A New Approach to the Stable Set Problem Based on Ellipsoids

Monia Giandomenico[1], Adam N. Letchford[2], Fabrizio Rossi[1], and Stefano Smriglio[1]

[1] Department of Computer Science, University of L'Aquila, Italy
{giandomenico,rossi,smriglio}@di.univaq.it
[2] Department of Management Science, Lancaster University, United Kingdom
A.N.Letchford@lancaster.ac.uk

Abstract. A new exact approach to the stable set problem is presented, which attempts to avoids the pitfalls of existing approaches based on linear and semidefinite programming. The method begins by constructing an ellipsoid that contains the stable set polytope and has the property that the upper bound obtained by optimising over it is equal to the Lovász theta number. This ellipsoid is then used to derive cutting planes, which can be used within a linear programming-based branch-and-cut algorithm. Preliminary computational results indicate that the cutting planes are strong and easy to generate.

Keywords: stable set problem, semidefinite programming, convex quadratic programming, cutting planes.

1 Introduction

Given an undirected graph $G = (V, E)$, a *stable set* is a set of pairwise non-adjacent vertices. The *stable set problem* (SSP) calls for a stable set of maximum cardinality. The SSP is $\mathcal{NP}$-hard [15], hard to approximate [14], and hard to solve in practice (e.g., [7, 23–25]). Moreover, it is a remarkable fact that sophisticated mathematical programming algorithms for the SSP, such as those in [4, 6, 12, 23, 25], have not performed significantly better than relatively simple algorithms based on implicit enumeration, such as those in [7, 24].

A possible explanation for the failure of mathematical programming approaches is the following. Linear Programming (LP) relaxations can be solved reasonably quickly, but tend to yield weak upper bounds. Semidefinite Programming (SDP) relaxations, on the other hand, typically yield much stronger bounds, but take longer to solve. Therefore, branch-and-bound algorithms based on either LP or SDP relaxations are slow, due to the large number of nodes in the search tree, or the long time taken to process each node, respectively.

In this paper we present a way out of this impasse. The key concept is that one can efficiently construct an *ellipsoid* that contains the stable set polytope, in such a way that the upper bound obtained by optimising over the ellipsoid

O. Günlük and G.J. Woeginger (Eds.): IPCO 2011, LNCS 6655, pp. 223–234, 2011.

is equal to the standard SDP bound, the so-called *Lovász theta number*. This ellipsoid can then be used to construct useful convex programming relaxations of the stable set problem or, more interestingly, to derive cutting planes. These cutting planes turn out to be strong and easy to generate.

We remark that our approach can be applied to the variant of the SSP in which vertices are weighted, and one seeks a stable set of maximum weight.

The paper is structured as follows. Some relevant literature is reviewed in Sect. 2, the new approach is presented in Sect. 3, some computational results are given in Sect. 4, and some concluding remarks are made in Sect. 5.

2 Literature Review

We now review the relevant literature. From this point on, $n = |V|$ and $m = |E|$.

2.1 Linear Programming Relaxations

The SSP has the following natural formulation as a 0-1 LP:

$$\begin{aligned} \max \quad & \textstyle\sum_{i \in V} x_i \\ \text{s.t.} \quad & x_i + x_j \le 1 \ (\{i,j\} \in E) && (1) \\ & x \in \{0,1\}^n, && (2) \end{aligned}$$

where the variable x_i takes the value 1 if and only if vertex i is in the stable set.

The convex hull in $\mathbb{R}^n$ of feasible solutions to (1)–(2) is called the *stable set polytope* and denoted by STAB(G). This polytope has been studied in great depth [4, 11, 21]. The most well-known facet-defining inequalities for STAB(G) are the *clique* inequalities of Padberg [21]. A *clique* in G is a set of pairwise adjacent vertices, and the associated inequalities take the form:

$$\sum_{i \in C} x_i \le 1 \qquad (\forall C \in \mathcal{C}), \tag{3}$$

where $\mathcal{C}$ denotes the set of maximal cliques in G. Note that the clique inequalities dominate the edge inequalities (1).

The separation problem for clique inequalities is $\mathcal{NP}$-hard [20]. Fortunately, some fast and effective separation heuristics exist, not only for clique inequalities, but also for various other inequalities (e.g., [4, 23, 25]). Nevertheless, LP-based approaches can run into difficulties when n exceeds 200, mainly due to the weakness of the upper bounds.

2.2 Semidefinite Programming Relaxations

Lovász [17] introduced an upper bound for the SSP, called the *theta number* and denoted by $\theta(G)$, which is based on an SDP relaxation. The bound can be derived in several different ways, and we follow the derivation presented in [11]. We start by formulating the SSP as the following non-convex quadratically-constrained program:

$$\max \quad \sum_{i \in V} x_i \tag{4}$$
$$\text{s.t. } x_i^2 - x_i = 0 \ (i \in V) \tag{5}$$
$$x_i x_j = 0 \quad (\{i,j\} \in E). \tag{6}$$

In order to linearise the constraints, we introduce an auxiliary matrix variable $X = xx^T$, along with the augmented matrix

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

We then note that Y is real, symmetric and positive semidefinite (psd), which we write as $Y \succeq 0$. This leads to the following SDP relaxation:

$$\max \quad \sum_{i \in V} x_i \tag{7}$$
$$\text{s.t. } x = \text{diag}(X) \tag{8}$$
$$X_{ij} = 0 \quad (\{i,j\} \in E) \tag{9}$$
$$Y \succeq 0. \tag{10}$$

Solving this SDP yields $\theta(G)$.

In practice, $\theta(G)$ is often a reasonably strong upper bound for the SSP (e.g., [3, 6, 12, 23]). Unfortunately, solving large-scale SDPs can be rather time-consuming, which makes SDP relaxations somewhat unattractive for use within a branch-and-bound framework.

The above SDP can be strengthened by adding various valid inequalities (e.g., [4, 6, 10, 12, 18, 26]). We omit details, for the sake of brevity.

2.3 The Lovász Theta Body and Ellipsoids

The following beautiful result can be found in Grötschel, Lovász & Schrijver [11]. Let us define the following polytope:

$$\text{QSTAB}(G) = \left\{ x \in \mathbb{R}^n_+ : (3) \text{ hold} \right\},$$

along with the following convex set:

$$\text{TH}(G) = \left\{ x \in \mathbb{R}^n : \exists X \in \mathbb{R}^{n \times n} : (8) - (10) \text{ hold} \right\}.$$

Then we have:

$$\text{STAB}(G) \subseteq \text{TH}(G) \subseteq \text{QSTAB}(G).$$

This implies that $\theta(G)$ dominates the upper bound for the SSP based on clique inequalities.

The set $\text{TH}(G)$ is called the *theta body*. In [11], a characterisation of $\text{TH}(G)$ is given in terms of linear inequalities. We are interested here, however, in a characterisation of TH(G) in terms of convex quadratic inequalities, due to Fujie & Tamura [9]. For a given vector $\mu \in \mathbb{R}^m$, let $M(\mu)$ denote the symmetric matrix with $\mu_{ij}/2$ in the ith row and jth column whenever $\{i,j\} \in E$, and zeroes elsewhere. Then, given vectors $\lambda \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^m$ such that $\mathrm{Diag}(\lambda) + M(\mu)$ is psd, the set

$$E(\lambda,\mu) = \left\{x \in \mathbb{R}^n : \, x^T(\mathrm{Diag}(\lambda) + M(\mu))x \le \lambda^T x\right\}$$

is easily shown to be an ellipsoid that contains STAB(G). The result we need is the following:

Theorem 1 (Fujie & Tamura, 2002). *For any graph G, we have:*

$$TH(G) = \bigcap_{\lambda,\mu:\mathrm{Diag}(\lambda)+M(\mu)\succeq 0} E(\lambda,\mu).$$

2.4 Relaxations of Non-convex Quadratic Problems

For what follows, we will also need a known result concerning relaxations of non-convex quadratic problems, mentioned for example in [8, 16, 22]. Given a problem of the form:

$$\begin{aligned} \inf \quad & x^T Q^0 x + c^0 \cdot x \\ \text{s.t.} \; & x^T Q^j x + c^j \cdot x = b_j \; (j = 1,\dots,r) \\ & x \in \mathbb{R}^n, \end{aligned} \tag{11}$$

we can construct the following SDP relaxation:

$$\begin{aligned} \inf \quad & Q^0 \bullet X + c^0 \cdot x \\ \text{s.t.} \; & Q^j \bullet X + c^j \cdot x = b_j \; (j = 1,\dots,r) \\ & Y \succeq 0, \end{aligned} \tag{12}$$

where $Q^j \bullet X$ denotes $\sum_{i=1}^n \sum_{k=1}^n Q^j_{ik} X_{ik}$. Alternatively, we can form a Lagrangian relaxation by relaxing the constraints (11), using a vector $\phi \in \mathbb{R}^r$ of Lagrangian multipliers. The relaxed problem is then to minimise

$$f(x,\lambda) \; = \; x^T \left(Q^0 + \sum_{j=1}^r \phi_j Q^j\right) x + \left(c^0 + \sum_{j=1}^r \phi_j c^j\right) \cdot x - \sum_{j=1}^r \phi_j b_j$$

subject to $x \in \mathbb{R}^n$.

The result that we need is as follows:

Theorem 2 (Various authors). *Suppose the SDP satisfies the Slater condition, so that an optimal dual solution to the SDP exists. Then optimal Lagrangian multipliers ϕ^* exist too, and are nothing but the optimal dual vectors for the constraints (12) in the SDP. Moreover, the function $f(x,\phi^*)$ is convex, and therefore the matrix $Q^0 + \sum_{j=1}^r \phi_j Q^j$ is psd.*

An analogous result holds when quadratic inequalities, rather than equations, are present: one simply makes the associated multipliers non-negative.

3 The New Approach

In this section, the new approach is described.

3.1 An 'Optimal' Ellipsoid

Recall that Theorem 1 expresses TH(G) as the intersection of an infinite family of ellipsoids. The following proposition states that one can efficiently compute a particular ellipsoid with a very desirable property.

Theorem 3. *Let λ^* and μ^* be optimal dual vectors for the constraints (8) and (9) in the SDP (7)-(10). Then:*

$$\theta(G) = \max\left\{\sum_{i\in V} x_i : x \in E(-\lambda^*, -\mu^*)\right\}.$$

Proof. The SDP (7)-(10) has the form specified in Theorem 3.1 of Tunçel [27], and therefore satisfies the Slater condition. As a result, the optimal dual solution (λ^*, μ^*) exists and its cost is equal to $\theta(G)$. Applying Theorem 2, but switching signs to take into account the fact that the SDP is of maximisation type, we have:

$$\theta(G) = \max\left\{\sum_{i\in V} x_i - x^T(\mathrm{Diag}(\lambda^*) + M(\mu^*))x + \lambda^* \cdot x : \ x \in \mathbb{R}^n\right\}.$$

Moreover, the matrix $-\mathrm{Diag}(\lambda^*) - M(\mu^*)$ must be psd.

Now, given that this is a concave maximisation problem, and the fact that the pair (λ^*, μ^*) form an optimal set of Lagrangian multipliers, we have:

$$\theta(G) = \max\left\{\sum_{i\in V} x_i : \ -x^T(\mathrm{Diag}(\lambda^*) + M(\mu^*))x \leq -\lambda^* \cdot x, \ x \in \mathbb{R}^n\right\},$$

which proves the result. □

In other words, the dual solution to the SDP can be used to construct a relaxation of the SSP that has a linear objective function and a single convex quadratic constraint, whose corresponding upper bound is equal to $\theta(G)$.

Example: Let G be the 5-hole, i.e., a chordless cycle on 5 nodes. The optimal dual solution has $\lambda_i^* = -1$ for all i and $\mu_e^* = (1 - \sqrt{5})/2$ for all e. The corresponding ellipsoid is:

$$\left\{x \in \mathbb{R}^n : \sum_{i\in V} x_i^2 + (\sqrt{5} - 1) \sum_{\{i,j\}\in E} x_i x_j \leq \sum_{i\in V} x_i\right\}.$$

Although it is hard to visualise this ellipsoid, one can show that all points in it satisfy the linear inequality:

$$\sum_{i\in V} x_i \leq \sqrt{5}.$$

This agrees with the known fact that, for the 5-hole, $\theta(G) = \sqrt{5}$ [11]. □

3.2 Cutting Planes from the Optimal Ellipsoid

One way in which to exploit the existence of the optimal ellipsoid would be to construct a branch-and-bound algorithm in which convex quadratically-constrained programs are solved at each node of the enumeration tree. For example, one could solve at each node a relaxation of the form:

$$\begin{aligned} \max \quad & \textstyle\sum_{i \in V} x_i \\ \text{s.t. } & x \in E(-\lambda^*, -\mu^*) \\ & \textstyle\sum_{i \in C} x_i \leq 1 \quad (C \in \mathcal{C}') \\ & x \in [0,1]^n, \end{aligned}$$

where $\mathcal{C}'$ is a suitably chosen collection of clique inequalities.

Here, however, we are concerned with the use of the optimal ellipsoid to generate cutting planes (violated valid linear inequalities), to be used within an LP-based cut-and-branch or branch-and-cut algorithm. In this subsection, we consider the (relatively) easy case in which the cutting planes are simply tangent hyperplanes to the optimal ellipsoid. A way to strengthen these cutting planes will be presented in the following subsection.

As before, let the optimal dual solution be (λ^*, μ^*). Our experience is that the matrix $\text{Diag}(\lambda^*) + M(\mu^*)$ is invariably negative definite in practice, which indicates that the ellipsoid $E(-\lambda^*, -\mu^*)$ is bounded. Under this condition, the matrix is invertible and the optimal ellipsoid is easily shown to have the unique central point $\hat{x} = \frac{1}{2}(\text{Diag}(\lambda^*) + M(\mu^*))^{-1}\lambda^*$.

Observe that the dual solution (λ^*, μ^*) and the centre $\hat{x}$ can be computed once and for all, as a kind of 'pre-processing' step, and stored in memory. Then, to generate cutting planes, one can use the following separation algorithm:

1. Let $x^* \in [0,1]^n$ be the current fractional point to be separated, and suppose that $x^* \notin E(-\lambda^*, -\mu^*)$.
2. Perform a line-search to find a point $\tilde{x}$, that is a convex combination of x^* and $\hat{x}$ and lies on the boundary of $E(-\lambda^*, -\mu^*)$.
3. Using a first-order Taylor approximation, find a linear inequality $a^T x \leq b$ that is both violated by x^*, and defines a tangent hyperplane to $E(-\lambda^*, -\mu^*)$ at $\tilde{x}$.
4. Scale the vector a and scalar b by a constant factor Δ and round down to the nearest integers.

More details will be given in the full version of this paper. We remark that the last step helps to avoid numerical difficulties due to rounding errors, and also makes it easier to strengthen the inequality, as described in the next subsection.

3.3 Cut Strengthening

Let $\alpha^T x \leq \beta$, with $\alpha \in \mathbb{Z}^n$ and $\beta \in \mathbb{Z}_+$, be a cutting plane generated by the procedure described in the previous subsection. To strengthen it, we examine one variable at a time, and solve a pair of small optimisation problems. Specifically, given a variable x_j, we compute for $k \in \{0, 1\}$ the value:

$$\gamma_j^k = \max \left\{ \sum_{i \neq j} \alpha_i x_i \, : \, x \in E(-\lambda^*, -\mu^*), \; x_j = k \right\}, \tag{13}$$

and then the value

$$\delta_j^k = \min \left\{ \beta, \lfloor \gamma_j^k \rfloor \right\}.$$

By construction, δ_j^k is an upper bound on the value taken by $\sum_{i \neq j} \alpha_i x_i$ in any feasible solution, subject to the constraint $x_j = k$. As a result, we can replace the original right-hand side β with δ_j^0 and change the coefficient of x_j from α_j to $\delta_j^0 - \delta_j^1$.

This strengthening procedure, which can be regarded as a special case of a procedure described in [2], can be applied to any set of variables, in any desired sequence. Different choices for the set and the sequence may lead to different cutting planes, starting from the same tangent hyperplane $\alpha^T x \leq \beta$. Again, details will be given in the full version of the paper.

The overall procedure described in this subsection and the last will be referred to as the *Ellipsoid Cut Generation Algorithm* (ECGA).

4 Computational Experiments

In order to gain some insight into the potential effectiveness of the 'ellipsoidal' approach, we have designed and coded a rudimentary LP-based 'cut-and-branch' framework, in which cutting planes are used to strengthen an initial 0-1 LP formulation, and the strengthened formulation is then fed into a standard branch-and-bound solver (see, e.g., [20]).

4.1 The Cut-and-Branch Algorithm

The initial 0-1 LP formulation contains a family of clique inequalities associated with a *clique-cover* of G, by which we mean a set of cliques such that each edge $\{i, j\} \in E$ is contained in at least one clique in the family (see, e.g., [3]). This clique-cover is computed by a simple greedy algorithm. The LP relaxation is then solved, and a cutting-plane algorithm based clique inequalities is then run. The separation problem for clique inequalities is solved by a greedy heuristic. We denote by $\text{UB}_{\text{clique}}$ the upper bound obtained when the clique separator fails.

At that point, a cutting-plane algorithm based on ECGA is executed. In the current implementation of ECGA, the parameter Δ is set to 10^4, and several strengthened cutting planes are generated from each original cutting plane. This is done by strengthening in turn the first k variables, then the second k, and so on (k is always chosen in the range $[3, 10]\%$ of n, except in the case of the `p_hat` graphs, as explained below). This choice is motivated by the empirical observation that it is usually the first few coefficient strengthenings that are the most important. We denote by $\text{UB}_{\text{ellips}}$ the upper bound obtained at the root node at completion of the ECGA-based cutting-plane algorithm.

The strengthening subproblems (13) are solved by the CPLEX `baropt` algorithm. (As pointed out by a referee, they could perhaps be solved directly with linear algebra, which would probably be much faster.) The strengthening step turned out to be of particular relevance, as the 'raw' ellipsoidal cuts described in Subsection 3.2 often turned out to be dense, have 'nasty' coefficients, and be nearly parallel to the objective function. Therefore, cut strengthening leads to better numerical stability.

The computations were run on a workstation equipped with 2 Intel Xeon 5150 processors clocked at 2.66 GHz and with 4GB of RAM, under the Linux 64bit operating system. However, all experiments were performed in single thread mode. The algorithm was implemented within the IBM CPLEX 11.2 framework, in which cuts are added by the `usercuts` option. All CPLEX parameters were left at their default settings, apart from the `branching direction`, which was set to `up`, and the `mip emphasis`, which was set to `optimality`.

4.2 Preliminary Computational Results

In this subsection, we give empirical evidence of the effectiveness of the cuts generated from the ellipsoid. The experiments are based on the instances from the DIMACS Second Challenge (Johnson & Trick [13]) with $n < 400$, available at the web site [5]. Among all such graphs, we consider only those instances for which $\lfloor \text{UB}_{\text{clique}} \rfloor > \alpha(G)$. The corresponding optimal ellipsoids $E(-\lambda^*, -\mu^*)$ have been computed by the SDP solver [19] (coded in Matlab) available at [1].

In Table 1 we report for each graph the name, the number of nodes n and edges m, the cardinality of the maximum stable set $\alpha(G)$, the Lovász theta number $\theta(G)$, and the upper bounds $\text{UB}_{\text{clique}}$ and $\text{UB}_{\text{ellips}}$ mentioned in the previous subsection.

In Table 2, a comparison between two cut-and-branch algorithms, one based only on clique inequalities and the other embedding also cuts from the ellipsoid, is presented. It is important to remark that the first cut-and-branch algorithm is in itself rather competitive. Indeed, it often outperforms the dedicated branch-and-cut algorithms described in [23] and and [25], even though they both benefit from having several separation routines, dedicated cut pool management, and specialised branching strategies.

For each algorithm, the number of evaluated subproblems, as well as the total CPU time (in seconds), are reported. In the case of the ellipsoid cut-and-branch two more columns show how the total CPU time is split between branch-and-bound phase and cut lifting, while the total separation time from $E^*(-\lambda^*, -\mu^*)$ (Step 1-3 of ECGA) is always negligible (less than 0.1 secs for largest instances) and is not explicitly reported. The last column in the table contains the number of cuts generated by ECGA.

Table 1 shows that our approach always returns upper bounds very close to $\theta(G)$. To our knowledge, this has not been achieved before using cutting planes

Table 1. Root upper bounds

Graph name	n	m	$\alpha(G)$	$\theta(G)$	$\text{UB}_{\text{clique}}$	$\text{UB}_{\text{ellips}}$
brock200_1	200	5,066	21	27.50	38.20	27.79
brock200_2	200	10,024	12	14.22	21.53	14.32
brock200_3	200	7,852	15	18.82	27.73	19.00
brock200_4	200	6,811	17	21.29	30.84	21.52
C.125.9	125	787	34	37.89	43.06	38.05
C.250.9	250	3,141	44	56.24	72.04	57.41
c-fat200-5	200	11,427	58	60.34	66.66	60.36
DSJC125.1	125	736	34	38.39	43.15	38.44
DSJC125.5	125	3,891	10	11.47	15.60	11.48
mann_a27	378	702	126	132.76	135.00	132.88
keller4	171	5,100	11	14.01	14.82	14.09
p_hat300-1	300	33,917	8	10.10	17.71	10.15
p_hat300-2	300	22,922	25	27.00	34.01	27.14
p_hat300-3	300	11,460	36	41.16	54.36	41.66
san200_0.7-2	200	5,970	18	18.00	20.14	18.10
sanr200_07	200	6,032	18	23.80	33.48	24.00
sanr200_09	200	2,037	42	49.30	60.04	49.77

Table 2. Cut-and-branch results

	Clique cut-and-branch		Ellipsoid cut-and-branch				
Graph name	#sub.	B&b time	#sub.	Lifting time	B&b time	Total time	#cuts
brock200_1	270,169	1,784.78	122,387	75.48	1,432.76	1,508.24	40
brock200_2	6,102	83.16	2,689	109.27	209.39	318.66	30
brock200_3	52,173	459.02	7,282	117.38	252.17	369.55	30
brock200_4	85,134	735.52	18,798	180.83	468.52	649.35	40
C.125.9	3,049	5.63	2,514	17.37	6.52	23.89	75
C.250.9	—	—	—	634.23	—	—	250
c-fat200-5	47	12.06	47	51.35	25.16	76.51	10
DSJC125.1	4,743	6.13	2,981	17.92	8.09	26.01	75
DSJC125.5	1,138	6.97	369	18.29	9.66	27.95	50
mann a27	4,552	2.06	1,278	15.60	2.46	18.06	10
keller4	4,856	21.88	3,274	45.21	36.55	81.76	34
p hat300-1	4,518	124.36	4,238	14.56	132.54	147.10	6
p hat300-2	7,150	194.27	1,522	66.59	204.88	271.47	10
p hat300-3	398,516	10,270.53	107,240	99.43	8,756.76	8,856.19	15
san200 0.7-2	86	8.58	0	66.23	0.76	66.99	10
sanr200_07	88,931	827.52	42,080	116.34	733.47	849.81	60
sanr200_09	635,496	2,650.55	274,261	158.64	1,281.52	1,440.16	100

— time limit reached

that involve only the x variables. (It may be asked why the bound $\text{UB}_{\text{ellips}}$ is not *superior* to $\theta(G)$, in light of Theorem 3. This is due to numerical issues, tailing off, and our chosen time limit.) The great potential of the cuts is confirmed by the cut-and-branch results of Table 2. Note that, for all instances, the number of evaluated subproblems decreases significantly (with the exception of c-fat-200-5) when cuts from the ellipsoid are generated. Moreover, this is accomplished by a fairly small number of cuts. Thus, although the cuts tend to be dense, the simplex algorithm does not appear to be particularly slowed down.

Notice also that in 5 instances out of 17, namely the largest and most difficult ones, ECGA is cost-effective, as the time spent in cut generation is more than compensated by the reduction in the size of the enumeration tree. It is worthwhile mentioning that, for the p_hat graphs, only one coefficient needed to be strengthened in order to obtain good cuts.

5 Concluding Remarks

The key idea in this paper is that one can use the dual solution to the SDP relaxation to construct an ellipsoid that wraps reasonably tightly around the stable set polytope, and that this ellipsoid can be used to construct quite strong cutting planes in the original (linear) space. The computational results, though preliminary, indicate that this approach is promising.

There is, however, room for further improvement. In particular, there are several ways in which the ECGA could be modified or extended. For example:

- Instead of using the 'optimal' ellipsoid to generate cutting planes, one could use some other ellipsoid, or indeed a whole family of ellipsoids. This raises the question of how to generate one or more ellipsoids in such a way that one obtains deep cutting planes, but without excessive computing times.
- In the cut-strengthening procedure, one might be able to obtain smaller coefficients γ_j^k by including some valid linear inequalities (such as clique inequalities) in the maximisation (13). This would lead to stronger cutting planes, but the time taken to compute the coefficients would increase.
- One could also search for effective heuristic rules for ordering the variables in the cut-strengthening step.
- As mentioned in Subsection 2.2, the SDP relaxation (7)-(10) can be strengthened by adding valid inequalities. (For example, Schrijver [26] suggested adding the inequalities $X_{ij} \geq 0$ for all $\{i,j\} \notin E$.) Let $\tilde{\theta}(G) < \theta(G)$ be an improved upper bound obtained by solving such a strengthened SDP. The proof of Theorem 3 can be modified to show the existence of an ellipsoid, say $\tilde{E}$, such that
$$\tilde{\theta}(G) = \max \left\{ \sum_{i \in V} x_i : x \in \tilde{E} \right\}.$$
Such an ellipsoid might produce stronger cutting planes.

Progress on these issues, if any, will be reported in the full version of the paper.

Finally, one could explore the possibility of adapting the ellipsoidal approach to other combinatorial optimisation problems. In our view, this is likely to work well only for problems that have a 'natural' formulation as a continuous optimisation problem with a linear objective function and non-convex quadratic constraints, like the formulation (4)–(6) of the SSP.

Acknowledgments. The second author was supported by the Engineering and Physical Sciences Research Council under grant EP/D072662/1. Thanks are due to an anonymous referee whose comments helped to improve the paper.

References

1. Alpen-Adria-Universität Klagenfurt website, http://www.math.uni-klu.ac.at/or/Software
2. Andersen, K., Pochet, Y.: Coefficient strengthening: a tool for reformulating mixed-integer programs. Math. Program. 122, 121–154 (2009)
3. Balas, E., Ceria, S., Cornuéjols, G., Pataki, G.: Polyhedral methods for the maximum clique problem. In: Johnson, D.S., Trick, M.A. (eds.) Cliques, Coloring and Satisfiability, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26, pp. 11–28 (1996)
4. R.: Borndörfer, Aspects of Set Packing, Partitioning and Covering. Doctoral Thesis, Technical University of Berlin (1998)
5. DIMACS Repository, ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique
6. Dukanovic, I., Rendl, F.: Semidefinite programming relaxations for graph coloring and maximal clique problems. Math. Program 109, 345–365 (2007)
7. Fahle, T.: Simple and Fast: Improving a Branch-And-Bound Algorithm for Maximum Clique. In: Möhring, R.H., Raman, R. (eds.) ESA 2002. LNCS, vol. 2461, pp. 47–86. Springer, Heidelberg (2002)
8. Fujie, T., Kojima, M.: Semidefinite programming relaxation for nonconvex quadratic programs. J. Glob. Opt. 10, 367–380 (1997)
9. Fujie, T., Tamura, A.: On Grötschel-Lovász-Schrijver's relaxation of stable set polytopes. J. Oper. Res. Soc. Japan 45, 285–292 (2002)
10. Giandomenico, M., Letchford, A.N.: Exploring the relationship between max-cut and stable set relaxations. Math. Program 106, 159–175 (2006)
11. Grötschel, M., Lovász, L., Schrijver, A.J.: Geometric Algorithms in Combinatorial Optimization. Wiley, New York (1988)
12. Gruber, G., Rendl, F.: Computational experience with stable set relaxations. SIAM J. Opt. 13, 1014–1028 (2003)
13. Johnson, D.S., Trick, M.A. (eds.): Cliques, Coloring and Satisfiability: the 2nd DIMACS Implementation Challenge. American Mathematical Society, Providence
14. Håstad, J.: Clique is hard to approximate within $n^{1-\epsilon}$. Acta Math. 182, 105–142 (1999)
15. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum, New York (1972)
16. Lemaréchal, C., Oustry, F.: SDP relaxations in combinatorial optimization from a Lagrangian viewpoint. In: Hadjisawas, N., Pardalos, P.M. (eds.) Advances in Convex Analysis and Global Optimization. Kluwer, Dortrecht (2001)
17. Lovász, L.: On the Shannon capacity of a graph. IEEE Trans. Inform. Th. IT-25, 1–7 (1979)

18. Lovász, L., Schrijver, A.J.: Cones of matrices and set-functions and 0-1 optimization. SIAM J. Optimization 1, 166–190 (1991)
19. Malick, J., Povh, J., Rendl, F., Wiegele, A. (2007) Boundary Point Method for solving SDPs: mprw.m, Inst. f. Mathematik, Alpen-Adria-Universität Klagenfurt (2007)
20. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley, New York (1988)
21. Padberg, M.W.: On the facial structure of set packing polyhedra. Math. Program. 5, 199–215 (1973)
22. Poljak, S., Rendl, F., Wolkowicz, H.: A recipe for semidefinite relaxation for (0,1)-quadratic programming. J. Global Opt. 7, 51–73 (1995)
23. Rebennack, S., Oswald, M., Theis, D.O., Seitz, H., Reinelt, G., Pardalos, P.M.: A branch and cut solver for the maximum stable set problem. J. Comb. Opt. (2010) (to appear)
24. Régin, J.-C.: Solving the maximum clique problem with constraint programming. In: Proceedings of CPAIOR 2003. LNCS, vol. 2883, pp. 634–648. Springer, Heidelberg (2003)
25. Rossi, F., Smriglio, S.: A branch-and-cut algorithm for the maximum cardinality stable set problem. Oper. Res. Lett. 28, 63–74 (2001)
26. Schrijver, A.J.: A comparison of the Delsarte and Lovász bounds. IEEE Trans. Inf. Th. IT-25, 425–429 (1979)
27. Tunçel, L.: On the Slater condition for SDP relaxations of nonconvex sets. Oper. Res. Lett. 29, 181–186 (2001)