Math. Program., Ser. A (2009) 120:381–401
DOI 10.1007/s10107-008-0219-8

FULL LENGTH PAPER

An application of the Lovász–Schrijver $M(K, K)$ operator to the stable set problem

Monia Giandomenico · Adam N. Letchford · Fabrizio Rossi · Stefano Smriglio

Received: 18 January 2007 / Revised: 29 February 2008 / Published online: 11 April 2008

Abstract Although the lift-and-project operators of Lovász and Schrijver have been the subject of intense study, their $M(K, K)$ operator has received little attention. We consider an application of this operator to the stable set problem. We begin with an initial linear programming (LP) relaxation consisting of clique and non-negativity inequalities, and then apply the operator to obtain a stronger extended LP relaxation. We discuss theoretical properties of the resulting relaxation, describe the issues that must be overcome to obtain an effective practical implementation, and give extensive computational results. Remarkably, the upper bounds obtained are sometimes stronger than those obtained with semidefinite programming techniques.

Keywords Lovász–Schrijver operators · Stable set problem · Semidefinite programming

Mathematics Subject Classification (2000) 90C27

M. Giandomenico · F. Rossi · S. Smriglio
Dipartimento di Informatica, Università Di L'Aquila, Via Vetoio, 67010 Coppito, AQ, Italy
e-mail: giandomenico@di.univaq.it

F. Rossi
e-mail: rossi@di.univaq.it

S. Smriglio
e-mail: smriglio@di.univaq.it

A. N. Letchford (✉)
Department of Management Science, Lancaster University Management School, Lancaster, LA1 4YX, UK
e-mail: A.N.Letchford@lancaster.ac.uk

1 Introduction

Let $G = (V, E)$ be an undirected graph, where V is the vertex set and E is the edge set, and let n denote $|V|$. A vertex set $S \subseteq V$ is called *stable* if the vertices in S are pairwise non-adjacent. The *stable set problem* calls for a stable set of maximum cardinality, or, if we are also given a weight vector $w \in \mathbb{Q}^n_+$, of maximum weight. The stable set problem is a well-known and fundamental combinatorial optimization problem, with many applications, for example in timetabling and scheduling. It is equivalent to the well-known *max-clique* problem.

The stable set problem is $\mathcal{NP}$-hard in the strong sense, and hard even to approximate [14]. This theoretical hardness is also borne out in practice: even with modern algorithms and computers, some instances with only 300 vertices or so are remarkably hard to solve to proven optimality. In particular, algorithms based on *Linear Programming* (LP) have so far given disappointing results. Indeed, for unweighted instances, relatively simple combinatorial algorithms (such as those of Régin [27] and Tomita and Kameda [30]) perform nearly as well as sophisticated LP-based algorithms (such as those of Nemhauser and Sigismondi [23] and Rossi and Smriglio [28]).

In a seminal paper, Lovász [20] proposed to use *Semidefinite Programming* (SDP) to compute upper bounds for the stable set problem. His SDP relaxation, called the *theta* relaxation, was studied in depth in Grötschel et al. [12]. Several researchers have performed computational experiments either with the theta relaxation or with stronger relaxations obtained by adding valid linear inequalities (e.g. [7,13,32]).

Another landmark paper was Lovász and Schrijver [21], which introduced several 'operators' that enable one to take the LP relaxation of any 0–1 LP and form stronger LP or SDP relaxations in spaces of higher dimension. Lovász and Schrijver applied two of their operators to the stable set problem: the N operator (based on LP) and the N_+ operator (based on SDP). Theoretically, the N_+ operator turned out to yield a much stronger relaxation than the N operator. Computational experiments with the two operators have been conducted by Balas et al. [1] and Burer and Vandenbussche [3].

Since [21] appeared, a huge number of papers have been written on the application of the Lovász–Schrijver operators to various combinatorial optimization problems. There is however another operator in [21] that has received little attention: the so-called $M(K, K)$ operator. Like the N operator, the $M(K, K)$ operator is based on LP rather than SDP. Roughly speaking, it 'squares' the size of a linear system by multiplying pairs of constraints together.

In this paper, we apply the $M(K, K)$ operator to the stable set problem. Our application is rather non-conventional, in that our initial LP relaxation is already of exponential size, consisting of the well-known *clique* and *non-negativity* inequalities. The resulting LP relaxation is also of exponential size, and $\mathcal{NP}$-hard to solve. Nevertheless, we show that it can be solved approximately, to a reasonable degree of accuracy, for many instances of interest. We discuss theoretical properties of this LP relaxation, which turns out to be remarkably tight. We also describe the algorithmic issues that must be overcome to solve it approximately, and give extensive computational results. Interestingly, the upper bounds obtained from our LP relaxation are sometimes stronger than those obtained with the best SDP techniques.

The remainder of the paper is structured as follows. In Sect. 2, we review the relevant literature on LP and SDP relaxations. In Sect. 3, we explain the $M(K, K)$ operator in detail, and prove some theoretical results that establish the strength of the resulting LP relaxation. This is done by projection, in the spirit of papers by Laurent et al. [18] and Giandomenico and Letchford [11]. In Sect. 4, we discuss implementation issues, and give extensive computational results on standard benchmark graphs. Finally, concluding remarks are given in Sect. 5.

2 Review of known results

2.1 Standard formulation and inequalities

The (weighted) stable set problem can be formulated as the following 0–1 LP:

$$\begin{aligned} &\max \sum_{i \in V} w_i x_i \\ &\text{s.t. } x_i + x_j \leq 1 \ (\forall \{i, j\} \in E) \\ &x \in \{0, 1\}^n. \end{aligned} \tag{1}$$

The inequalities (1) are commonly called *edge* inequalities. The *stable set polytope*, often denoted by STAB(G), is the convex hull in $\mathbb{R}^n_+$ of the incidence vectors of stable sets, i.e.

$$\text{STAB}(G) = \text{conv}\left\{x \in \{0, 1\}^n : (1) \text{ hold}\right\}.$$

The study of STAB(G) was initiated by Padberg [24], who observed the following:

– For any $i \in V$, the *non-negativity* inequality $x_i \geq 0$ is facet-inducing.
– For any maximal *clique* (set of pairwise adjacent vertices) $C \subset V$, the *clique* inequality $\sum_{i \in C} x_i \leq 1$ is facet-inducing.
– Given any $H \subset V$ inducing a simple cycle of odd cardinality, the *odd cycle* inequality $\sum_{i \in H} x_i \leq \left\lfloor \frac{|H|}{2} \right\rfloor$ is valid. (When $|H| \geq 5$ and the cycle is chordless, the inequality is called an *odd hole* inequality.)
– Given any $H \subset V$ inducing an *odd antihole* (i.e. the complement of an odd hole), the *odd antihole* inequality $\sum_{i \in A} x_i \leq 2$ is valid.
– The odd hole and odd antihole inequalities are not facet-inducing in general and can often be strengthened by *lifting*. For example, if H induces an odd hole in G and $j \notin H$ is adjacent to all vertices in H, then the *odd wheel* inequality $\sum_{i \in H} x_i + \left\lfloor \frac{|H|}{2} \right\rfloor x_j \leq \left\lfloor \frac{|H|}{2} \right\rfloor$ is valid.

The polytope defined by the edge and non-negativity inequalities is usually called the *fractional stable set polytope* and denoted by FRAC(G). The polytope defined by the clique and non-negativity inequalities is usually denoted by QSTAB(G). Clearly, we have STAB(G) $\subseteq$ QSTAB(G) $\subseteq$ FRAC(G), and inclusion is generally strict [12].

Trotter [31] introduced the *web* and *antiweb* inequalities, which include clique, odd hole and odd antihole inequalities as special cases. Let p and q be integers satisfying $p > 2q+1$ and $q > 1$. Here, arithmetic modulo p is used. A (p, q)-*web* is a graph with vertex set $\{1, \ldots, p\}$ and with edges from i to $\{i+q, \ldots, i-q\}$, for every $1 \leq i \leq p$. A

(p, q)-*antiweb* is the complement of a (p, q)-web. The web inequalities take the form $\sum_{i \in W} x_i \le q$ for every vertex set W inducing a (p, q)-web, and the antiweb inequalities take the form $\sum_{i \in AW} x_i \le \lfloor p/q \rfloor$ for every vertex set AW inducing a (p, q)-antiweb. Web and antiweb inequalities may again need to be lifted to obtain facets.

For the sake of brevity, we do not review the other known classes of valid inequalities for STAB(G).

Note that the clique, odd hole, odd antihole, web and antiweb inequalities can all be exponential in number. Thus, to use them as cutting planes, one needs a *separation algorithm* (see again [12]). Gerards and Schrijver [10] gave a polynomial-time separation algorithm for odd cycle inequalities; Grötschel et al. [12] did the same for odd wheel inequalities. The complexity of odd antihole, web and antiweb separation is unknown, although Cheng and De Vries [4] gave a polynomial-time algorithm for antiweb inequalities with fixed q.

The separation problem for clique inequalities is strongly NP-hard (e.g. Grötschel et al. [12]). However, some effective separation heuristics are known for clique, lifted odd hole and antihole inequalities [2,15,23], and for the so-called *rank* inequalities, which include the web and antiweb inequalities as a special case [28].

Much more powerful separation results can be obtained using SDP—see the following subsections.

These polyhedral results have been used in exact algorithms for the stable set problem. Nemhauser and Sigismondi [23] described a cut-and-branch algorithm based on clique and lifted odd hole inequalities, and, more recently, Rossi and Smriglio [28] presented a branch-and-cut algorithm based on general rank inequalities. Although such algorithms perform reasonably well, they can still run into difficulties when the number of vertices exceeds around 300, especially when the graph is relatively sparse.

2.2 The Lovász theta relaxation

We now describe the famous *theta* relaxation of the stable set problem, due to Lovász [20]. For ease of exposition, we follow the presentation of Lovász and Schrijver [21]. We introduce, for all $\{i, j\} \subset V$, the quadratic variable x_{ij}, representing the product $x_i x_j$. Note that $x_{ji} = x_{ij}$ for all $\{i, j\} \subset V$ and $x_{ii} = x_i$ for all $i \in V$. Now, let $X = xx^{\mathrm{T}}$ be the $n \times n$ matrix in which the entry in row i and column j is x_{ij}. Also, let Y be an augmented matrix representing the product $\binom{1}{x}\binom{1}{x}^{\mathrm{T}}$. That is,

$$Y := \begin{pmatrix} 1 & x^{\mathrm{T}} \\ x & X \end{pmatrix}.$$

Since Y is the product of a real matrix and its transpose, it is real, symmetric, square and positive semidefinite (psd). Then, an upper bound for the stable set problem is given by:

$$\begin{aligned} \max \ & \textstyle\sum_{i \in V} w_i x_i \\ \text{s.t.} \ & x_i = x_{ii} \quad (i \in V) \\ & x_{ij} = 0 \quad (\{i, j\} \in E) \\ & Y \in S_+^{n+1}, \end{aligned}$$

where S_+^{n+1} denotes the cone of real symmetric square psd matrices of order $n + 1$. This upper bound is denoted by $\theta(G, w)$ (or just $\theta(G)$ in the unweighted case).

Grötschel et al. [12] denote by TH(G) the projection of the feasible region of this SDP relaxation onto the subspace defined by the original (non-quadratic) variables. TH(G) is convex, but not polyhedral in general. Remarkably, we have STAB(G) $\subseteq$ TH(G) $\subseteq$ QSTAB(G), with equality if and only if G is perfect. Since SDP can be solved to arbitrary precision in polynomial time, this implies that there exists a polynomial-time separation algorithm for a class of inequalities which includes all clique inequalities. This is so despite the fact that clique separation itself is strongly NP-hard.

The bound $\theta(G, w)$ is quite strong in practice (see, e.g. [13,32]), and there exist classes of graphs for which it is much stronger than the bound obtained by using non-negativity and clique inequalities [17]. The current fastest methods for computing $\theta(G, w)$ appear to be the augmented Lagrangian algorithm of Povh et al. [26] and the regularization method of Malick et al. [22].

2.3 Combining linear and semidefinite relaxations

Stronger formulations in the extended space can be obtained by adding valid linear inequalities to the Lovász SDP relaxation. Schrijver [29] suggested adding the non-negativity inequalities $x_{ij} \geq 0$ for all $\{i, j\} \notin E$, which are not implied by the condition $Y \in S_+^{n+1}$. The resulting upper bound is often called $\theta'(G, w)$.

As we mentioned in the introduction, Lovász and Schrijver [21] defined several general operators for strengthening LP relaxations of 0–1 LPs. Applying their so-called M_+ operation to FRAC(G), one obtains the polytope M_+(FRAC(G)). This is formed by adding the following inequalities to the above-mentioned relaxation of Schrijver:

$$x_{ik} + x_{jk} \leq x_k \qquad (\{i, j\} \in E, k \neq i, j), \tag{2}$$

$$x_i + x_j + x_k \leq 1 + x_{ik} + x_{jk} \quad (\{i, j\} \in E, k \neq i, j). \tag{3}$$

The projection of M_+(FRAC(G)) onto the non-quadratic space is called N_+(FRAC(G)). Lovász and Schrijver showed that N_+(FRAC(G)) satisfies all clique, odd cycle, odd antihole and odd wheel inequalities. Giandomenico and Letchford [11] showed that, in fact, it satisfies all web inequalities. Thus, SDP provides a polynomial-time separation algorithm for a class of inequalities which includes all web and odd wheel inequalities (and therefore all clique, odd hole and odd antihole inequalities).

Computational results obtained by optimising over M_+(FRAC(G)) have been given for example by Balas et al. [1] (using lift-and-project cutting plane methods) and Burer and Vandenbussche [3] (using an augmented Lagrangian method). The upper bounds are often noticeably better than $\theta'(G, w)$, but at the expense of very large running times. Indeed, for many instances, the algorithms presented in [1,3] could not optimise over M_+(FRAC(G)) to within any meaningful accuracy within a reasonable time.

An even stronger relaxation can be obtained from a consideration of the so-called *boolean quadric polytope* (see, e.g. [25]). This polytope is the convex hull of all matrices X of the form xx^{T} for some $x \in \{0, 1\}^n$. As pointed out by Padberg, STAB(G)

can be obtained by taking the face of the boolean quadric polytope defined by the equations $x_{ij} = 0$ for all $\{i, j\} \in E$, and projecting it onto the non-quadratic space. The inequalities (2) and (3), along with the non-negativity inequalities $x_{ij} \geq 0$ for all $\{i, j\} \notin E$, are then easily seen to be special cases of the well-known *triangle* inequalities, that induce facets of the boolean quadric polytope. This suggests that the relaxation $M_+(\text{FRAC}(G))$ can be strengthened by adding the remaining triangle inequalities, which are:

$$x_{ik} + x_{jk} \leq x_k + x_{ij} \quad (\forall \text{ stable } \{i, j, k\} \subset V) \tag{4}$$

$$x_i + x_j + x_k \leq 1 + x_{ij} + x_{ik} + x_{jk} \quad (\forall \text{ stable } \{i, j, k\} \subset V). \tag{5}$$

Some experiments with the resulting SDP relaxation were conducted by Gruber and Rendl [13] using an interior-point cutting-plane method. The upper bounds obtained were very good, although again at the expense of large running times.

Dukanovic and Rendl [7] examined a weakened version of the relaxation of Gruber and Rendl, in which the inhomogeneous constraints (3) and (5) are omitted. (This relaxation still dominates $\theta'(G, w)$, but is incomparable with $M_+(\text{FRAC}(G))$.) Dukanovic and Rendl showed how to exploit the special structure of this relaxation within an interior-point algorithm, to reduce the running time somewhat.

For related projection results, and connections with the well-known *max-cut* problem, see Laurent et al. [18] and Giandomenico and Letchford [11].

3 Applying the $M(K, K)$ operator to QSTAB(G)

3.1 Definition and elementary results

As we mentioned in the introduction, the Lovász–Schrijver $M(K, K)$ operator essentially amounts to 'squaring' a given linear system in 0–1 variables. Specifically, for any pair of linear inequalities $\alpha x - \beta \geq 0$ and $\alpha' x - \beta' \geq 0$, the 'product' inequality $\left(-\beta \ \alpha^{\mathrm{T}}\right) Y \begin{pmatrix} -\beta' \\ \alpha' \end{pmatrix} \geq 0$ is computed. The products $x_i x_j$, for all $1 \leq i < j \leq n$, are then replaced with new variables x_{ij}, and the terms x_i^2, for $1 \leq i \leq n$, are replaced with x_i (which is valid when x_i is binary.) This yields an extended LP formulation which is provably stronger than the original.

In this paper, we have decided to investigate $M(\text{QSTAB}(G),\text{QSTAB}(G))$; that is, the relaxation obtained by applying the $M(K, K)$ operation to the LP relaxation consisting of the non-negativity and clique inequalities. For brevity, we refer to this relaxation simply as $M(K, K)$ in what follows. We will see that, although solving the $M(K, K)$ relaxation is theoretically hard, one can solve it to reasonable accuracy in practice.

If we let Ω denote the set of all maximal cliques of G, QSTAB(G) is defined by the following linear system:

$$1 - \sum_{i \in C} x_i \geq 0 \quad (C \in \Omega) \tag{6}$$

$$x_i \geq 0 \quad (i \in V). \tag{7}$$

Applying the $M(K, K)$ operation yields the following linear system:

$$x_i - \sum_{j \in C} x_{ij} \geq 0 \qquad (C \in \Omega, i \in V) \tag{8}$$

$$1 - \sum_{i \in C} x_i - \sum_{i \in C'} x_i + \sum_{i \in C, j \in C'} x_{ij} \geq 0 \quad (C, C' \in \Omega) \tag{9}$$

$$x_{ij} \geq 0 \qquad (\{i, j\} \subset V). \tag{10}$$

Inequalities (8), (9) and (10) are obtained by multiplying a clique inequality and a non-negativity inequality, two clique inequalities, and two nonnegativity inequalities, respectively.

Note that, when $i \in C$, the inequalities (8) reduce to $\sum_{j \in C \setminus \{i\}} x_{ij} \leq 0$. Hence, $M(K, K)$ satisfies the equations $x_{ij} = 0$ for all $\{i, j\} \in E$. Thus, the linear system can be written in the following simplified form:

$$\sum_{j \in C: \{i,j\} \in \bar{E}} x_{ij} - x_i \leq 0 \qquad (C \in \Omega, i \in V \setminus C) \tag{11}$$

$$\sum_{i \in C \cup C'} x_i - \sum_{\{i,j\} \in \bar{E}(C:C')} x_{ij} \leq 1 \quad (C, C' \in \Omega) \tag{12}$$

$$x_{ij} = 0 \qquad (\{i, j\} \in E),$$

$$x_{ij} \geq 0 \qquad (\{i, j\} \in \bar{E}),$$

where $\bar{E} := \{\{i, j\} \subset V : \{i, j\} \notin E\}$ denotes the set of 'non-edges', and $\bar{E}(C : C')$ denotes $\left\{\{i, j\} \in \bar{E} : i \in C, j \in C'\right\}$.

It is easy to show that the inequalities (11) and (12) dominate the inequalities (2) and (3) that appear in the definition of $M_+(\text{FRAC}(G))$. However, they do not in general dominate the additional triangle inequalities (4), (5). In any case, since we are not imposing psd-ness on Y, in general $M(K, K)$ neither contains nor is contained in any of the convex sets mentioned in Subsects. 2.2 and 2.3.

It can also be shown that the inequalities (11) and (12) are special cases of the *rounded psd* inequalities explored by Giandomenico and Letchford [11]. In [11] it was proved that the rounded psd inequalities imply by projection all web and antiweb inequalities (and therefore all edge, clique, odd hole and odd antihole inequalities), together with various lifted versions. In the following three subsections, we show that these projection results still hold even if we restrict ourselves to the special inequalities (11) and (12).

We follow Lovász and Schrijver [21] in letting $N(K, K)$ denote the projection of $M(K, K)$ onto the subspace of the original (non-quadratic) variables.

3.2 A class of disjunctive cuts including all antiwebs

We now show that $N(K, K)$ satisfies a wide class of disjunctive cuts that includes all antiweb inequalities. For this, we will need the following result of Balas et al. [1]:

Theorem 1 [1] *Let $P = \{x \in [0, 1]^n : Ax \leq b\}$ be a polytope and let $C \subset \{1, \ldots, n\}$ be such that $\sum_{i \in C} x_i^* \leq 1$ for all $x^* \in P$. Consider the extended formulation obtained by multiplying the system $Ax \leq b$ by x_i for all $i \in C$, and by $1 - \sum_{i \in C} x_i$. The* *projection of the resulting polytope into the original space equals*

$$conv\,\{x \in P : x_i \in \{0,1\}\ (i \in C)\}.$$

This more or less immediately implies the following:

Corollary 1 *For any $C \in \Omega$, $N(K, K)$ satisfies all inequalities that are implied by the system* (6)*,* (7)*, and the following disjunction:*

$$\left(\sum_{i \in C} x_i = 0\right) \vee \left(\bigvee_{i \in C} x_i = 1\right). \tag{13}$$

Proof It suffices to let P equal QSTAB(G) in Theorem 1 and note that, regardless of the choice of $C \in \Omega$, $N(K, K)$ is contained in the projected polytope mentioned in the theorem. □

In particular, we have:

Corollary 2 *$N(K, K)$ satisfies all antiweb inequalities.*

Proof Let $AW(p, q)$ be an antiweb and let $r = p \bmod q$. Since the vertex set $\{1, 2, \ldots, q\}$ forms a maximal clique, it suffices to show that the antiweb inequality $\sum_{i \in AW} x_i \le \lfloor p/q \rfloor$ is implied by the clique and non-negativity inequalities and the following disjunction:

$$\left(\sum_{i=1}^{q} x_i = 0\right) \vee (x_1 = 1) \vee \cdots \vee \left(x_q = 1\right).$$

All points in QSTAB(G) satisfying the first term of the disjunction clearly satisfy $\sum_{i=1}^{r} x_i \le 0$. This, together with the clique inequalities $\sum_{i=sq+r+1}^{(s+1)q+r} x_i \le 1$ for $s = 0, \ldots, \lfloor p/q \rfloor - 1$, implies the antiweb inequality. Similarly, all points in QSTAB(G) satisfying $x_1 = 1$ clearly satisfy $\sum_{i=2}^{q} x_i \le 0$ and $\sum_{i=p-r+1}^{p-1} x_i \le 0$. These, together with the clique inequalities $\sum_{i=sq+1}^{(s+1)q} x_i \le 1$ for $s = 1, \ldots, \lfloor p/q \rfloor - 1$, imply the antiweb inequality. The other terms of the disjunction are handled analogously by symmetry. □

3.3 Web inequalities

As mentioned in Subsect. 2.3, Giandomenico and Letchford [11] proved that the Lovász–Schrijver relaxation $N_+(G)$ satisfies all web inequalities. We will now show that this also holds for $N(K, K)$. Note that this is not a corollary of Theorem 1. Indeed, one can show that the (p, q)-web inequality arises from a disjunction of the form (13) only if $p \bmod q \le \lfloor p/q \rfloor$.

In what follows, we will refer to inequalities of the form (11) and (12) as *clique-variable* and *clique-product* inequalities, or just *CVIs* and *CPIs*, respectively. Note that the CVIs and CPIs remain valid for $M(K, K)$ even if C and/or C' are not maximal cliques in G. We will also use the notation $\omega = \lfloor p/q \rfloor$ and $r = p \bmod q$. Note that ω is the cardinality of a maximum clique in the web $W(p, q)$ and that $p = \omega q + r$.

Lemma 1 *Let $G = W(p, q)$ be a web. For any $j \in \{1, \ldots, q-1\}$, the inequality*

$$\sum_{i=1}^{p} x_{i,i+j} + \sum_{i=1}^{p} x_{i+j,i+q} \le \sum_{i=1}^{p} x_i \tag{14}$$

is implied by the CVIs.

Proof For any fixed $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, q-1\}$, we have the trivial CVI $x_{i,i+j} + x_{i+j,i+q} \le x_{i+j}$. Summing these CVIs over all i yields (14). □

Lemma 2 *Let $G = W(p, q)$ be a web. For any $j \in \{1, \ldots, r-1\}$, the inequality*

$$(\omega+2) \sum_{i=1}^{p} x_i - \sum_{i=1}^{p} x_{i,i+j} - \sum_{i=1}^{p} x_{i+j,i+r} \le p \tag{15}$$

is implied by the CPIs.

Proof Let $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, r-1\}$ be fixed. If we let $C = \{i, i+q, \ldots, i+(\omega-1)q\}$ and $C' = \{i-r-q+j, i-r+j\}$, the CPI (12) reduces to:

$$\sum_{i \in C \cup C'} x_i - x_{i-r-q,i-r-q+j} - x_{i,i-r+j} \le 1.$$

(To see this, note that $i + (\omega-1)q \bmod p = i - r - q$.) Summing these CPIs over all i yields (15). □

Lemma 3 *Let $G = W(p, q)$ be a web. For any $j \in \{1, \ldots, q-r-1\}$, the inequality*

$$(\omega+1) \sum_{i=1}^{p} x_i - \sum_{i=1}^{p} x_{i-r,i+j} - \sum_{i=1}^{p} x_{i+j,i+q} \le p \tag{16}$$

is implied by the CPIs.

Proof Let $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, q-r-1\}$ be fixed. If we let $C = \{i, i+q, \ldots, i+(\omega-1)q\}$ and $C' = \{i-q+j\}$, the CPI (12) reduces to:

$$\sum_{i \in C \cup C'} x_i - x_{i-q-r,i-q+j} - x_{i,i-q+j} \le 1.$$

Summing these CPIs over all i yields (16). □

Lemma 4 *Let $G = W(p, q)$ be a web. The inequality*

$$(\omega + 1) \sum_{i=1}^{p} x_i - \sum_{i=1}^{p} x_{i,i+r} \leq p \tag{17}$$

is implied by the CPIs.

Proof Let $i \in \{1, \ldots, p\}$ be fixed. If we let $C = \{i, i+q, \ldots, i+(\omega-1)q\}$ and $C' = \{i - q\}$, the CPI (12) reduces to:

$$\sum_{i \in C} x_i + x_{i-q} - x_{i-q,i-r-q} \leq 1.$$

Summing these CPIs over all i yields (17). □

Theorem 2 *$N(K, K)$ satisfies all web inequalities.*

Proof If we sum together the inequalities (14) over all $j \in \{1, \ldots, q-1\}$, and simplify, we obtain:

$$2 \sum_{i=1}^{p} \sum_{j=1}^{q-1} x_{i,i+j} \leq (q-1) \sum_{i=1}^{p} x_i. \tag{18}$$

If we sum together the inequalities (15) over all $j \in \{1, \ldots, r-1\}$, and simplify, we obtain:

$$(r-1)(\omega+2) \sum_{i=1}^{p} x_i - 2 \sum_{i=1}^{p} \sum_{j=1}^{r-1} x_{i,i+j} \leq p(r-1). \tag{19}$$

If we sum the inequalities (16) over all $j \in \{1, \ldots, q-r-1\}$, and simplify, we obtain:

$$(q-r-1)(\omega+1) \sum_{i=1}^{p} x_i - 2 \sum_{i=1}^{p} \sum_{j=r+1}^{q-1} x_{i,i+j} \leq p(q-r-1). \tag{20}$$

Finally, summing together (18), (19), (20) and two times (17), and simplifying, we obtain $\sum_{i=1}^{p} px_i \leq pq$, which is equivalent to the web inequality $\sum_{i=1}^{p} x_i \leq q$. □

3.4 Sequential lifting

In Giandomenico and Letchford [11], a certain sequential lifting procedure was introduced for the stable set problem, and it was proved that, if a valid inequality for STAB(G) is implied by the rounded psd inequalities, then so is any inequality obtained by applying the lifting procedure. In this subsection, we prove that the same result holds for $N(K, K)$.

Let $G = (V, E)$ be a graph, let $\alpha^T x \le \beta$ be a valid inequality for STAB(G), and let S be a stable set in G. For a given vertex $i \in S$, we denote by $n(i)$ the set of *neighbours* of i, and let $n(S) := \bigcup_{i \in S} n(i)$. Now consider a new graph $\tilde{G} = (\tilde{V}, \tilde{E})$ obtained from G by adding an extra vertex (u, say) which is adjacent to every vertex in $S \cup n(S)$. We construct a valid inequality $\tilde{\alpha}^T x \le \beta$ for STAB($\tilde{G}$) by setting $\tilde{\alpha}_i = \alpha_i$ for all $i \in V$ and $\tilde{\alpha}_u = \sum_{i \in S} \alpha_i$. We say that the inequality $\tilde{\alpha}^T x \le \beta$ has been obtained from $\alpha^T x \le \beta$ by *lifting on* S. (When $|S| = 1$, this lifting operation reduces to the classical *replication* operation studied for example by Lovász [19] and Fulkerson [8].)

Theorem 3 *Let $G = (V, E)$ be a graph, let $\alpha^T x \le \beta$ be implied by the CVIs and CPIs, and let S be a stable set in G. The lifted inequality $\tilde{\alpha}^T x \le \beta$ for $STAB(\tilde{G})$, obtained by lifting on S, is also implied by the CVIs and CPIs.*

Proof Let us suppose that the inequality $\alpha^T x \le \beta$ is a non-negative linear combination of a family R of CVIs, of the form

$$\sum_{j \in C_r : \{i_r, j\} \in \bar{E}} x_{i_r j} - x_{i_r} \le 0 \qquad (\forall r \in R)$$

and a family T of CPIs, of the form

$$\sum_{i \in C_t \cup C'_t} x_i - \sum_{\{i,j\} \in \bar{E}(C_t : C'_t)} x_{ij} \le 1 \qquad (\forall t \in T).$$

Let $\lambda_r \ge 0$ and $\lambda_t \ge 0$, for $r \in R$ and $t \in T$, be the multipliers given to these CVIs and CPIs in the linear combination.

For a given vertex i, let R_i be the set of CVIs that involve x_i, i.e. $R_i := \{r \in R : i = i_r\}$, and let T_i be the set of CPIs that involve x_i, i.e. $T_i := \{t \in T : i \in C_k \cup C'_k\}$. We have by assumption that

$$\sum_{t \in T_i} \lambda_t - \sum_{r \in R_i} \lambda_r = \alpha_i \qquad (\forall i \in V). \tag{21}$$

Similarly, for a given 'non-edge' $\{i, j\} \in \bar{E}$, let R_{ij} be the set of CVIs that involve x_{ij}, i.e.

$$R_{ij} := \{r \in R : i = i_r, j \in C_r\} \cup \{r \in R : j = i_r, i \in C_r\},$$

and let T_{ij} be the set of CPIs that involve x_{ij}, i.e.

$$T_{ij} := \{t \in T : \{i, j\} \subset C_k \cup C'_k\}.$$

We have by assumption that

$$\sum_{r \in R_{ij}} \lambda_r - \sum_{t \in T_{ij}} \lambda_t = 0 \qquad (\forall \{i, j\} \in \bar{E}). \tag{22}$$

Now we modify the CVIs and CPIs in the linear combination, introducing where necessary the additional vertex u, so as to obtain the desired lifted inequality. We use the notation $\bar{E}^+ := \bar{E}(\tilde{V} : \tilde{V})$.

For each $r \in R$, we do the following. If $i_r \notin S$ and $|C_r \cap S| = 1$, we insert u into C_r so that the CVI becomes

$$\sum_{j \in C_r \cup \{u\} : \{i_r, j\} \in \bar{E}^+} x_{i_r j} - x_{i_r} \leq 0. \tag{23}$$

If $C_r \cap S = \emptyset$ and $i_r \in S$, we sum together the original CVI and the CVI obtained by replacing i_r with u, so that the CVI becomes:

$$\sum_{j \in C_r : \{i_r, j\} \in \bar{E}} x_{i_r j} + \sum_{j \in C_r : \{u, j\} \in \bar{E}^+} x_{uj} - x_{i_r} - x_u \leq 0. \tag{24}$$

In all other cases, the CVI remains unchanged.

For each $t \in T$, we do the following. If $C_t \cap S = C'_t \cap S = \emptyset$, the CPI remains unchanged. If $|C_t \cap S| = 1$ and $C'_t \cap S = \emptyset$, we insert u into C_t, so that the CPI becomes:

$$\sum_{i \in C_t \cup C'_t \cup \{u\}} x_i - \sum_{\{i,j\} \in \bar{E}^+(C_t \cup \{u\} : C'_t)} x_{ij} \leq 1. \tag{25}$$

If $|C'_t \cap S| = 1$ and $C_t \cap S = \emptyset$, we insert u into C'_t analogously. If $|C_t \cap S| = |C'_t \cap S| = 1$, we insert u into both C_t and C'_t, yielding:

$$\sum_{i \in C_t \cup C'_t \cup \{u\}} x_i - \sum_{\{i,j\} \in \bar{E}(C_t : C'_t)} x_{ij} \leq 1.$$

We now show that the linear combination of these modified CVIs and CPIs (using the same multipliers as before) is the desired lifted inequality. Clearly, the coefficients for variables not involving u are unchanged, and so is the right hand side β. The coefficient of x_{iu}, for any $i \in V \setminus (S \cup n(S))$, is:

$$\sum_{j \in S} \left(\sum_{r \in R_{ij}} \lambda_r - \sum_{t \in T_{ij}} \lambda_t \right),$$

which, by Eq. (22), is zero. Finally, the coefficient of x_u is:

$$\sum_{t \in \bigcup_{i \in S} T_i} \lambda_t - \sum_{r \in R : i_r \in S, C_r \cap S = \emptyset} \lambda_r.$$

This is equivalent to:

$$\sum_{i \in S} \left(\sum_{t \in T_i} \lambda_t - \sum_{r \in R_i} \lambda_r \right) - \sum_{\{i,j\} \subset S} \left(\sum_{r \in R_{ij}} \lambda_r - \sum_{t \in T_{ij}} \lambda_t \right).$$

By Eqs. (22), the second summation vanishes. By Eqs. (21), the first summation equals $\sum_{i \in S} \alpha_i$, as required. □

4 Computational experiments

In this section, we turn our attention from theory to computation. In Subsect. 4.1, we explain how we approximately solve the $M(K, K)$ relaxation. In Subsect. 4.2, we give extensive computational results, and compare the upper bound obtained with some of the other upper bounds that we mentioned in Sect. 2. We will see that the $M(K, K)$ relaxation gives a very strong bound in many cases, despite the fact that it is based on LP rather than SDP. In fact, it is frequently stronger than the Lovász theta bound.

To our knowledge, the only other paper comparing LP and SDP bounds for the stable set problem is Balas et al. [1]. One of their experiments, in which regular lift-and-project cuts were compared with lift-and-project cuts using psd-ness, showed that imposing psd-ness makes little difference if clique inequalities are included in the initial LP relaxation. This is in line with our results.

4.1 The algorithm

Recall that the number of CVIs and CPIs depends on the number $|\Omega|$ of maximal cliques in G. (To be precise, there are $n|\Omega|$ CVIs and $|\Omega|(|\Omega|-1)/2$ CPIs.) Since $|\Omega|$ is typically exponential in n, it is natural to consider using a standard simplex-based cutting plane algorithm, in which violated CVIs and CPIs are iteratively added to the LP relaxation. Unfortunately, some difficulties prevent such an approach. First, both separation problems associated with the CPIs and CVIs are strongly NP-hard [9]. This in itself is not a major drawback, since effective separation heuristics can be devised. However, we experienced that the cutting plane approach performs very badly, exhibiting severe primal and dual degeneracy and 'tailing off'.

A better approach turned out to be the following: construct a large collection of 'promising' CVIs and CPIs, and then feed them into an LP solver. After a great deal of experimentation [9], we found that the following (non-standard) "three-phase" approach allows one to optimize over $M(K, K)$ to good precision and in a reasonable amount of time for many graphs of interest:

1. **Clique selection.** The first step consists of running the cutting plane algorithm used in [28] in the original (non-quadratic) space, including only clique inequalities as cutting planes, and collecting all of the maximal cliques generated during the algorithm. We then build two distinct collections Ω_{CVI} and Ω_{CPI} containing

the cliques whose associated clique inequalities have a small slack (computed at the final fractional point). The thresholds for the slack (`clique_slack_CVI` and `clique_slack_CPI`, respectively) are parameters in our algorithm. Then, an inequality pool is constructed including all the CVIs and CPIs corresponding to the cliques in Ω_{CVI} and Ω_{CPI}, respectively.

2. **Core selection.** In this phase a subset of the CVIs and CPIs stored in the pool is selected so as to reduce the formulation size without degrading the resulting upper bound. We construct a Lagrangian relaxation, obtained by dualizing all the constraints in the pool and keeping only the box constraints in the Lagrangian subproblem. The traditional subgradient algorithm is used to improve the Lagrangian multipliers, but is interrupted when the current value of the Lagrangian dual drops below the optimal value of the clique relaxation of the previous phase. Then, the `core_size` inequalities corresponding to the CVIs and CPIs with the largest Lagrangian multipliers are loaded into the final formulation.
3. **Optimization.** An interior-point algorithm is executed to solve the core LP to optimality. The use of interior-point rather than simplex enables one to avoid problems with degeneracy and slow convergence.

We also tested the following alternative core selection strategy: solve the Lovász theta relaxation, yielding an optimal solution matrix $\bar{Y}$, and then use CVIs and CPIs which are near-tight at $\bar{Y}$ to construct the core. Although this approach worked well for a few graphs, it was outperformed by the Lagrangian approach. Thus, our preferred method does not require any SDP tools.

4.2 Computational results

The algorithm was coded in C++ and the experiments run on a 2.0 GHz Pentium with 2 GB RAM. The LP solver in the clique selection phase was ILOG CPLEX 9.1, while the interior point algorithm in the optimization phase was MOSEK 5.0.0.60.

The test-bed contains all of the graphs from the DIMACS second challenge [16] with $n < 400$, available at the web site [5]. There are 34 such instances. It also includes the uniform random graphs used in Dukanovic and Rendl [7] (downloadable from [6]), and some very sparse random graphs generated with the same parameters as those tested in Gruber and Rendl [13]. The graphs in the first two test sets were complemented, because we are interested in the stability number rather than the clique number. All instances are unweighted instances, which tend to be the most difficult in practice. The upper bound obtained by our algorithm is denoted by UB_{MKK}.

Experiment 1: DIMACS benchmark graphs. Table 1 compares UB_{MKK} with an upper bound obtained by optimising approximately over QSTAB(G) (denoted by UB_{clique}), with $\theta(G)$, and with the bounds reported in [7] (DR) and [3] (BV). An asterisk in the DR or BV columns means that results were not reported in the corresponding paper for that instance. A rectangle is drawn whenever the corresponding entry is the (unique) best bound.

In ten cases out of 34, $\theta(G)$ is the unique best bound. Although the bounds in [3,7] are theoretically stronger than $\theta(G)$, few results are given in [7] and the method

Table 1 DIMACS graphs: comparison among upper bounds

Graph	n	$\|E\|$	$\alpha(G)$	UB_{clique}	$\theta(G)$	UB_{MKK}	DR	BV
brock200_1	200	5,066	21	38.20	[27.5]	30.25	*	27.98
brock200_2	200	10,024	12	21.53	[14.22]	16.09	*	17.08
brock200_3	200	7,852	15	27.73	[18.82]	21.16	*	20.79
brock200_4	200	6,811	17	30.84	[21.29]	23.80	*	22.8
C.125.9	125	787	34	43.06	37.89	[36.53]	*	*
C.250.9	250	3,141	44	71.50	[56.24]	59.96	*	*
c-fat200-1	200	18,336	12	12.53	12	12	*	14.97
c-fat200-2	200	16,665	24	24	24	24	*	24.08
c-fat200-5	200	11,427	58	66.67	60.34	[58]	*	58.17
DSJC125.1	125	736	34	43.15	38.39	[36.99]	*	*
DSJC125.5	125	3,891	10	15.6	11.47	11.41	11.4	*
DSJC125.9	125	6,961	4	4.72	4.00	4	4.06	*
mann_a9	45	72	16	18.50	17.47	[16.85]	*	17.17
mann_a27	378	702	126	135.00	132.76	[131.39]	*	*
gen200_p0.9_44	200	1,990	44	44	44	44	*	*
gen200_p0.9_55	200	1,990	55	55	55	55	*	*
hamming6-2	64	192	32	32	32	32	*	32
hamming6-4	64	1,312	4	5.33	5.33	4	4	4.54
hamming8-2	256	1,024	128	128	128	128	*	128
hamming8-4	256	11,776	16	16	16	16	*	20.54
johnson8-2-4	28	168	4	4.22	4	4	*	4
johnson8-4-4	70	560	14	14	14	14	*	14
johnson16-2-4	120	1,620	8	8.20	8	8	*	10.26
keller4	171	5,100	11	14.82	14.01	[13.17]	*	15.41
p_hat300_1	300	33,917	8	15.68	[10.1]	11.40	*	18.66
p_hat300_2	300	22,922	25	34.01	[27]	30.00	*	30.1
p_hat300_3	300	11,460	36	54.74	[41.16]	47.32	*	43.32
san200_0.7-1	200	5,970	30	30	30	30	*	30.7
san200_0.7-2	200	5,970	18	21.14	18	18	*	20.01
san200_0.9-1	200	1,990	70	70	70	70	*	70.54
san200_0.9-2	200	1,990	60	60	60	60	*	60.72
san200_0.9-3	200	1,990	44	45.13	44	44	*	44.4
sanr200_07	200	6,032	18	33.48	[23.8]	26.12	*	24.97
sanr200_09	200	2,037	42	60.04	[49.3]	50.73	*	49.31

proposed in [3] frequently failed to compute a bound as good as $\theta(G)$, presumably due to time or memory problems. In six cases, UB_{MKK} is the unique best bound; and in those cases it improves on $\theta(G)$ by a large amount. Notice that all such instances apart from c-fat200-5 have at most 35% density. In two other cases (DSJC125.5 and hamming6-4), a slight improvement on $\theta(G)$ is obtained by both UB_{MKK} and DR. In the remaining sixteen cases, $\alpha(G) = \theta(G) = UB_{MKK}$.

The major difficulty for our algorithm comes from the large number of x_{ij} variables, which limits the size of the core LP solvable by MOSEK. Thus, rather small values for core_size are sometimes needed, yielding an impairment of the upper bound. This occurs particularly for brock200_1, C.250.9 and p_hat300-3.

In Table 2, some other computational details are reported: the sizes of the collection Ω_{CVI}, Ω_{CPI}, the total number of CVIs and CPIs, the number of constraints in the core LP, the time for core selection, the time for LP solving and the total time. In the last column, we also include the computing times reported in [3]. They were obtained on a Pentium 4 under Linux with a 2.4 GHz processor and 1GB RAM.

The total time required to compute UB_{MKK} turns out to be remarkably smaller than that reported in [3] (differences in the computers are not significant). Nevertheless, we found that a very large number of subgradient iterations were required for effective core selection in the case of the brock, p_hat and keller4 instances. This is the reason for the large running times in those cases.

A complete comparison with Balas et al. [1] cannot be done since, in that paper, the bounds at the root node of the branch-and-bound tree are not reported. However, in 17 cases out of the 21 tested in [1], UB_{MKK} equals the stability number. We found that in the remaining 4 cases (keller4, C125.9, brock200_2 and p_hat300-1), the upper bound UB_{clique} is not significantly improved by the CPLEX disjunctive cuts (even with the so-called 'aggressive' setting). Thus, we conclude that $M(K, K)$ gives much stronger bounds than lift-and-project cuts. Of course, it should be borne in mind that lift-and-project cuts can be easily embedded into a branch-and-bound scheme, as demonstrated in [1], whereas embedding $M(K, K)$ within branch-and-bound is likely to be more difficult.

Experiment 2: uniform random graphs. In our second experiment, we took the random graphs used in Dukanovic and Rendl [7]. Since we had to complement them, graph $x.y$ represent the complement of the graph with $n = x$ and $100 - y$ density reported in [7]. Table 3 compares UB_{MKK} with UB_{clique}, $\theta(G)$ and the bound reported in [7]. The asterisks correspond to instances with memory requirements larger than 2GB.

For $n = 100$, $M(K, K)$ is computationally manageable and always returns the best bound. As the size increases ($n \geq 150$), the comparison among relaxations is affected by the graph density.

In the denser cases ($x.90$ and $x.75$), DR is always the stronger bound, being often slightly better than $\theta(G)$; UB_{MKK} in these cases is slightly worse than $\theta(G)$.

In the medium density cases ($x.50$), DR slightly improves on $\theta(G)$ for $n = 150$ but cannot be computed for $n \geq 200$ due to memory limits. In all these cases, UB_{MKK} is outperformed by $\theta(G)$, and the difference is more marked for the larger instances.

In the sparsest cases ($x.25$, $x.10$), DR cannot be computed due to memory limits. On the contrary, UB_{MKK} is competitive. It improves on $\theta(G)$ in the case of 150.10 and is still close to $\theta(G)$ in the case of 200.10. For 250.10, the core size had to be reduced, leading to a deterioration in UB_{MKK}.

Table 2 DIMACS graphs: computational details

Graph	$\|\Omega_{CVI}\|$	$\|\Omega_{CPI}\|$	CVIs (#)	CPIs (#)	Core size (#)	Core sel. time	LP sol. time	Total time	BV time
brock200_1	1,035	1,035	202,315	535,095	180,000	13,600	4,070	17,670	28,590
brock200_2	1,018	1,018	195,052	517,653	350,000	24,803	1,698	26,501	67,302
brock200_3	1,076	1,076	208,256	578,350	200,000	20,829	1,557	22,386	51,665
brock200_4	1,104	1,104	214,461	608,856	200,000	20,712	4,650	25,362	43,433
C.125.9	645	645	79,110	207,690	286,800	4	223	227	*
C.250.9	1,531	1,531	378,693	1,171,215	49,000	3,600	5,797	9,397	*
c-fat200-1	300	300	55,164	44,850	34,853	5	6	11	126,103
c-fat200-2	300	300	57,655	44,850	69,043	4	48	52	83,691
c-fat200-5	658	658	129,626	216,153	174,327	6	259	265	44,483
DSJC125.1	623	623	76,448	193,753	270,201	4	270	274	*
DSJC125.5	560	560	66,157	156,520	79,680	235	142	377	*
DSJC125.9	118	118	11,830	6,903	4,661	11	2	13	*
mann_a9	54	54	2,313	1,431	3,744	<1	<1	<1	50
mann_a27	628	628	236,091	196,878	432,969	5	388	393	*
gen200_p0.9_44	1,601	1,601	316,291	1,280,800	80,000	163	442	605	*
gen200_p0.9_55	1,253	1,253	247,464	784,378	80,000	114	640	754	*
hamming6-2	192	192	11,904	18,336	30,240	0	3	3	15
hamming6-4	98	98	5,181	4,753	9,934	<1	4	4	1,416
hamming8-2	1,024	1,024	260,096	523,776	191,232	100	4	104	728
hamming8-4	309	309	74,576	47,586	19,896	602	2,350	2,952	90,169
johnson8-2-4	34	34	782	561	1,343	<1	<1	<1	59
johnson8-4-4	114	114	6,669	6,441	4,042	4	3	7	479
johnson16-2-4	52	52	5,591	1,326	6,917	<1	31	31	3,140
keller4	1,029	1,029	167,728	528,906	650,000	13,676	1,648	15,324	19,319
p_hat300_1	1,673	332	471,567	54,946	200,000	3,500	1,410	4,910	322,287
p_hat300_2	911	911	265,303	414,505	30,000	18,200	6,137	24,337	244,428
p_hat300_3	1,140	1,140	336,003	649,230	30,000	41,643	4,765	46,408	101,995
san200_0.7-1	412	412	76,996	84,666	15,543	10	72	82	31,049
san200_0.7_2	173	173	33,179	14,878	15,543	55	245	300	37,102
san200_0.9-1	1,491	1,491	294,958	110,795	19,221	17	1	18	6,947
san200_0.9-2	925	925	182,893	537,166	19,221	14	2	16	6,977
san200_0.9-3	603	603	114,827	181,503	19,221	13	140	143	12,281
sanr200_07	4,443	1,049	874,433	549,676	200,000	7,400	2,571	9,971	36,576
sanr200_09	1,783	901	352,404	405,450	200,000	3,517	4,966	8,483	9,428

It is well known that improving on $\theta(G)$ for random graphs is a rather difficult task [7]. These results show that UB_{MKK} can give meaningful improvements in the cases in which $M(K, K)$ is computationally manageable (see also Table 4).

Table 3 Random graphs: comparison among upper bounds

Graph	n	|E|	$\alpha(G)$	UB_{clique}	$\theta(G)$	UB_{MKK}	DR07
100.10	100	490	31	37.27	33.16	31.76	32.34
100.25	100	1,216	17	23.33	19.49	19.03	19.26
100.50	100	2,419	9	13.96	10.82	10.58	10.74
100.75	100	3,710	5	7.45	5.82	5.47	5.80
100.90	100	4,463	4	4.21	4	4	4
150.10	150	1,096	37	49.08	41.99	41.56	*
150.25	150	2,724	19	31.57	24.33	25.25	*
150.50	150	5,510	10	18.40	12.90	13.61	12.82
150.75	150	8,373	6	9.80	6.86	6.86	6.84
150.90	150	10,038	5	5.31	5	5	5
200.10	200	1,958	42	61.44	50.14	51.28	*
200.25	200	4,851	22	39.48	28.68	31.21	*
200.50	200	9,874	11	22.33	14.68	16.57	*
200.75	200	14,801	7	12.00	7.81	8.34	7.78
200.90	200	17,853	4	6.86	4.44	4.66	4.44
250.10	250	2,998	46	73.85	58.06	62.18	*
250.25	250	7,584	23	46.18	31.83	37.20	*
250.50	250	15,457	11	25.99	16.19	19.52	*
250.75	250	23,199	7	14.11	8.53	9.89	8.50
250.90	250	27,976	4	7.77	4.80	5.25	4.80

As for running times, we experienced that, as the graph size increases, a larger number of subgradient iterations are necessary to select the core LP, yielding larger total times.

Experiment 3*: uniform random graphs with* [1, 5]% *density.* This experiment deals with very sparse random graphs. Its relevance comes from the fact that these are the only random graphs for which a significant improvement on $\theta(G)$ has been achieved. Specifically, this was achieved by Gruber and Rendl [13], who added triangle inequalities to the SDP relaxation.

The results are reported in Tables 5 and 6. In the last column, the percentage gap closed with respect to $\theta(G)$ is included. This shows that UB_{MKK} is significantly stronger than $\theta(G)$ on these instances. Notice that in three cases, namely, 170.3, 200.2 and 400.1, UB_{MKK} equals the stability number.

Even if a precise comparison with [13] cannot be conducted, since the graphs involved are not exactly the same, the percentage gap closed with respect to $\theta(G)$ by the two approaches looks comparable. As for running times, our LP-based approach seems to be much faster.

Table 4 Random graphs: computational details

Graph	$\lvert\Omega_{CVI}\rvert$	$\lvert\Omega_{CPI}\rvert$	CVIs (#)	CPIs (#)	Core size (#)	Core sel. time	LP sol. time	Total time
100.10	433	433	42,306	93,528	135,834	1	91	92
100.25	1,081	1,081	105,220	583,740	500,000	140	224	364
100.50	1,024	1,024	96,790	523,776	80,000	3,360	48	3,408
100.75	1,577	204	140,302	20,706	30,000	1,200	3,181	4,381
100.90	1,109	442	86,965	97,461	5,000	295	1	296
150.10	983	983	145,149	482,653	400,000	100	1,877	1,977
150.25	2,574	1,157	379,182	668,746	700,000	1,047	871	1,918
150.50	1,227	1,227	175,381	752,151	300,000	3,100	423	3,523
150.75	8,419	846	1,206,755	357,435	500,000	4,856	65	4,921
150.90	966	149	119,868	11,026	5,000	3	2	5
200.10	1,690	1,690	333,932	1,427,205	200,000	1,800	13,977	15,777
200.25	1,449	1,449	283,808	1,049,076	210,000	5,200	5,748	10,948
200.50	1,078	1,078	206,821	580,503	200,000	3,000	1,903	4,903
200.75	6,040	533	1,124,114	141,778	700,000	3,000	274	3,274
200.90	3,671	202	643,122	20,301	650,000	4	1,122	1,126
250.10	1,579	1,579	390,583	1,245,831	60,000	11,400	4825	16,225
250.25	1,264	1,264	309,892	798,216	40,000	15,000	7201	22,201
250.50	948	948	228,524	448,878	100,000	16,800	6211	23,011
250.75	3,312	258	775,886	33,153	809,039	7	616	623
250.90	3,935	285	869,360	40,470	909,830	8	180	188

Table 5 Random graphs [1, 5]%: upper bounds and CPU times

Graph	n	|E|	$\alpha(G)$	UB_{clique}	$\theta(G)$	UB_{MKK}	% gap closed
150.4	150	459	58	67.50	62.40	60.21	49.77
150.5	150	556	55	62.00	58.01	55.33	89.03
170.3	170	451	70	79.50	73.51	70.00	100.00
200.2	200	420	93	97.50	94.77	93.00	100.00
200.3	200	603	80	89.00	83.63	80.38	89.53
300.2	300	905	121	142.00	128.10	123.80	60.56
350.2	350	1,206	132	156.00	141.94	137.77	41.95
400.1	400	816	187	199.00	191.42	187.00	100.00

In summary, the three experiments show that optimizing over $M(K, K)$, which does not theoretically dominate any of the SDP relaxations, yields in several cases upper bounds which are stronger than those obtained so far with methods based on SDP.

Table 6 Random graphs [1, 5]%: computational details

Graph	$\lvert\Omega_{CVI}\rvert$	$\lvert\Omega_{CPI}\rvert$	CVIs (#)	CPIs (#)	Core size (#)	Core sel. time	LP sol. time	Total time
150.4	515	515	76,169	132,355	208,524	1	556	557
150.5	549	549	81,159	150,426	231,585	1	659	660
170.3	342	342	57,437	58,311	22,000	17	6	23
200.2	284	284	56,220	40,186	23,000	13	2	15
200.3	693	693	137,149	239,778	376,927	3	3,952	3,955
300.2	1,084	1,084	322,977	586,986	190,000	1,000	12,906	13,906
350.2	881	881	306,534	387,640	150,000	1,900	6,535	8,435
400.1	646	646	257,102	208,335	170,000	76	25	101

5 Conclusions

We have explored for the first time, from both theoretical and computational points of view, the polytope obtained by applying the Lovász–Schrijver $M(K, K)$ operation to the clique polytope QSTAB(G). The main theoretical conclusion is that this polytope (and its projection into the non-quadratic space) satisfies all web and antiweb inequalities, along with various inequalities obtained by sequential lifting. This extends the projection results given in Laurent et al. [18] and Giandomenico and Letchford [11]. The computational results show that the upper bound on the stability number obtained by optimising over $M(K, K)$ is very strong, sometimes even stronger than the best bounds obtained by SDP-based techniques.

A natural next step would be to attempt to use the CVIs and CPIs within an exact (branch-and-bound or branch-and-cut) scheme for the stable set problem. However, for such an approach to be viable, faster methods for (approximately) optimizing over $M(K, K)$ will be required. Our computational experience suggests that the most promising methods would be of Lagrangian type, such as bundle methods.

Acknowledgments The research of the second author was supported by the Engineering and Physical Sciences Research Council under grant EP/D072662/1.

References

1. Balas, E., Ceria, S., Cornuéjols, G., Pataki G.: Polyhedral methods for the maximum clique problem. In: Johnson, D.S., Trick, M.A. (eds.) Op. cit., pp. 11–28. (1996)
2. Borndörfer, R.: Aspects of Set Packing, Partitioning and Covering. Doctoral Thesis, Technical University of Berlin (1998)
3. Burer, S., Vandenbussche, D.: Solving lift-and-project relaxations of binary integer programs. SIAM J. Opt. **16**, 726–750 (2006)
4. Cheng, E., De Vries, S.: Antiweb-wheel inequalities and their separation problems over the stable set polytopes. Math. Program. **92**, 153–175 (2002)
5. Repository ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique
6. Website http://www.math.uni-klu.ac.at/or/Software/software.html

7. Dukanovic, I., Rendl, F.: Semidefinite programming relaxations for graph coloring and maximal clique problems. Math. Program. **109**, 345–365 (2007)
8. Fulkerson, D.R.: Anti-blocking polyhedra. J. Comb. Th. (B) **12**, 50–71 (1972)
9. Giandomenico, M.: Extended Formulations for the Stable Set Problem: Theory and Experiments. Phd Thesis, University of Rome ‘La Sapienza’ (2006)
10. Gerards, A.M.H., Schrijver, A.J.: Matrices with the Edmonds–Johnson property. Combinatorica **6**, 365–379 (1986)
11. Giandomenico, M., Letchford, A.N.: Exploring the relationship between max-cut and stable set relaxations. Math. Program. **106**, 159–175 (2006)
12. Grötschel, M., Lovász, L., Schrijver, A.J.: Geometric Algorithms and Combinatorial Optimization. Wiley, New York (1988)
13. Gruber, G., Rendl, F.: Computational experience with stable set relaxations. SIAM J. Opt. **13**, 1014–1028 (2003)
14. Håstad, J.: Clique is hard to approximate within $n^{1-\epsilon}$. Acta Math. **182**, 105–142 (1999)
15. Hoffman, K., Padberg, M.W.: Solving airline crew scheduling problems by branch-and-cut. Manage. Sci. **39**, 657–682 (1993)
16. Johnson, D.S., Trick, M.A. (eds.) Cliques, Coloring and Satisfiability: the 2nd DIMACS Implementation Challenge. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26. American Mathematical Society, Providence (1996)
17. Juhász, F.: The asymptotic behaviour of Lovász’ θ function for random graphs. Combinatorica **2**, 153–155 (1982)
18. Laurent, M., Poljak, S., Rendl, F.: Connections between semidefinite relaxations of the max-cut and stable set problems. Math. Program. **77**, 225–246 (1997)
19. Lovász, L.: Normal hypergraphs and the perfect graph conjecture. Discr. Math. **2**, 253–267 (1972)
20. Lovász, L.: On the Shannon capacity of a graph. IEEE Trans. Inf. Theory **25**, 1–7 (1979)
21. Lovász, L., Schrijver, A.J.: Cones of matrices and set-functions and 0–1 optimization. SIAM J. Opt. **1**, 166–190 (1991)
22. Malick, J., Povh, J., Rendl, F., Wiegele, A.: Regularization methods for semidefinite programming. Optimization online, http://www.optimization-online.org/DB_FILE/2007/10/1800.pdf (2007)
23. Nemhauser, G.L., Sigismondi, G.: A strong cutting plane/branch-and-bound algorithm for node packing. J. Opl. Res. Soc. **43**, 443–457 (1992)
24. Padberg, M.W.: On the facial structure of set packing polyhedra. Math. Program. **5**, 199–215 (1973)
25. Padberg, M.W.: The boolean quadric polytope: some characteristics, facets and relatives. Math. Program. **45**, 139–172 (1989)
26. Povh, J., Rendl, F., Wiegele, A.: A boundary point method to solve semidefinite programs. Computing **78**, 277–286 (2006)
27. Régin, J.-C.: Using constraint programming to solve the maximum clique problem. In: Rossi, F. (ed) Proceedings of CP 2003. Lecture Notes in Computer Science, vol. 2833. Springer, Berlin (2003)
28. Rossi, F., Smriglio, S.: A branch-and-cut algorithm for the maximum cardinality stable set problem. Oper. Res. Lett. **28**, 63–74 (2001)
29. Schrijver, A.J.: A comparison of the Delsarte and Lovász bounds. IEEE Trans. Inf. Theory **IT-25**, 425–429 (1979)
30. Tomita, E., Kameda, T.: An efficient branch-and-bound algorithm for finding a maximum clique, with computational experiments. J. Glob. Opt. **37**, 95–111 (2007)
31. Trotter, L.E.: A class of facet-producing graphs for vertex packing polyhedra. Discr. Math. **12**, 373–388 (1975)
32. Yildirim, E.A., Fan, X.: On extracting maximum stable sets in perfect graphs using Lovász’s theta function. Comput. Opt. Appl. **33**, 229–247 (2006)