# Computational experience with Stable Set Relaxations

Gerald Gruber * Franz Rendl *

December 14, 2000

Universität Klagenfurt, Institut für Mathematik
Universitätsstraße 65-67, A-9020 Klagenfurt, Austria
{gerald.gruber, franz.rendl}@uni-klu.ac.at

**Abstract.** We investigate relaxations for the maximum stable set problem based on the Lovász number $\vartheta(G)$ as an initial upper bound. We strengthen this relaxation by adding two classes of cutting planes, odd circuit and triangle inequalities. We present computational results using this tighter model on many classes of graphs.

## 1 The Stable Set Problem

Let $G = (V(G), E(G))$ denote an undirected graph with $|V(G)| = n$ and $|E(G)| = m$. An edge joining nodes $i$ and $j$ is represented by $(ij)$.

A stable set $S$ (also called independent set) in $G$ is by definition a subset of $V(G)$ such that no two vertices in $S$ are joined by an edge in $E(G)$. The *maximum stable set problem* is the problem of finding a stable set of maximum cardinality. We will denote the size of a largest stable set in $G$ by $\alpha(G)$.

The following notation will be used throughout. We will denote by $S(n)$ the space of $n \times n$ symmetric matrices and write $\text{diag}(X)$ to denote the vector formed from the diagonal elements of a square matrix $X$. $\text{trace}(X) = \sum_i X_{ii}$. The symbol $\succeq$ denotes the Löwner partial order, i.e., $A \succeq B$ if $A - B$ is positive semidefinite. $A \succ 0$ means $A$ is positive definite. Let $e$ denote the vector of all ones, $J$ the matrix of all ones and let $E_{ij} = e_i e_j^t + e_j e_i^t$ where $e_i$ stands for the ith column of the identity matrix of appropriate size.

## 2 Tractable Relaxations

The stable set problem can be formulated in several ways as an integer optimization problem. Let us introduce the binary variables $x_i$ with $x_i = 1$ if $i$ is to be contained in some stable set. Then we get the following integer linear program for the stable set problem.

$$(STAB) \qquad \alpha(G) = \max \sum_{i=1}^{n} x_i \text{ such that } x_i + x_j \leq 1 \text{ if } (ij) \in E(G), x_i \in \{0,1\} \text{ for all } i \in V(G).$$

The linear formulation $(STAB)$ is the starting point for the polyhedral approach to the stable set problem, going back to the 1970's. We will review some of the main results of this approach in subsection 2.1.

*Partial financial support from the Austrian Science Foundatation, FWF Project P12660-MAT is gratefully acknowledged.

Modeling the condition, that for $(ij) \in E(G)$ at most one of the nodes $i$ and $j$ can be taken into a stable set, can not only be expressed as $x_i + x_j \leq 1,\ x_i \in \{0,1\}$ but also as $x_i x_j = 0,\ x_i \in \{0,1\}$. This leads to the following integer quadratic optimization formulation of the stable set problem.

$$(STAB) \qquad \alpha(G) = \max x^T x \text{ such that } x_i x_j = 0 \text{ if } (ij) \in E(G),\ x_i^2 = x_i \text{ for all } i \in V(G). \tag{1}$$

This formulation will give rise to the well-known semidefinite relaxation introduced by Lovász in 1979, see [15]. We summarize the main results of this approach in subsection 2.2.

### 2.1 Linear Relaxations

Linear programming based relaxations of a combinatorial optimization problem are based on the study of the convex hull of its integer solutions. Let us denote by $STAB(G)$ the convex hull of incidence vectors of stable sets in $G$, i.e.

$$STAB(G) := \operatorname{conv}\Big\{x \in \{0,1\}^n : x_i + x_j \leq 1 \text{ for all } (ij) \in E(G)\Big\} \tag{2}$$

The following inequalities are all valid for STAB(G).

| | | | |
|---|---|---|---|
| | $x_i \leq 1$ | for isolated nodes | |
| (nonnegativity constraints): | $x_i \geq 0$ | $i \in V(G)$ | (3) |
| (edge constraints): | $x_i + x_j \leq 1$ | $(ij) \in E(G)$ | (4) |
| (odd-cycle constraints): | $\sum_{i \in V(C)} x_i \leq \frac{1}{2}(\lvert V(C)\rvert - 1)$ | $C$ odd cycle in G | (5) |
| (clique constraints): | $\sum_{i \in Q} x_i \leq 1$ | $Q$ clique in G | (6) |

There are many more classes of inequalities, which are valid for $STAB(G)$, such as the odd-anti-hole constraints, the wheel constraints and so on. We refer to Chapter 9 of [8] for a thorough treatment of linear relaxations of the stable set problem.

Several subsets of the constraint classes above were given names. The fractional stable set polytope is given by

$$FRAC(G) = \Big\{x : x \text{ satisfies (3) and (4)}\Big\}.$$

The odd-cycle polytope is given by

$$CSTAB(G) = \Big\{x : x \text{ satisfies } (3), (4) \text{ and } (5)\Big\}.$$

The Clique-constraints are collected in the clique-polytope

$$QSTAB(G) = \Big\{x : x \text{ satisfies (3) and (6)}\Big\}.$$

Here we recall a few of the most fundamentals facts, which are relevant for our approach. The simplest constraints (3) and (4) suffice only for the trivial case of bipartite graphs.

**Fact 1**. [8] *$STAB(G) = FRAC(G)$ if and only if $G$ is bipartite and has no isolated nodes.*

While the separation problem for $CSTAB(G)$ can be done in polynomial time via shortest path computations in graphs with nonnegative edge weights, it is likely to be difficult to optimize over $QSTAB(G)$.

**Fact 2**.[7] *Any linear function can be optimized over $CSTAB(G)$ in polynomial time.*

**Fact 3**.[6] *It is NP-complete to optimize a linear objective function over $QSTAB(G)$.*

$STAB(G)$ is in general a proper subset of $QSTAB(G)$. The question of characterizing graphs $G$ where QSTAB(G) = STAB(G) leads to one of the most intriguing topics of graph theory related to the stable set problem, the study of perfect graphs. For a formal definition of perfect graphs, we refer to [6]. In the 1970's Fulkerson [5] and Chvátal [4] characterized the graphs for which $QSTAB(G) = STAB(G)$. They showed,

**Fact 4**. *$STAB(G) = QSTAB(G)$ if and only if $G$ is perfect.*

## 2.2 Theta function

In this section we recall the stable set relaxation introduced by Lovász, see [15]. There are many different ways to derive this relaxation. Most of them were already analyzed in the original paper [15].

A quick way to obtain this relaxation was proposed by Lovász and Schrijver [17], and goes as follows. Suppose $x \in \{0,1\}^n$ is the characteristic vector of a stable set in a graph $G$. Consider the rank-one matrix $Y = \binom{x}{1}\binom{x}{1}^T$. If we partition this matrix as $Y = \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix}$, then the main diagonal of $X$ is contained in the following set $TH(G)$.

$$TH(G) = \left\{ x \in \mathbb{R}^n : \exists Y = \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix},\ Y \succeq 0,\ \operatorname{diag}(X) = x,\ x_{ij} = 0\ (ij) \in E(G) \right\} \tag{7}$$

Therefore optimizing over $TH(G)$ gives a relaxation of $\alpha(G)$, which we denote by $\vartheta(G)$.

$$\vartheta(G) := \max \left\{ \sum_i x_i : x \in TH(G) \right\} \geq \alpha(G). \tag{8}$$

**Remark 5** *It is well known that $\begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0$ if and only if $X - xx^T \succeq 0$.*

$\vartheta(G)$ is obtained as the solution of a semidefinite programming problem (SDP). It can be computed in polynomial time to some fixed prescribed precision, see [7, 21]. It is pointed out in [22] that the relaxation leading to $\vartheta(G)$ can alternatively be derived by taking the Lagrangian dual of the stable set formulation as a quadratically constrained quadratic problem in binary variables (1). This is also observed more recently in [14]. The derivation through the Lagrangian dual is quite general and this approach applies to many other discrete optimization problems, see e.g. [22]. The set $TH(G)$ can be viewed as the projection of the set of $(n+1) \times (n+1)$ matrices $Y$ satisfying

$$y_{ij} = 0 \text{ for all } (ij) \in E(G), \quad y_{ii} = y_{i,n+1} \text{ for all } i \in V(G), \quad y_{n+1,n+1} = 1, \quad Y \succeq 0 \tag{9}$$

onto its main diagonal entries 1 to $n$.

Even though the derivation of the relaxation (8) is quite natural, there are several seemingly different relaxations of $\alpha(G)$, which all give the same value $\vartheta(G)$. We review some basic results which are useful for our purposes.

Let $x \neq 0$ again be the characteristic vector of some stable set in $G$. Then $e^T x = x^T x$ and the matrix $X := \frac{1}{x^T x} x x^T$ has the following properties:

$$X \succeq 0,\ x_{ij} = 0 \text{ if } (ij) \in E(G),\ \operatorname{trace}(X) = 1.$$

Moreover $\operatorname{trace}(JX) = e^T x$. Thus

$$z(G) := \max \left\{ \operatorname{trace}(JX) : \operatorname{trace}(X) = 1,\ x_{ij} = 0 \text{ if } (ij) \in E(G),\ X \succeq 0 \right\} \geq \alpha(G). \tag{10}$$

The relaxation $z(G)$ is among the formulations of $\vartheta(G)$, investigated in [15]. It is shown e.g. in [17] that indeed $z(G) = \vartheta(G)$. Since this result is important for our computational approach, we include the following proof, which differs from the argument in [17]. We start out with some simple observations.

**Lemma 6** *Let $X \succeq 0$ and set $z = \mathrm{trace}(JX)$. Then there exists a matrix $B = (b_1, \ldots)$ such that $X = BB^T$ and $Xe = \sqrt{z} b_1$.*

**Proof.** If $X \succeq 0$, then there exists $C$ such that $X = CC^T$. We set $x = \frac{1}{\sqrt{z}} C^T e$ and note that $x^T x = \frac{1}{z} e^T CC^T e = \frac{1}{z}\mathrm{trace}(JX) = 1$. Thus there exists an orthogonal matrix $Q = (x, \ldots)$ with $x$ as first column. The matrix $CQ = B = (b_1, \ldots)$ satisfies $BB^T = CQQ^T C^T = X$ and $Xe = BQ^T C^T e = \sqrt{z} BQ^T x = \sqrt{z} B e_1 = \sqrt{z} b_1$.

□

**Lemma 7** *Let $X$ be optimal for (10) with value $z(G) = \mathrm{trace}(JX)$. Then $Xe = z(G)\mathrm{diag}(X) \in TH(G)$.*

**Proof.** The dual of (10) is the following SDP, see e.g. [12].

$$\min \lambda \text{ such that } S := \lambda I - J + \sum_{(ij)\in E(G)} y_{ij} E_{ij} \succeq 0.$$

Let $X$ be optimal for (10), and $\lambda, \{y_{ij} : (ij) \in E(G)\}$ be optimal for the dual problem. Strong duality holds for these problems, because both problems satisfy the Slater constraint qualification. Therefore $z(G) = \lambda$ and $SX = 0$. Thus

$$0 = (SX)_{ii} = s_{ii} x_{ii} + \sum_{(ij)\in E(G)} s_{ij} x_{ij} + \sum_{(ij)\notin E(G)} s_{ij} x_{ij} = (z(G) - 1) x_{ii} - \sum_{(ij)\notin E(G)} x_{ij},$$

because $s_{ii} = z(G) - 1$, $x_{ij} = 0$ for $(ij) \in E(G)$ and $s_{ij} = -1$ for $(ij) \notin E(G)$. This shows that $z(G) x_{ii} = (Xe)_i$.

From Lemma 6 we get $X = BB^T$ and $d := z(G)\mathrm{diag}(X) = Xe = \sqrt{z(G)} b_1$. Therefore $z(G)X - dd^T = z(G)(BB^T - b_1 b_1^T) = z(G)\sum_{i>1} b_i b_i^T \succeq 0$. Moreover, $x_{ij} = 0$ for $(ij) \in E(G)$, showing that $d \in TH(G)$.

□

**Lemma 8** *Let $x \in TH(G)$, i.e. $\exists X = (x_{ij})$, $x = \mathrm{diag}(X)$, $x_{ij} = 0$ if $(ij) \in E(G)$, $X - xx^T \succeq 0$. Suppose $\sum_i x_i = \gamma > 0$. Then $Y = \frac{1}{\gamma} X$ is feasible for (10) and $\mathrm{trace}(JY) \geq \gamma$.*

**Proof.** Feasibility of $Y$ for (10) is obvious. Since $X - xx^T \succeq 0$ we get $e^T X e \geq (e^T x)^2 = \gamma^2$. Therefore $e^T Y e \geq \gamma$.

□

Using these results we show now how optimal solutions of (10) and (8) relate to each other.

**Theorem 9** *Let $x \in TH(G)$ be optimal for (8), i.e. $\exists X = (x_{ij})$ with $x = \mathrm{diag}(X)$, $x_{ij} = 0$ if $(ij) \in E(G)$, $X - xx^T \succeq 0$ and objective value $\vartheta(G) = \sum_i x_i$. Further, let $Y$ be optimal for (10) with $z(G) = \mathrm{trace}(JY)$. Then*

$$z(G)\mathrm{diag}(Y) \textit{ is optimal for } (8),$$

$$\frac{1}{\vartheta(G)} X \textit{ is optimal for } (10) \textit{ and } z(G) = \vartheta(G).$$

**Proof.** Let $x, X, Y$ satisfy the hypotheses of the theorem. Lemma 7 shows that $\bar{z} := z(G)\mathrm{diag}(Y)$ is feasible for (8) and $z(G) = \sum_i \bar{z}_i \leq \sum_i x_i = \vartheta(G)$. Lemma 8 shows that $\vartheta(G) \leq e^T(\frac{1}{\vartheta(G)} X)e \leq z(G)$.

□

There is some asymmetry between the two transformations. While it is straightforward to transform feasible solutions from (8) to feasible solutions of (10) with the same objective value or higher, see Lemma 8, it is not clear how arbitrary feasible solutions of (10) can be transformed into feasible solutions of (8). Lemma 7 makes essential use of $X$ being optimal.

We close this section with some basic facts about the set $TH(G)$.

**Fact 10** [6, 7]

1. *$STAB(G) \subseteq TH(G) \subseteq QSTAB(G)$.*
2. *$TH(G)$ is polyhedral if and only if $G$ is perfect.*
3. *$STAB(G) = TH(G) = QSTAB(G)$ if and only if $G$ is perfect.*
4. *We can optimize linear functions over $TH(G)$ in polynomial time.*

**Remark 11**

1. *It is easy to see that adding the rank-one constraint to matrices satisfying (9) produces precisely the characteristic vectors of stable sets.*

$$\Big\{Y : Y \text{ satisfies } (9),\ \operatorname{rank}(Y) = 1\Big\} = \left\{ \begin{pmatrix} x \\ 1 \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix}^T : x \text{ characteristic vector of a stable set in } G \right\}.$$

2. *Adding the rank-one constraint to the relaxation (10) gives again $\alpha(G)$:*

$$\alpha(G) = \max\Big\{\operatorname{trace}(JX) : \operatorname{trace}(X) = 1,\ x_{ij} = 0\ (ij) \in E(G),\ X \succeq 0,\ \operatorname{rank}(X) = 1\Big\}.$$

**Proof.** *Let $X$ be an optimal solution to (10) of rank-one, $X = uu^T$. Since $u_i u_j = 0$ for $(ij) \in E(G)$, we see that $u$ is nonzero only on some stable set $A$. We have to show that $u_i = u_j$ for $i, j \in A$. Suppose to the contrary that $u_i > u_j > 0$ for $i, j \in A$. We set $v_k = u_k$ for $k \neq i, j$ and let $v_i = v_j = \sqrt{\frac{u_i^2 + u_j^2}{2}}$. Then $vv^T$ is feasible for (10) but $v_i + v_j > u_i + u_j$, contradicting the optimality of $uu^T$.* □

*This argument also shows that not all feasible rank-one matrices of (10) lead to multiples of characteristic vectors of stable sets.*

3. *In view of Theorem 9, we have two SDP models available to compute $\vartheta(G)$. Theoretically, they are equivalent, but the practical computational effort is much smaller for (10). In the following table we include computation times for both models, and also compare with the software package CSDP3.0 written in C, which also contains a module to compute (10). The test problems are taken from this software package. Column $n$ gives the number of vertices, column $m$ the number of edges in the graph $G$.*

| | | | | CPU time [sec] | | |
|---|---|---|---|---|---|---|
| File | $n$ | $m$ | $\vartheta(G)$ | (10) | (8) | CSDP 2.3 |
| g100 | 100 | 497 | 32.8792 | 7 | 68 | 16 |
| g150 | 150 | 1105 | 42.1660 | 43 | 645 | 166 |
| g200 | 200 | 1948 | 50.3212 | 176 | 4172 | 921 |
| g250 | 250 | 3027 | 57.2323 | 589 | no attempt | 4230 |

Table 1: *Comparison of running times for computing $\vartheta(G)$*

### 2.3 Matrix relaxations: Lift and Project

The derivation of $\vartheta(G)$ can be viewed as going from $\mathbb{R}^n$ into the space of symmetric $n \times n$ matrices, and then back to $\mathbb{R}^n$ again, by projecting onto the main diagonal. This idea is further elaborated in [17], and was independently used by many other researchers, see e.g. [1, 25]. We now recall the approach investigated in [17]. Given a polytope $P = \{x : Ax - b \geq 0\}$, let $P_I$ denote the convex hull of its 0-1 integral points. Multiplying each inequality defining $P$ by $x_k$ and $1 - x_k$ results in a system of quadratic inequalities. This system is linearized by substituting $y_{ij}$ for $x_i x_j$. 0-1 solutions also satisfy $x_i^2 = x_i$, hence $x_i^2$ is replaced by $y_{ii}$. Imposing further that $y_{ij} = y_{ji}$ leads to inequalities which are now linear in $y_{ij}$. We denote the set of all symmetric matrices $Y$ satisfying these linearized constraints by $M(P)$. Projection onto the main diagonal gives the set $N(P)$,

$$N(P) = \Big\{x : \exists Y \in M(P) \text{ such that } x = \operatorname{diag}(Y)\Big\}.$$

It is clear from the definition that

$$P_I \subseteq N(P) \subseteq P.$$

We now recall the results of [17] in the case $P = FRAC(G)$. Multiplying $x_i \geq 0$ by $x_k \geq 0$ and $1 - x_k \geq 0$, and linearizing gives

$$0 \leq y_{ik} \leq y_{ii}.$$

Similarly we multiply $1 - x_i - x_j \geq 0$ with $x_k$ and $1 - x_k$ and get for all $(ij) \in E(G)$ and for all $k$:

$$y_{kk} - y_{ik} - y_{jk} \geq 0 \text{ and } 1 - y_{ii} - y_{jj} - y_{kk} + y_{ik} + y_{jk} \geq 0.$$

It should be noted that $y_{ik} \leq y_{kk}$ is implied by the second set of inequalities. Moreover, it is easy to see that $y_{ij} = 0$ for $(ij) \in E(G)$. We conclude

$$M(FRAC(G)) = \Big\{Y = (y_{ij}) : Y \geq 0, y_{ii} + y_{jj} + y_{kk} - 1 \leq y_{ik} + y_{jk} \leq y_{kk} \quad \forall (ij) \in E(G),\ \forall k\Big\}. \tag{11}$$

Since $M(FRAC(G))$ is given by an explicit system of inequalities (of polynomial size), we can optimize in polynomial time over $N(FRAC(G))$.

We close this section with yet another view to lift and project. It is well known that the quadric boolean polytope $\operatorname{conv}\{xx^T : x \in \{0,1\}^n\}$ is contained in the metric polytope $MET(n)$. By definition, a matrix $X$ satisfies

$$X \in MET(n) :\Leftrightarrow \begin{cases} 0 \leq x_{ij} \leq x_{ii}, \\ x_{ii} + x_{jj} - x_{ij} \leq 1, \\ -x_{kk} - x_{ij} + x_{ik} + x_{jk} \leq 0, \\ x_{kk} + x_{ii} + x_{jj} - x_{ij} - x_{ik} - x_{jk} \leq 1, \text{ for } 1 \leq i,j,k \leq n. \end{cases} \tag{12}$$

Let us define $MET(G)$ to consist of those matrices in $MET(n)$, where $x_{ij} = 0$ for $(ij) \in E(G)$,

$$MET(G) = \Big\{X \in MET(n) : x_{ij} = 0 \text{ if } (ij) \in E(G)\Big\}.$$

We observe that $MET(G) \subseteq M(FRAC(G))$, because $M(FRAC(G))$ contains all the inequalities of $MET(G)$ for which $(ij) \in E(G)$. On the other hand, $MET(G)$ contains the additional inequalities $-x_{kk} - x_{ij} + x_{ik} + x_{jk} \leq 0$, and $x_{kk} + x_{ii} + x_{jj} - x_{ij} - x_{ik} - x_{jk} \leq 1$ for stable sets $i, j, k$, which clearly are not contained in M(FRAC(G)). It turns out however, that these constraints do not further tighten the relaxation.

**Theorem 12**

$$N(FRAC(G)) = CSTAB(G) = N(MET(G)).$$

This result follows directly from Theorem 2.3 in [17], which gives the first equality, and Proposition 6 in [18], which yields the second equality (in our notation). The argument below combines ideas from both papers. Before proceeding with the proof, we need to introduce some more notation, and recall some facts about the triangle and cycle polytopes.

Suppose $G$ is a graph. The cycle polytope of $G$, $CYCLE(G)$, is defined as follows.

$$CYCLE(G) := \left\{ x \in \mathbb{R}^{E(G)} : 0 \leq x \leq 1,\ x(F) - x(C \setminus F) \leq |F| - 1\ \forall \text{ cycles } C, F \subseteq C, |F| \text{ odd} \right\}.$$

A closely related structure is the triangle polytope $TRI(n)$.

$$TRI(n) := \Big\{ X = X^T :\ diag(X) = e,\ x_{ij} + x_{ik} + x_{jk} \geq -1,\ x_{ij} - x_{ik} - x_{jk} \geq -1, \text{ for all distinct triples } i,j,k \leq n \Big\}.$$

The following result of Barahona shows that for graphs $G$ on $n$ vertices there is a close relation between these two polytopes.

**Fact 13** [2] *Let $G$ be a graph on $n$ vertices. Then $\xi \in CYCLE(G) \Longrightarrow \exists X \in TRI(n)$ such that $x_{ij} = 1 - 2\xi_{ij}\ \forall (ij) \in E(G)$.*

A second well known fact relates $TRI(n+1)$ to $MET(n)$.

**Fact 14** [26] *Let $Y$ be a matrix of order $n+1$, indexed by $\{0,\ldots,n\}$. Let $V = (v_{ij})$ be defined by $v_{ii} = \frac{1}{2}(1 - y_{0i})$, $v_{ij} = \frac{1}{4}(1 + y_{ij} - y_{0i} - y_{0j})$. Then $Y \in TRI(n+1) \Rightarrow V \in MET(n)$.*

We mention that both results have a converse, not explicitely stated here.

As a final result, we recall the following fact from [18].

**Fact 15** *Let $G$ be a graph on $n$ vertices and let $G^{\triangledown} := (V(G) \cup 0, E(G) \cup (0i),\ i \in V(G))$ denote the graph obtained from $G$ by adding a new vertex 0 and joining it to all vertices of $G$. Let $d \in \mathbb{R}^n$ and define*

$$\xi_{0i} := d_i,\ \xi_{ij} := d_i + d_j\ \forall (ij) \in E(G). \tag{13}$$

*Then*

$$d \in CSTAB(G) \Rightarrow \xi \in CYCLE(G^{\triangledown}).$$

We are now ready to prove the Theorem.

**Proof.** Since $N(MET(G)) \subseteq N(FRAC(G))$, it suffices to show that $N(FRAC(G)) \subseteq CSTAB(G)$ and $CSTAB(G) \subseteq N(MET(G))$.

To show that $N(FRAC(G)) \subseteq CSTAB(G)$ we use an argument from [16] and assume $x \in N(FRAC(G))$. Hence there exists a matrix $Y \in M(FRAC(G))$ such that $x = \mathrm{diag}(Y)$. Clearly $x$ satisfies (3) and (4). Let $C$ be an odd cycle in $G$, wlog. $C = \{1,\ldots,2k+1\}$. Starting with edge $(2,3)$ we alternatively use the first and second inequality from $M(FRAC(G))$ and stop with edge $(2k, 2k+1)$. We get, using $y_{12} = y_{1,2k+1} = 0$

$$\begin{aligned}
y_{22} + y_{33} - y_{13} &\leq 1 - y_{11} \\
y_{13} + y_{14} &\leq y_{11} \\
y_{44} + y_{55} - y_{14} - y_{15} &\leq 1 - y_{11} \\
y_{15} + y_{16} &\leq y_{11} \\
&\vdots \\
y_{1,2k-1} + y_{1,2k} &\leq y_{11} \\
y_{2k,2k} + y_{2k+1,2k+1} - y_{1,2k} &\leq 1 - y_{11}
\end{aligned}$$

Adding up all these constraints yields $\sum_{i \in C} y_i \leq k$, thus $x = \mathrm{diag}(Y) \in CSTAB(G)$.

To see that $CSTAB(G) \subseteq N(MET(G))$ we follow the argument from [18]. From Fact 15 we conclude that

$$d \in CSTAB(G) \Rightarrow \xi \in CYCLE(G^{\bigtriangledown})$$

with $\xi$ given by (13). Fact 13 in turn implies the existence of a matrix $Y \in TRI(n+1)$, indexed from 0 to $n$, such that

$$y_{ij} = 1 - 2\xi_{ij} \ \forall (ij) \in E(G^{\bigtriangledown}).$$

Hence, $y_{ii} = 1,\ y_{0i} = 1 - 2d_i,$ and $y_{ij} = 1 - 2d_i - 2d_j \ \forall (ij) \in E(G)$. Fact 14 shows that $Y \in TRI(n+1)$ implies $V \in MET(n)$, where $v_{ii} = \frac{1}{2}(1 - y_{0i}) = d_i,\ v_{ij} = \frac{1}{4}(1 + y_{ij} - y_{0i} - y_{0j}) = \frac{1}{4}[1 + (1 - 2d_i - 2d_j) - (1 - 2d_i) - (1 - 2d_j)] = 0 \ \forall (ij) \in E(G)$. Therefore $V \in MET(G)$, and $d = diag(V) \in N(MET(G))$.

$\square$

Lovasz and Schrijver propose a further refinement of $M(P)$ and $N(P)$ by observing that matrices $Y$ in $M(P)$ satisfy the following semidefiniteness condition.

$$Y - \mathrm{diag}(Y)\mathrm{diag}(Y)^T \succeq 0. \tag{14}$$

Hence they define $M_+(P) := \{Y \in M(P) : Y$ satisfies (14)$\}$, and similarly $N_+(P)$ to be the projection of $M_+(P)$ onto the main diagonal.

In case of $P = FRAC(G)$ we get

$$M_+(FRAC(G)) = \Big\{Y = (y_{ij}) : Y \in M(FRAC(G)), Y \text{ satisfies } (14)\Big\} \tag{15}$$

Optimizing over $N_+(FRAC(G))$ can still be done in polynomial time [17], but practical implementations are currently not available. It is clear that $N_+(FRAC(G)) \subseteq TH(G)$.

In this paper we are interested in optimizing over $N_+(FRAC(G))$. This amounts to solving an SDP with $m = |E(G)|$ equality constraints ($x_{ij} = 0 \ (ij) \in E(G)$) and $O(mn)$ inequalities, given by $M(FRAC(G))$. Solving this SDP directly with current algorithms is therefore out of reach for graphs of interesting sizes, say $n \geq 30, m \geq 100$.

In view of Theorem 12 it is clear that when optimizing over $N_+(FRAC(G))$, we can ignore the odd cycle constraints, as they are automatically satisfied. We will show below that these constraints are nontheless quite useful in the case where we approximate the optimization over $N_+(FRAC(G))$ by an iterative procedure, where only some of the linear inequalities defining $N_+FRAC(G))$ are considered.

## 3 Towards approximating $N_+(FRAC(G))$

In Table 1 we gave some idea about the problem sizes for which it is feasible to optimize over $TH(G)$ yielding $\vartheta(G)$. Now we are interested to investigate by how much we can improve $\vartheta(G)$, when approximating $N_+(FRAC(G))$. Even though we could directly include all $O(nm)$ inequalities from (15) and solve the resulting SDP, we do not follow this approach as it is highly inefficient. Instead we use the approach from [12, 11] where an iterative scheme is proposed, in which only a small subset of (12) is selected carefully to be added to

$$F(G) := \{X \in S(n) : \mathrm{diag}(X) \in TH(G)\}.$$

Since we are interested in tightening the relaxation $TH(G)$ by adding further linear inequalities, we need a model, where those constraints can conveniently be included. In case of (8) this is straightforward. It is not clear how the odd circuit constraints for instance could be included in (10). We therefore use the fast model (10) to generate an initial relaxation $X$ which we transform with Theorem 9 into an optimal solution of (8). Starting from this point we include further violated cutting planes, if there are any.

In [11] some preliminary computational experience is reported for optimizing over $F(G) \cap MET(G)$. The main message seems to be that it is computationally very demanding to optimize over this set as the number of inequalities defining $MET(G)$ grows cubically with $n$.

It should be mentioned that Schrijver [24] proposed a strengthening of the Lovász bound $\vartheta(G)$ by adding the nonnegativity constraints. These are only a subclass of the constraints of $MET(G)$, and there does not seem to be any reason to prefer them over the general constraints defining $MET(G)$. Optimizing over $F(G) \cap MET(G)$ clearly dominates the Schrijver model.

In our approach one refinement of $TH(G)$ with problem-specific cutting planes consists in intersecting $F(G)$ with

$$C(G) := \{X \in S(n) : \mathrm{diag}(X) \in CSTAB(G)\}.$$

Since the separation problem for $CSTAB(G)$ reduces to the shortest path problem in graphs with nonnegative edge weights, see Grötschel et al [7], we can easily detect constraints violating $CSTAB(G)$ and incorporate them in our model. Directly solving this relaxation is computationally nontrivial, as there could be an exponential number of constraints. In contrast, $MET(n)$ has only $O(n^3)$ constraints.

Ultimately we are interested in getting improved bounds on $\alpha(G)$ as quickly as possible. Practical experience has shown that at the beginning the inclusion of a moderate number of triangles and odd circuits induces a significant improvement of the bound on $\alpha(G)$ contrary to adding the most violated triangles only. However, this favorable effect typically vanishes with an increasing number of added triangles. Ultimately we expect the odd circuits to become redundant in view of Theorem 12. We reemphasize that including cycle inequalities together with triangle inequalities dominates optimizing over $F(G) \cap MET(G)$ mostly at the beginning. We exploit this observation and consider as tightest relaxation in this paper optimizing over $F(G) \cap C(G) \cap MET(G)$. For more details the reader is refered to [9].

It is the main point of this work to compare the quality of the different relaxations and investigate their computational effort on a series of test problems. The computational effort to solve these relaxations is heavily influenced by the number of constraints included into the semidefinite program and is ultimately growing at least cubically with the number of constraints (because a linear system has to be solved for the dual variables). To get a feeling for the computational effort to solve these relaxations (at least approximately), we include in Table 2 some computation times on several randomly generated problems of various size. This table compares computation times from optimizing over the different relaxations. Column $n$ gives the respective order of the graphs. We also provide the average number $m$ of edges and the density. Each line lists the average CPU time to solve 10 problems of given size and density. The values in the parenthesis indicate the (average number of) active constraints at termination. It should be noted that optimizing over $MET(G)$ is not done exactly, but the iterations are stopped once the maximum violation of $MET(G)$ has fallen below a threshold (of 50 % of the maximum initial violation in our case). On the positive side, optimizing over $F(G) \cap C(G)$ is not much harder than computing $\vartheta(G)$ and as can be seen later leads to significantly improved bounds on $\alpha(G)$. This justifies our approach.

## 4 Interior-Point implementation

In this section we discuss the essential ingredients of our cutting plane technique. The implementation of our algorithm involves solving a number of semidefinite problems. These programs are solved using primal-dual path-following interior-point methods. The search direction used in our implementation was proposed independently by Helmberg et al. [12], Kojima et al. [13] and later based on a quite different approach by Monteiro [20]. Therefore this search direction is called HKM-direction. All our codes are implemented in Matlab with interfaces in C.

| | | CPU time [sec] from optimizing over | | | | |
|---|---|---|---|---|---|---|
| $n$ | $density[\%] - m$ | $F(G)$ | $F(G) \cap C(G)$ | | $F(G) \cap C(G) \cap MET(G)$ | |
| 50 | 10 - 125 | 0.7125 | 4.3357 | (10) | 8.8111 | (8/47) |
| 100 | 10 - 500 | 5.0945 | 41.4115 | (9) | 141.8472 | (2/180) |
| 150 | 5 - 559 | 10.0519 | 150.7397 | (42) | 354.3603 | (22/266) |
| 200 | 3 - 597 | 16.1386 | 309.5576 | (94) | 958.8453 | (63/383) |
| 250 | 2 - 623 | 25.5563 | 424.7727 | (95) | 1458.3181 | (98/297) |

Table 2: *Average Running Times on randomly generated sparse graphs*

### 4.1 Computing $\vartheta(G)$

We first compute the optimal solution of (10). Let $Y^*$ denote this solution. From Theorem 9, it follows that the Lovász number $\vartheta(G)$ is the corresponding optimal objective value. Moreover also from Theorem 9 we know that

$$\bar{X} = \begin{pmatrix} X & d \\ d^T & 1 \end{pmatrix} \quad \text{where} \quad X = \vartheta(G)Y^* \text{ and } d = \operatorname{diag}(X)$$

is optimal for (8). We take the value of (10) as initial upper bound on $\alpha(G)$.

### 4.2 Rounding

The main diagonal of the optimal $X \in S(n)$ is used to generate a stable set as follows. A large entry $x_{ii}$ indicates that vertex $i$ is likely to be contained in a large stable set. Hence we run through all the vertices of $G$ in the order given by the main diagonal of $X$ sorted nonincreasingly. We select the vertices to be contained in a stable set with probability proportional to $x_{ii}$. If vertex $j$ is accepted to be in a stable set we exclude all the neighbours of $j$. The final stable set is locally improved in a second phase of the rounding heuristic. The complete process is repeated several times. We include the size of the largest stable set found this way in the tables of section 5.

### 4.3 Tightening the Relaxation

The initial semidefinite bound $\vartheta(G)$ (in general the upper bound of the current model) can be improved now by identifying and including violated odd circuit constraints and/or triangle inequalities. In general there are a lot more violated cutting planes than we are willing to add to the current relaxation. The computational effort to reoptimize is growing drastically with the number of constraints. For this reason we follow an iterative procedure and add only the most violated constraints. Since interior-point techniques are used for solving the relaxations we have to assure that the initial point is an interior point (not necessarily feasible point). Hence the current point has to be modified for restarting. A detailed description of this modification is given in section 4.4.

Now let us turn to the process of adding cutting planes and reoptimizing. For practical reasons we do not solve to optimality after each phase of adding cutting planes. While iterating we search for cutting planes once primal and dual feasibility is reached and add the new violated ones. After several rounds of collecting inequalities we solve the present model to optimality. This allows us to remove constraints that have become inactive. We then proceed to the next phase of adding cutting planes, until some stopping condition is satisfied. We refer the reader to [9, Chapter 5] for extensive implementation details.

In our experiments we found that optimizing over $F(G) \cap C(G)$ can be done efficiently with computational effort of the same order of magnitude as computing $\vartheta(G)$. On the other hand we found it impractical to optimize over all constraints $MET(G)$, because there was a typical tailing off effect. The biggest improvement in the bound is obtained in the first few iterations of adding violated inequalities. In subsequent iterations, the bound is hardly reduced, the largest violation of $MET(G)$ is already quite small, but there are still a lot of inequalities violating $MET(G)$. In this situation it does not seem to be advisable to continue the iterations, because of the excessive computational effort.

### 4.4 Restarting

Let the linear operator $\mathcal{B}(\cdot)$ be defined such that all added cutting planes with complying right hand side $b$ may be written as $\mathcal{B}(X) \leq b$. The current optimal $X$ is no longer an interior point, because it violates the most recently added cuts. Since the interior-point machinery requires an initial $X \succ 0$ which strictly satisfies these inequalities, we use a simple trick in formulating the current semidefinite model. We introduce a primal slack variable $s = b - \mathcal{B}(X) \geq 0$. We then introduce an artificial dual slack variable $t \geq 0$ for the dual variable $r$ corresponding to $s$. This way we get the additional dual constraint $t = r$. This restatement of the semidefinite model simplifies the update of the variables used in the interior-point code. To get the initial variables $(X^\circ, y^\circ, Z^\circ)$ we implemented a simple backtracking strategy from the current $(X, y, Z)$ in the direction to a well centered point in the interior of the feasible set of (8). This is a standard technique, therefore we do not restate it here precisely. The initial variables $r^\circ, s^\circ$ and $t^\circ$ are constructed as follows. We set $(r_i, s_i, t_i) = (0, 0.1, 0.1)$ for every component $i$ corresponding to a newly added inequality. We take the current variables $r, s$ and $t$ for updating the remaining components. We set

$$r_i^\circ = \begin{cases} r_i & \text{if } r_i \geq 0.1 \\ 0 & \text{otherwise} \end{cases}, \qquad s_i^\circ = \begin{cases} s_i & \text{if } s_i \geq 0.1 \\ 0.1 & \text{otherwise} \end{cases} \qquad \text{and} \qquad t_i^\circ = \begin{cases} t_i & \text{if } t_i \geq 0.1 \\ 0.1 & \text{otherwise.} \end{cases}$$

The point constructed this way is strictly interior but not feasible for the current semidefinite model.

### 4.5 Optimizing Over *CSTAB(G)*

We also report results from optimizing over the linear relaxation $CSTAB(G)$ of the stable set polytope. ILOG CPLEX 6.5 is used to solve the linear programs.

Note that $x = \frac{1}{3}e,\ x \in \mathbb{R}^n$ is contained in $CSTAB(G)$. Therefore the optimal objective value is at least $\frac{n}{3}$. In our extensive experiments it has been observed that optimizing over $CSTAB(G)$ often yields $x = \frac{1}{3}e$ if $\alpha(G) < \frac{n}{3}$. Therefore we present numerical results from optimizing over $CSTAB(G)$ only if $\alpha(G) > \frac{n}{3}$.

## 5 Computational results

In this section we report computational experience on different classes of test problems. It is the primary objective of this section to compare the quality of the different relaxations described above. We have already discussed the computational effort in section 3. Therefore running times are presented scantily. All our computational tests were performed on a DEC Alpha workstation.

### 5.1 Test Problems

Our numerical experiments were carried out on the following sets of graphs.

(i) Random graphs with given density,

(ii) Sanchis graphs, see [23, 10],

(iii) Mannino graphs, see [19],

(iv) Johnson graphs, see [10],

(v) Triangulated planar graphs,

(vi) Graphs with high girth,

(vii) Geometric graphs.

Sanchis graphs are hard instances for the minimum vertex cover problem (and hence for the equivalent maximum clique problem). Originally they were designed to test algorithms for these problems. These graphs are constructed in such a way that $\alpha(G)$ is known, see [23]. The independence number is also known for the Mannino graphs and the Johnson graphs. The Mannino graphs are obtained from the set covering formulation of the Steiner Triple problem [19] and the Johnson graphs emerge in coding problems [10]. For our tests on Sanchis, Mannino and Johnson graphs we take the complement graphs and indicate this by suffix _c in the tables. These classes of graphs are some of the DIMACS benchmark instances and can be downloaded from the web.

Triangulated planar graphs are perfect, thus $\vartheta(G) = \alpha(G)$, see [8]. To generate graphs with high girth, we start with a random graph $G$ and randomly replace edges of $G$ by odd paths of prescribed maximum length.

Geometric graphs are constructed as follows. First we randomly generate points in the unit square. If the euclidean distance between two points is less than a prescribed distance then an edge connecting these two points is inserted.

## 5.2 Computations

All experiments were performed starting from solving the fast model (10) as initial relaxation. Thereafter we switch to the larger semidefinite formulation (8) because we do not know how to include $MET(G)$ in (10).

The computational results on the seven problem sets are reported in Tables 3 to 9. Column $n$ gives the number of vertices, column $m$ the number of edges in the graph G. The column labeled *lbd on* $\alpha(G)$ stands for "lower bound on $\alpha(G)$". This column gives the size of the largest stable set which we have found using our heuristic method sketched in the previous section. The remaining columns contain the upper bounds found by our code. In addition to the value of the upper bound, we also provide the gap in % with respect to the values given in column *lbd on* $\alpha(G)$, separated from the upper bound by a dash. The numbers given in parenthesis represent the number of active inequality constraints of the final relaxation. Two numbers separated by a slash within one parenthesis show remaining odd circuit and triangle inequalities separately.

We first note that for triangulated planar graphs, which are perfect, the computation of $\alpha$ is easy, see Table 7. The running times to obtain $\vartheta(G)$ are reasonably moderate for these graphs, but still much larger than the time needed to optimize over $CSTAB(G)$. We note that already the linear bound on $\alpha(G)$ is often tight. These findings show that the semidefinite approach for this class of graphs is not advisable. (It should be added that $\alpha(G) > \frac{n}{3}$ for all tested triangulated planar graphs.)

The results in Table 3 and Table 8 illustrate that optimizing over $F(G) \cap C(G)$ leads to substantial improvements of $\vartheta(G)$. This optimization is carried out exactly, i.e. no cycle inequalities are violated at the final solution.

The inclusion of additional inequalities from $MET(G)$ leads to further improvements, as can be seen from the last columns in these tables, however associated with an enormous increase of computational cost (see Table 2). We did not iterate until all triangle inequalities are satified, because this is computationally too involved. The results in the column labeled $F(G) \cap C(G) \cap MET(G)$ are obtained by the following stopping conditions: First, all cycle inequalities are satisfied. Secondly, the maximum violation of triangle inequalities has follen below below a threshold of .03 (in the test sets of dimension 50 to 300 in Table 3). We also provide the number of active odd cycle and triangle constraints at termination. It should be noted that only a few odd cycle constraints are active in the final solution. We have found the optimal solution for many of the test problems with this approach. This is denoted by an asterisk in column *lbd on* $\alpha(G)$.

It is also interesting to see that the class of Sanchis graphs seems to be easy to approximate through $\vartheta(G)$. This is in contrast to several heuristic approaches, which have difficulties in finding large stable sets in these graphs, see [3]. Table 5 shows our experiments on the class of Mannino graphs. We observed that no odd circuits get violated by the optimal $\bar{X}$ to (8). Hence the relaxation $TH(G)$ cannot be further strenghtened by considering the odd circuits. The violation of the triangle inequalities is so small that a substantial improvement of the upper bound on $\alpha(G)$ through including the violated ones may not be expected. Therefore our experiments on the class of Mannino graphs are restricted on optimizing over $TH(G)$. Anyhow, in the leading two cases our rounding heuristic applied to the optimal $\bar{X}$ to (8) yields a stable set with cardinality $\alpha(G)$. The size of the largest stable set in the remaining instance is also very close to $\alpha(G)$. This is again an improvement compared to the heuristic approaches proposed in [3]. In the Tables 4 and 5 the column headed with BPG contains the size of the largest clique obtained by Bomze et al. [3].

Table 6 shows that optimizing over $TH(G)$ gives tight bounds on $\alpha(G)$ for the Johnson graphs.

We ran extensive tests on geometric graphs of different sizes and density. It turned out that $\vartheta(G)$ was tight in all instances. We present a choice of test results for this class of graphs in Table 9. The column headed by $\delta$ in this table contains the distances needed for the construction of these graphs, see section 5.1.

# 6 Concluding Comments

In this paper, we presented a semidefinite interior-point cutting plane approach for approximating the maximum stable set problem. In section 2.2 we have investigated two different semidefinite relaxations for the maximum stable set problem. We have provided a simple argument for the well known fact that these two models are equivalent. In particular, we have shown how optimal solutions from one formulation can be transformed into optimal solutions of the other formulation.

Our computational tests give rise to the following conclusions. The essential message is that the process of adding and dropping cutting planes works robust and can significantly improve the quality of the final solution. This was demonstrated on many different categories of test instances. Optimizing over $F(G) \cap C(G) \cap MET(G)$ produces the widest progress. The computational effort however increases dramatically with the number of cutting planes. Moreover, no practically efficient method is known to optimize exactly over this set, even though in principle this can be done in polynomial time. In contrast, optimizing over $F(G) \cap C(G)$ can be done exactly and fast. Optimizing over this set is not much harder than computing $\vartheta(G)$, which can be done routinely for graphs with up to several thousand edges.

A disadvantage of our approach is the large memory requirement for large scale problems. This comes from the Cholesky factorization used in our implementation. A promising way for future research would be to examine the relaxations in the context of Branch and Bound methods or bundle methods.

| | | lbd on | optimizing over | | | | |
|---|---|---|---|---|---|---|---|
| $n$ | $m$ | $\alpha(G)$ | $F(G)$ | $F(G) \cap C(G)$ | | $F(G) \cap C(G) \cap MET(G)$ | |
| 50 | 251 | 14* | 15.0016 - 7.15 | 14.9990 - 7.14 | (1) | 14.6960 - 4.97 | (1/130) |
| 80 | 320 | 27* | 28.1924 - 4.42 | 28.1053 - 4.09 | (20) | 27.5501 - 2.04 | (8/315) |
| 100 | 393 | 35 | 37.0451 - 5.84 | 36.8770 - 5.36 | (21) | 36.2010 - 3.43 | (7/320) |
| 100 | 434 | 33 | 35.1823 - 6.61 | 35.0532 - 6.22 | (21) | 34.3682 - 4.15 | (3/361) |
| 150 | 451 | 59 | 62.2523 - 5.51 | 61.4944 - 4.23 | (77) | 60.6595 - 2.81 | (37/694) |
| 150 | 552 | 55 | 58.0287 - 5.51 | 57.6335 - 4.79 | (49) | 56.5200 - 2.76 | (21/710) |
| 170 | 383 | 77* | 79.3290 - 3.02 | 77.6756 - 0.88 | (60) | 77.5463 - 0.71 | (44/318) |
| 170 | 423 | 75* | 76.7735 - 2.36 | 75.6992 - 0.93 | (56) | 75.4973 - 0.66 | (40/305) |
| 200 | 418 | 93* | 94.5638 - 1.68 | 93.2256 - 0.24 | (65) | 93.0111 - 0.01 | (52/401) |
| 200 | 563 | 81 | 85.5002 - 5.56 | 84.3313 - 4.11 | (99) | 83.1449 - 2.65 | (39/1059) |
| 300 | 894 | 121 | 129.3327 - 6.89 | 127.7402 - 5.57 | (153) | 126.4218 - 4.48 | (55/1195) |
| 350 | 994 | 146 | 152.5927 - 4.52 | 150.3067 - 2.95 | (183) | 150.1405 - 2.84 | (122/213) |
| 400 | 835 | 191* | 193.4866 - 1.30 | 191.0000 - 0.00 | (132) | 191.1058 - 0.06 | (72/5) |

Table 3: *Numerical Results on randomly generated sparse graphs. The relative error in % with respect to the values given in column lbd on $\alpha(G)$ is separated from the upper bound by a dash. The numbers given in parenthesis represent the number of active inequality constraints of the final relaxation. Two numbers separated by a slash within one parenthesis show remaining odd circuit and triangle inequalities separately.*

## References


[1] E. Balas, S.Ceria, and G. Cornuéjols. A lift project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, 58(3):195–324, 1993.

[2] F. Barahona. On cuts and matchings in planar graphs. *Mathematical Programming*, 36(1):53–68, 1993.

[3] I. M. Bomze, M. Pelillo, and R. Giacomini. Evolutionary approach to the maximum clique problem: Empirical evidence on a larger scale. In I. M. Bomze, T. Csendes, R. Horst, and P. M. Pardalos, editors, *Developments in Global Optimization*, pages 95–108. Kluwer Academic Publishers, Dordrecht, 1997.

[4] V. Chvátal. On certain polytopes associated with graphs. *J. Combin. Theory B*, 18:138–154, 1975.

[5] D. R. Fulkerson. On the perfect graph theorem. In T. C. Hu and S. M. Robinson, editors, *Mathematical Programming*, pages 69–76. Academic Press, New York, 1973.

[6] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[7] M. Grötschel, L. Lovász, and A. Schrijver. Relaxations of vertex packing. *Journal of Combinatorial Theory*, 40:330–343, 1986.

[8] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, New York, 1988.

| | | | | | lbd on | optimizing over | |
|---|---|---|---|---|---|---|---|
| File | $n$ | $m$ | $\alpha(G)$ | BPG | $\alpha(G)$ | $F(G)$ | $F(G) \cap C(G) \cap MET(G)$ |
| san200_0.9_1.clq_c | 200 | 1990 | 70 | 46 | 70 | 70.0000 | - |
| san200_0.9_2.clq_c | 200 | 1990 | 60 | 36 | 60 | 60.0000 | - |
| san200_0.9_3.clq_c | 200 | 1990 | 44 | 33 | 44 | 44.0000 | - |
| sanr200_0.9.clq_c | 200 | 2037 | - | 41 | 42 | 49.2735 | 48.8689 (0/294) |

Table 4: *Numerical Results on Sanchis graphs*

| | | | | | | optimizing over |
|---|---|---|---|---|---|---|
| File | $n$ | $m$ | $\alpha(G)$ | BPG | *lbd on* $\alpha(G)$ | $F(G)$ |
| MANN_a9.clq_c | 45 | 72 | 16 | 16 | 16 | 17.4750 |
| MANN_a27.clq_c | 378 | 702 | 126 | 121 | 126 | 132.7629 |
| MANN_a45.clq_c | 1035 | 1980 | 345 | 336 | 343 | 356.0488 |

Table 5: *Numerical Results on Mannino graphs*

| | | | | | optimizing over |
|---|---|---|---|---|---|
| File | $n$ | $m$ | $\alpha(G)$ | *lbd on* $\alpha(G)$ | $F(G)$ |
| johnson8-2-4.clq_c | 28 | 168 | 4 | 4 | 4.0 |
| johnson8-4-4.clq_c | 70 | 560 | 14 | 14 | 14.0 |
| johnson8-16-2-4.clq_c | 120 | 1680 | 8 | 8 | 8.0 |

Table 6: *Numerical Results on Johnson graphs*

| | | | optimizing over | | | |
|---|---|---|---|---|---|---|
| $n$ | $m$ | $\alpha(G)$ | $F(G)$ | CPU time [sec] | $CSTAB(G)$ | CPU time [sec] |
| 100 | 294 | 44 | 44.0000 | 2.2698 | 44.6667 | 0.9380 |
| 200 | 594 | 87 | 87.0000 | 16.5672 | 87.3333 | 4.5259 |
| 300 | 894 | 126 | 126.0000 | 53.6591 | 127.0000 | 13.0489 |
| 400 | 1194 | 174 | 174.0000 | 155.8248 | 174.6667 | 29.7529 |
| 500 | 1494 | 220 | 220.0000 | 335.4370 | 220.3333 | 64.3262 |
| 600 | 1794 | 270 | 270.0000 | 606.9892 | 270.0000 | 59.0361 |

Table 7: *Numerical Results on triangulated planar graphs*

| | | lbd on | optimizing over | | | | |
|---|---|---|---|---|---|---|---|
| $n$ | $m$ | $\alpha(G)$ | $F(G)$ | $F(G) \cap C(G)$ | | $F(G) \cap C(G) \cap MET(G)$ | |
| 118 | 152 | 57* | 58.6107 - 2.83 | 57.4000 - 0.70 | (18) | 57.3252 - 0.57 | (14/83) |
| 292 | 365 | 141* | 144.2935 - 2.34 | 142.6667 - 1.18 | (86) | 141.9742 - 0.69 | (10/181) |
| 280 | 394 | 134* | 137.1680 - 2.36 | 136.6476 - 1.98 | (94) | 134.4599 - 0.34 | (3/215) |
| 282 | 339 | 136 | 139.5288 - 2.59 | 137.6923 - 1.24 | (19) | 137.7490 - 1.29 | (16/120) |
| 372 | 534 | 179* | 181.2776 - 1.27 | 179.0000 - 0.00 | (96) | 179.0000 - 0.00 | (104/370) |
| 448 | 465 | 220 | 223.8245 - 1.74 | 222.3333 - 1.06 | (20) | 222.3333 - 1.06 | (19/1) |
| 520 | 947 | 245* | 250.1892 - 2.12 | 250.1473 - 2.10 | (7) | 245.9738 - 0.40 | (0/354) |
| 544 | 688 | 264 | 269.0532 - 1.91 | 267.0000 - 1.14 | (104) | 266.1425 - 0.81 | (24/254) |

Table 8: *Numerical Results on graphs with long odd circuits. The relative error in % with respect to the values given in column lbd on $\alpha(G)$ is separated from the upper bound by a dash. The numbers given in parenthesis represent the number of active inequality constraints of the final relaxation. Two numbers separated by a slash within one parenthesis show remaining odd circuit and triangle inequalities separately.*


[9] G. Gruber. *On Semidefinite Programming and Applications in Combinatorial Optimization.* PhD thesis, University of Technology, Graz, Austria, 2000. Shaker Verlag, Aachen - Maastricht, ISBN 3-8265-7541-5.

[10] J. Hasselberg, Panos M. Pardalos, and G. Vairaktarakis. Test case generators and computational results for the maximum clique problem. *Journal of Global Optimization*, 3:463–482, 1993.

[11] C. Helmberg, S. Poljak, F. Rendl, and H. Wolkowicz. Combining semidefinite and polyhedral relaxations for integer programs. In E. Balas and J. Clausen, editors, *Integer Programming and Combinatorial Optimization (Proceedings of the 4th International IPCO Conference, Copenhagen, Denmark, May 1995)*, volume 920 of *Lecture Notes in Computer Science*, pages 124–134. Springer Verlag, Berlin, Germany, 1995.

[12] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior point method for semidefinite programming. *SIAM Journal on Optimization*, 6(2):342–361, 1996.

[13] M. Kojima, S. Shindoh, and S. Hara. Interior point methods for the monotone linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization*, 7(1):86–125, 1997.

[14] C. Lemaréchal and F. Oustry. Semidefinite Relaxations and Lagrangian Duality with Application to Combinatorial Optimization. Technical report, Institut National de Recherche en Informatique et en Automatique, INRIA, St Martin, France, 1999.

[15] L. Lovász. On the shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, 1979.

[16] L. Lovász. Semidefinite programs and combinatorial optimization. Lecture Notes, 1995.

[17] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal of Optimization*, 1(2):166–190, 1991.

| | | | lbd on | optimizing over |
|---|---|---|---|---|
| $n$ | $m$ | $\delta$ | $\alpha(G)$ | $F(G)$ |
| 50 | 128 | 0.2 | 16 | 16.0000 |
| 50 | 476 | 0.4 | 7 | 7.0000 |
| 50 | 698 | 0.6 | 4 | 4.0000 |
| 100 | 185 | 0.1 | 39 | 39.0000 |
| 100 | 561 | 0.2 | 21 | 21.0000 |
| 100 | 1065 | 0.3 | 13 | 13.0000 |
| 150 | 327 | 0.1 | 53 | 53.0000 |
| 150 | 1137 | 0.2 | 22 | 22.0000 |
| 150 | 2358 | 0.3 | 13 | 13.3710 |

Table 9: *Numerical Results on Geometric graphs*

[18] M. Laurent, S. Poljak, F. Rendl. Connections between semidefinite relaxations of the max-cut and stable set problems. *Mathematical Programming*, 77(2):225–246, 1997.

[19] C. Mannino and A. Sassano. Edge projection and the maximum cardinality stable set problem. In David S. Johnson and Michael A. Trick, editors, *Cliques, Coloring, and Satisfiability*, volume 26 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 205–219. AMS, 1996.

[20] R. D.C. Monteiro. Primal-dual path-following algorithms for semidefinite programming. *SIAM Journal on Optimization*, 7(3):636–678, 1997.

[21] Y. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming : Theory and Algorithms.* SIAM Publications. SIAM, Philadelphia, USA, 1993.

[22] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for $(0,1)$-quadratic programming. *Journal of Global Optimization*, 7(1):51–73, 1995.

[23] L. A. Sanchis. Test case construction for the vertex cover problem. In *Computational Support for Discrete Mathematics*, volume 15 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1052–1798. American Mathematical Society, 1994.

[24] A. Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, IT-25:425–429, 1979.

[25] H. D. Sherali and W. P. Adams. A hierarchy of relaxation between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal Discrete Mathematics*, 3(3):411–430, 1990.

[26] C. De Simone. The cut polytope and the boolean quadric polytope. *Discrete Applied Mathematics*, 79:71–75, 1989.