kappa,\nu) + x_2\gamma(\sigma,\theta,\vartheta,\kappa,\nu)
\end{aligned}
$$

and so

$$
\begin{aligned}
&\frac{\omega_{a_1,a_2,a_3,a_4}(S'')}{\omega^*} \\
&\geq\ \alpha(\theta,\vartheta,\kappa,\nu) + x_1(\beta(\sigma,\theta,\vartheta,\kappa,\nu)-\delta) + x_2(\gamma(\sigma,\theta,\vartheta,\kappa,\nu)-\delta(1-\delta)) \\
&=: h(\delta,\sigma,\theta,\vartheta,\kappa,\nu,x_1,x_2) \qquad (2)
\end{aligned}
$$

With $\lambda_i\ :=\ \frac{\omega(S_i'')}{\omega^*}$ for $i=1,2,3,4$ it is not difficult to show:

$$
\begin{aligned}
a_1\lambda_1 + a_2\lambda_2 + a_3\lambda_3 + a_4\lambda_4 &\geq h(\delta,\sigma,\theta,\vartheta,\kappa,\nu,x_1,x_2) \\
\lambda_1+\lambda_2+\lambda_3+\lambda_4 &\geq 1
\end{aligned}
$$

Define $M_1(p) := \begin{pmatrix} p^2 & 0 & 0 & 0 \\ p(1-p) & p & 0 & 0 \\ p(1-p) & 0 & p & 0 \\ (1-p)^2 & 1-p & 1-p & 1 \end{pmatrix}$, $M_2(q) := \begin{pmatrix} 1 & 1-q & 1-q & (1-q)^2 \\ 0 & q & 0 & q(1-q) \\ 0 & 0 & q & q(1-q) \\ 0 & 0 & 0 & q^2 \end{pmatrix}$.

Furthermore $M(\delta,\sigma)\ :=\ \begin{cases} M_1\left(\frac{\sigma}{\delta}\right), & \text{if}\quad \sigma\leq\delta<\frac{1}{2} \\ M_2\left(\frac{1-\sigma}{1-\delta}\right), & \text{if}\quad 0\leq\delta<\sigma \\ M_2\left(\frac{\sigma}{1-\delta}\right), & \text{if}\quad \frac{1}{2}\leq\delta<1-\sigma \\ M_1\left(\frac{1-\sigma}{\delta}\right), & \text{if}\quad 1-\sigma\leq\delta\leq 1 \end{cases}$ in the symmetric case and

$M(\delta,\sigma)\ :=\ \begin{cases} M_1\left(\frac{\sigma}{\delta}\right), & \text{if}\quad \sigma\leq\delta\leq 1 \\ M_2\left(\frac{1-\sigma}{1-\delta}\right), & \text{if}\quad 0\leq\delta<\sigma \end{cases}$ in the asymmetric case.

Then it is straightforward to show:

$$
\begin{aligned}
z := E&\left[\frac{\omega_{a_1,a_2,a_3,a_4}(S''')}{\omega^*}\right] \\
=&\ (a_1\ a_2\ a_3\ a_4)\cdot M(\delta,\sigma)\cdot(\lambda_1\ \lambda_2\ \lambda_3\ \lambda_4)^T \qquad (3)\\
=:&\ f_{a_1,a_2,a_3,a_4}(\delta,\sigma,\lambda_1,\lambda_2,\lambda_3,\lambda_4) \qquad (4)
\end{aligned}
$$

The expected approximation factor of Algorithm *Graph Partitioning* thus is:

$$\min_{0\le\delta\le1}\left[\begin{array}{c}\min\quad z\\ \text{s.t.}\\ a_1\lambda_1+a_2\lambda_2+a_3\lambda_3+a_4\lambda_4\ge h(\delta,\sigma,\theta,\vartheta,\kappa,\nu,x_1,x_2)\\ \lambda_1+\lambda_2+\lambda_3+\lambda_4\ge 1\\ 0\le\lambda_1,\lambda_2,\lambda_3,\lambda_4\\ z\ge f_{a_1,a_2,a_3,a_4}(\delta,\sigma,\lambda_1,\lambda_2,\lambda_3,\lambda_4)\\ \text{[for MAX-}\tfrac{n}{2}\text{-DC, MAX-}\tfrac{n}{2}\text{-DU, MAX-}\tfrac{n}{2}\text{-DS, MAX-}\tfrac{n}{2}\text{-VC:}\\ z\ge f_{a_4,a_3,a_2,a_1}(\delta,\sigma,\lambda_1,\lambda_2,\lambda_3,\lambda_4)]\end{array}\right] \tag{5}$$

Note that for fixed δ,σ and constants $\theta,\vartheta,\kappa,\nu,x_1,x_2$ the last inner minimization problem is a linear program in the variables $z,\lambda_1,\lambda_2,\lambda_3,\lambda_4$.

4 The Optimization Algorithm

By Theorem 1 and (5), the expected approximation factor for the general maximization problem (1) is z, where $z=z(\delta,\sigma,\theta,\vartheta,\kappa,\nu,x_1,x_2)$ is a function depending on the parameters $\delta\in[0,1],\sigma\in(0,1),\theta,\vartheta,\nu\in[0,1],\kappa\in[-1,1],x_1\in\mathbb{R},x_2\in\mathbb{R}_0^+$ in a complicated way. A polynomial-time algorithm for an optimal choice of all parameters in (5) is not known. Thus we choose a hierarchical approach.

For this we need the following theorem which shows that we can solve (5) by hand for arbitrary, but fixed $\delta,\sigma,\theta,\vartheta,\kappa,\nu$. Note that for a previous approximation factor ϱ and a candidate z for a new approximation factor, we would like to show $z\ge\varrho$ for a suitable set of parameters.

Theorem 2. *Let $f_{a_1,a_2,a_3,a_4}(\lambda_1,\lambda_2,\lambda_3,\lambda_4)=b_1\lambda_1+b_2\lambda_2+b_3\lambda_3+b_4\lambda_4$ with suitable b_1,b_2,b_3,b_4 (note that according to (3) and (4), (b_1,b_2,b_3,b_4) is given by the vector $(a_1,a_2,a_3,a_4)\cdot M(\delta,\sigma)$). Furthermore let $v:=\min_{l\in\{1,2,3,4\}}\{b_l\}, w:=\min_{l\in\{i\in I\,|\,a_i=1\}}\{b_l\}$.*
Then for all maximization problems except MAX-$\frac{n}{2}$-DIRECTED-CUT, MAX-$\frac{n}{2}$-DIRECTED-UNCUT, MAX-$\frac{n}{2}$-DENSE-SUBGRAPH, MAX-$\frac{n}{2}$-VERTEX-COVER it holds:
a) *(5) has the solution*

$$z=\begin{cases} w\cdot h, & \text{if } h\ge 1\\ v\cdot(1-h)+w\cdot h, & \text{if } 0\le h<1\\ v, & \text{if } h<0\end{cases}$$

b) *For $v=w$, the condition $z\ge\varrho$ is equivalent to*

$$h\ge\max\left\{\frac{\varrho}{w},1\right\} \tag{6}$$

c) *For $v\ne w$, the condition $z\ge\varrho$ equivalent to*

$$h\ge\min\left\{\frac{\varrho}{w},\frac{\varrho-v}{w-v}\right\} \tag{7}$$

For the proof of Theorem 2 we refer to the full paper.

Remark 1. We can derive similar expressions for z for the four remaining maximization problems leading to inequalities for h. Again we refer to the full paper.

An Algorithm for Parameter Setting

Our approach consists of three main steps. Let us consider σ as fixed.

1. Fixing the right-hand side.
   Let ϱ_0 be the previously known best approximation factor for the problem in the literature [11], [12], [5], and put $\varrho := \varrho_0 + k \cdot 0.0001$ for $k = 0, 1, \cdots$. We would like to prove $z \geq \varrho$ for a k as large as possible.
2. The linear program $LP(\Delta)$.
   For the moment let us fix the parameters $\theta, \vartheta, \kappa, \nu$ and consider them as constants. Let $h = h(\delta, \sigma, \theta, \vartheta, \kappa, \nu, x_1, x_2)$ be the function defined in (2). Since h is a linear function in x_1 and x_2 due to (2), we may write $h(x_1, x_2) = f_1(\delta)x_1 + f_2(\delta)x_2 + f_3$, suppressing the dependence of h on $\theta, \vartheta, \kappa, \nu$, writing $h(x_1, x_2)$ instead of $h(\delta, \sigma, \theta, \vartheta, \kappa, \nu, x_1, x_2)$, and putting the dependence of h on δ into the coefficients $f_1(\delta)$ and $f_2(\delta)$.
   Since by Theorem 2 a), z is only piecewise linear in h, $z \geq \varrho$ is not a linear inequality in h. But by Theorem 2 b), c) $z \geq \varrho$ is equivalent to a linear inequality in x_1 and x_2.
   Still, the dependence on δ is an obstacle. We choose a discretization of $[0, 1]$ for the δ's, i.e. we define $\Delta := \left\{k \cdot \frac{1}{10^l}, k = 0, 1, \cdots, 10^l\right\}$ for a sufficiently large $l \in \mathbb{N}$. The inequalities in (6) and (7), respectively for all $\delta \in \Delta$ form a finite linear program in the variables x_1 and x_2 which we denote by $LP(\Delta)$.
3. Discretization of the other parameters.
   Whether $LP(\Delta)$ is solvable or not depends on the choice of the parameters θ, ϑ, κ, ν. We discretize the ranges of these parameters in finitely many points. For $\theta, \vartheta, \nu \in [0, 1], \kappa \in [-1, 1]$ we take the discretization of both intervals with step size $\frac{1}{10}$ (for some cases we try even the finer discretization with step size $\frac{1}{100}$). We consider all possible values of $(\theta, \vartheta, \kappa, \nu)$ in this discretization and denote it by the parameter set $\mathcal{P}$. We test about 250,000 possibilities of tuples $(\theta, \vartheta, \kappa, \nu)$.

The algorithm for finding the parameters $\theta, \vartheta, \kappa, \nu, x_1, x_2$ and a good approximation factor ϱ is the following.

Algorithm *Parameter Set*

1. Choose ϱ as the best previously known approximation factor ϱ_0.
2. Choose $(\theta, \vartheta, \kappa, \nu)$ from the parameter set $\mathcal{P}$.
3. Given ϱ, solve $LP(\Delta)$ in the variables x_1 and x_2 by the simplex algorithm using CPLEX.
4. a) If $LP(\Delta)$ is solvable, increase ϱ by 0.0001 and goto 3.
   b) If $LP(\Delta)$ is not solvable and if not all parameters are tested, goto 2.
5. Output ϱ.

Remark 2. Note that x_1, x_2 live in a large range, i.e. $x_1 \in \mathbb{R}$ and $x_2 \in \mathbb{R}_0^+$, while $\theta, \vartheta, \nu, \kappa$ are only in the relatively small ranges $[-1, 1]$ and $[0, 1]$, so that we have optimized the two most difficult parameters.

5 The Final Approximation Factors

We state the results in the following tables.

	MAX-k-C		MAX-k-UC		MAX-k-DC		MAX-k-DU	
σ	Prev.	Our Meth.	Prev.	Our Meth.	Prev.	Our Meth.	Prev.	Our Meth.
0.02	0.5	0.5	0.9608	0.9608	0.5	0.1439	–	0.9804
0.04	0.5	0.5	0.9232	0.9232	0.5	0.18	–	0.9616
0.06	0.5	0.5	0.8872	0.8872	0.5	0.2211	–	0.9436
0.08	0.5	0.5	0.8528	0.8528	0.5	0.258	–	0.9264
0.1	0.5	0.5	0.82	0.82	0.5	0.2916	–	0.91
0.12	0.5	0.5	0.7888	0.7888	0.5	0.3223	–	0.8944
0.14	0.5	0.5	0.7592	0.7592	0.5	0.351	–	0.8796
0.16	0.5	0.5	0.7312	0.7312	0.5	0.3791	–	0.8656
0.18	0.5	0.5	0.7048	0.7048	0.5	0.4062	–	0.8524
0.2	0.5	0.5	0.68	0.68	0.5	0.4321	–	0.84
0.22	0.5	0.5	0.6568	0.6568	0.5	0.456	–	0.8284
0.24	0.5	0.5026	0.6352	0.6352	0.5	0.4779	–	0.8176
0.26	0.5	0.5252	0.6152	0.6152	0.5	0.498	–	0.8076
0.28	0.5	0.5467	0.5968	0.5968	0.5	0.5165	–	0.7984
0.3	0.527	0.567	0.58	0.58	0.5	0.5335	–	0.79
0.32	0.562	0.5864	0.5648	0.5648	0.5	0.5493	–	0.7824
0.34	0.593	0.6045	0.5512	0.5512	0.5	0.5644	–	0.7756
0.36	0.616	0.6218	0.5392	0.5644	0.5	0.5786	–	0.7696
0.38	0.642	0.6451	0.5288	0.5787	0.5	0.5914	–	0.7644
0.4	0.671	0.6727	0.5258	0.5973	0.5	0.603	–	0.7705
0.42	0.698	0.6994	0.5587	0.6238	0.5	0.6134	–	0.7776
0.44	0.721	0.7216	0.6013	0.6483	0.5	0.6227	–	0.785
0.46	0.734	0.7351	0.6353	0.668	0.5	0.6305	–	0.7919
0.48	0.725	0.7257	0.6451	0.6737	0.5	0.6371	–	0.798
0.5	0.7027	0.7016	0.6414	0.6415	0.644	0.6507	0.811	0.8164

	MAX-k-DS		MAX-k-VC			MAX-k-DS		MAX-k-VC	
σ	Prev.	Our Meth.	Prev.	Our Meth.	σ	Prev.	Our Meth.	Prev.	Our Meth.
0.02	0.02	0.0193	0.75	0.75	0.52	0.6022	0.6339	0.822	0.843
0.04	0.04	0.0407	0.75	0.75	0.54	0.6161	0.6471	0.8307	0.8532
0.06	0.06	0.0604	0.75	0.75	0.56	0.6287	0.6585	0.8377	0.8625
0.08	0.08	0.084	0.75	0.75	0.58	0.6402	0.6667	0.8425	0.8707
0.1	0.1	0.1123	0.75	0.75	0.6	0.6488	0.6753	0.8453	0.8784
0.12	0.12	0.1421	0.75	0.75	0.62	0.6539	0.6807	0.8556	0.886
0.14	0.14	0.1726	0.75	0.75	0.64	0.6563	0.685	0.8704	0.8934
0.16	0.16	0.2027	0.75	0.75	0.66	0.66	0.6888	0.8844	0.9008
0.18	0.18	0.2335	0.75	0.75	0.68	0.68	0.6927	0.8976	0.9081
0.2	0.2008	0.2644	0.75	0.75	0.7	0.7	0.6976	0.91	0.916
0.22	0.232	0.295	0.75	0.75	0.72	0.72	0.7024	0.9216	0.9241
0.24	0.2631	0.3248	0.75	0.75	0.74	0.74	0.7068	0.9324	0.9328
0.26	0.2942	0.3548	0.75	0.75	0.76	0.76	0.7266	0.9424	0.9424
0.28	0.3245	0.3833	0.75	0.75	0.78	0.78	0.7491	0.9516	0.9516
0.3	0.3541	0.4102	0.75	0.75	0.8	0.8	0.7714	0.96	0.96
0.32	0.3827	0.4359	0.75	0.75	0.82	0.82	0.7934	0.9676	0.9676
0.34	0.4105	0.4619	0.75	0.75	0.84	0.84	0.8152	0.9744	0.9744
0.36	0.4372	0.4864	0.75	0.75	0.86	0.86	0.8367	0.9804	0.9804
0.38	0.4626	0.5092	0.75	0.7538	0.88	0.88	0.858	0.9856	0.9856
0.4	0.4867	0.5305	0.75	0.7684	0.9	0.9	0.8806	0.99	0.99
0.42	0.5095	0.5505	0.7518	0.7819	0.92	0.92	0.9048	0.9936	0.9936
0.44	0.531	0.5688	0.7687	0.7947	0.94	0.94	0.9288	0.9964	0.9964
0.46	0.5511	0.5861	0.7844	0.8082	0.96	0.96	0.9527	0.9984	0.9984
0.48	0.5697	0.6031	0.7987	0.8209	0.98	0.98	0.9764	0.9996	0.9996
0.5	0.6221	0.6223	0.8452	0.8454					

Comparison with Previous Results

For all six problems we compute approximation factors derived from our algorithm and compare it with the best approximation factors previously known. We consider $\sigma = 0.02, 0.04, \cdots, 0.98$ for MAX k-DENSE-SUBGRAPH and MAX k-VERTEX-COVER and $\sigma = 0.02, 0.04, \cdots 0.5$ otherwise, because in these cases the approximation factors for σ are the same as for $1 - \sigma$.
We implemented the computation of the approximation factors in C++, using the program package CPLEX to solve the linear programs.

MAX-k-CUT The previously best factors are due to Ageev and Sviridenko [2] for $\sigma = 0.02, \cdots 0.28$ and due to Han, Ye, Zhang [12] for $\sigma = 0.3, \cdots 0.48$. We have an improvement for $\sigma = 0.24, \cdots 0.48$. For the case $\sigma = 0.5$ we get the same approximation factor 0.7016 as Halperin and Zwick. Feige and Langberg [6] improved this factor to 0.7027, using the RPR^2 rounding technique, which additionally analyzes the correction step of changing the sides of so-called misplaced vertices.

MAX-k-UNCUT For $\sigma = 0.02, \cdots 0.38$ the previously best factors were received by Feige and Langberg [5] and for $\sigma = 0.4, \cdots 0.48$ by Han, Ye, Zhang [12]. We improve these factors for $\sigma = 0.36, \cdots 0.48$. For $\sigma = 0.5$ the approximation factor of 0.6414 [2] can be improved by our algorithm to 0.6415.

MAX-k-DIRECTED-CUT Ageev and Sviridenko [1] showed an approximation factor of 0.5 for arbitrary σ. For $\sigma = 0.28, \cdots 0.48$, we substantially improve this factor. For the case $\sigma = 0.5$ we also improve the approximation factor of 0.644 of Halperin and Zwick to 0.6507.

MAX-k-DIRECTED-UNCUT For $\sigma = 0.02, \cdots 0.48$, the approximation factors have not been considered until now. For $\sigma = 0.5$ the approximation factor of 0.811 [3] can be improved by our algorithm to 0.8164.

MAX-k-DENSE-SUBGRAPH For $\sigma = 0.2, \cdots 0.48$ and $\sigma = 0.52, \cdots 0.64$, the previously best approximation factors were given by Han, Ye and Zhang, for $\sigma = 0.5$ by Halperin and Zwick and in the other cases by Feige and Langberg [5]. Our improvement is for $\sigma = 0.04, \cdots 0.68$.

MAX-k-VERTEX-COVER For $\sigma = 0.02, \cdots 0.4$, Ageev and Sviridenko [2] found the previously best approximation factors. For $\sigma = 0.42, \cdots 0.48$ and $\sigma = 0.52, \cdots 0.6$ they were found by Han, Ye and Zhang and for $\sigma = 0.5$ by Halperin and Zwick. For $\sigma = 0.62, \cdots 0.98$ Feige, Langberg [5] found the previously best factors. Our improvement is for $\sigma = 0.38, \cdots 0.74$.

---

[2] The approximation factor of 0.6436 of Halperin and Zwick seems to be incorrect. On page 16 [11], Halperin and Zwick claim that $\min_{x \in [-\frac{1}{3}, 0]} \{4 \arccos(d_2 x) - 3 \arccos\left(d_2 \frac{4x-1}{3}\right) - \arccos(d_2)\} \geq 0$ holds for all $d_2 \geq 0$. But for $d_2 = 0.81$ (their parameter for MAX-k-UNCUT) and $x = -\frac{1}{3}$ we have: $4 \arccos(-\frac{1}{3} \cdot 0.81) - 3 \arccos(-\frac{7}{9} \cdot 0.81) - \arccos(0.81) < 0$. Using $d_2 = 0.81$, we get an approximation factor of 0.6414.

[3] Again the approximation factor of 0.8118 of Halperin and Zwick seems to be incorrect, as for $d_2 = 0.74$ and $x = -\frac{1}{3}$ we have: $4 \arccos(-\frac{1}{3} \cdot 0.74) - 3 \arccos(-\frac{7}{9} \cdot 0.74) - \arccos(0.74) < 0$. Their approximation factor becomes 0.811.

Acknowledgement

We would like to thank Jiawei Zhang for some helpful discussions.

References

1. A. Ageev, R. Hassin, M. Sviridenko, *A 0.5–Approximation Algorithm for MAX DICUT with Given Sizes of Parts*, SIAM Journal on Discrete Mathematics 14 (2), p. 246-255, 2001.
2. A. Ageev, M. Sviridenko, *Approximation Algorithms for Maximum Coverage and Max Cut with Given Size of Parts*, IPCO '99, p. 17-30, 1999.
3. Y. Asahiro, K. Iwama, H. Tamaki, T. Tokuyama, *Greedily Finding a Dense Subgraph*, Journal of Algorithms 34, p. 203-221, 2000.
4. D. Bertsimas, Y. Ye, *Semidefinite Relaxations, Multivariate Normal Distributions, and Order Statistics*, Handbook of Combinatorial Optimization (Vol. 3), Kluwer Academic Publishers, p. 1-19, 1998.
5. U. Feige, M. Langberg, *Approximation Algorithms for Maximization Problems arising in Graph Partitioning*, Journal of Algorithms 41, p. 174-211, 2001.
6. U. Feige, M. Langberg, *The RPR^2 rounding technique for semidefinite programs*, Proceedings of the 33th Annual ACM Symposium on Theory of Computing, Crete, Greece, p. 213–224, 2001.
7. U. Feige, G. Kortsarz, D. Peleg, *The Dense k-Subgraph Problem*, Algorithmica 29, p. 410–421, 2001.
8. U. Feige, M. Seltser, *On the densest k-subgraph problem*, Technical report, Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot, September 1997.
9. A. Frieze, M. Jerrum, *Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION*, Algorithmica 18, p. 67-81, 1997.
10. M.X. Goemans, D.P. Williamson, *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming*, Journal of the ACM, 42, p. 1115–1145, 1995.
11. E. Halperin, U. Zwick, *A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems*, Random Structures and Algorithms 20 (3), p. 382-402, 2002.
12. Q. Han, Y. Ye, J. Zhang, *An Improved Rounding Method and Semidefinite Programming Relaxation for Graph Partition*, Mathematical Programming 92 (3), p. 509–535, 2002.
13. S. Mahajan, H. Ramesh, *Derandomizing approximation algorithms based on semidefinite programming*, SIAM Journal on Computing 28, p. 1641–1663, 1999.
14. A. Srivastav, K. Wolf, *Finding Dense Subgraphs with Semidefinite Programming*, Approximation Algorithms for Combinatorial Optimization '98, p. 181–191, 1998. (Erratum, Mathematisches Seminar, Universität zu Kiel, 1999.)
15. Y. Ye, *A .699–approximation algorithm for MAX-Bisection*, Mathematical Programming 90 (1), p. 101–111, 2001.
16. Y. Ye, J. Zhang, *Approximation of Dense-$\frac{n}{2}$-Subgraph and the complement of Min-Bisection*, Unpublished Manuscript, 1999.