Combining Semidefinite and Polyhedral Relaxations for Integer Programs

C. Helmberg[1], S. Poljak[2], F. Rendl[1], and H. Wolkowicz[3]

[1] Technische Universität Graz, Institut für Mathematik, Kopernikusgasse 24, A-8010 Graz, Austria
[2] Universität Passau, Institut für Mathematik und Informatik, Innstraße 33, 94030 Passau, Germany
[3] University of Waterloo, Department of Combinatorics and Optimization, Waterloo, Canada

Abstract. We present a general framework for designing semidefinite relaxations for constrained 0-1 quadratic programming and show how valid inequalities of the cut–polytope can be used to strengthen these relaxations. As examples we improve the ϑ–function and give a semidefinite relaxation for the quadratic knapsack problem. The practical value of this approach is supported by numerical experiments which make use of the recent development of efficient interior point codes for semidefinite programming.

Key words: integer linear programming, semidefinite programming, quadratic 0-1 optimization, interior point methods.

1 Introduction

Semidefinite Programming offers excellent possibilities for the design of very tight relaxations for combinatorial problems. With the development of efficient interior point methods [10] it is now possible to solve these relaxations in reasonable time. Indeed, for some problems the semidefinite approach is already more efficient than sophisticated linear programming codes [9]. We present a general framework for designing semidefinite relaxations for constrained 0-1 quadratic programming based on ideas of Lovász et al [15] and Balas et al [2]. We show that these relaxations can be considerably strengthened by exploiting the equivalence of the 0-1 semidefinite relaxation for 0-1 quadratic programming and the (-1,1) semidefinite relaxation for max–cut. We believe that this is the first step towards the development of a new powerful tool for solving difficult subproblems in integer programming.

Probably the best known semidefinite relaxation is the Lovász ϑ–function for the maximum clique problem [14]. The potential of this relaxation was recognized immediately, but since no efficient semidefinite programming codes were available at that time it was more or less impossible to solve it computationally. In the sequel semidefinite formulations were — as far as practical applications are concerned — mainly used to construct new valid inequalities for linear programs. Combinatorial applications were also an important inducement for the development of interior point algorithms for semidefinite programming [1, 10]. Recently several results confirm the usefulness of semidefinite programming in combinatorial optimization. Based on the semidefinite formulation of the max–cut upper bound $\varphi(G)$ introduced in [6] Goemans and Williamson developed an 0.878 approximation algorithm for max–cut, thereby proving a worst case error of 14% for $\varphi(G)$. It was noted by Laurent and Poljak [12] that there is a close relationship between the max–cut polytope and the feasible set of the semidefinite relaxation $\varphi(G)$. Computationally this connection is exploited in [9] where $\varphi(G)$ is considerably improved by an interior point cutting plane approach using the clique facets of the max–cut polytope.

The max–cut problem is well known to be equivalent to 0-1 quadratic programming. However, we are interested in the more general case of constrained quadratic programming,

$$
\begin{array}{llll}
\text{(CQP)} & \text{Maximize} & x^t C x & \\
& \text{subject to} & Ax \leq b & \\
& & x^t A_i x \leq a_i & i = 1 \dots k \\
& & x \in \{0,1\}^n , &
\end{array}
$$

with C being an arbitrary symmetric $n \times n$ matrix, A an $m \times n$ matrix, $b \in \Re^n$. The A_i are symmetric $n \times n$ matrices, and $a_i \in \Re$. Since $x_i^2 = x_i$, we may assume without loss of generality that there is no linear term in the cost function, or to put it differently, the linear term consists of the main diagonal of the matrix C. We will use ideas from Lovás et al. [15] and Balas et al. [2, 3, 4] to derive tractable relaxations of (CQP). It turns out that these relaxations have equivalent formulations in the max–cut setting. We can therefore exploit the properties of the cut–polytope for the 0-1 programming case, too.

To demonstrate the quality and potential of this approach we give computational results for two specific applications, the first being the maximum clique problem. Let $G(V,E)$ denote an undirected loopless graph on n vertices $V = \{1,\dots,n\}$ with edge set E. We want to find the maximum clique with respect to weights w_i on the nodes,

$$
\begin{array}{llll}
\text{(MCL)} & \text{Maximize} & w^T x & \\
& \text{subject to} & x_i x_j = 0 & \forall (ij) \notin E \\
& & x \in \{0,1\}^n . &
\end{array}
$$

The semidefinite relaxation is indeed the ϑ–function, we will improve it further by adding triangle facets of the cut–polytope.

The second problem is the quadratic knapsack problem, i.e. quadratic programming with one inequality constraint,

$$
\begin{array}{lll}
\text{(QK)} & \text{Maximize} & x^t C x \\
& \text{subject to} & a^t x \leq b \\
& & x \in \{0,1\}^n , a \in \Re^n, b \in \Re.
\end{array}
$$

We will show the effect of the different possibilities to model the constraint and again we will improve the relaxation by adding triangle facets.

In Section 2 we model the 0-1 variables using semidefinite matrices. We show the equivalence of this approach to a semidefinite relaxation for (-1,1) variables in Section 3. Section 4 describes the modeling of linear constraints. To illustrate the techniques we give possible semidefinite relaxations of the two applications (MCL) and (QK) together with numerical results in Section 5.

2 Modeling 0-1 Variables

Let $B := \{0,1\}^n$ denote the set of binary n vectors. We first recall various ways to model this set by semidefinite matrices. The key idea from [3, 15] consists in using an $n \times n$ symmetric matrix variable Y having the following interpretation ($x \in B$):

$$y_{ij} = x_i x_j \quad i \neq j, \quad \text{and}$$

$$y_{ii} = x_i^2 = x_i.$$

The following linear inequalities constraining the elements of Y exploit 0-1 properties of x, i.e. $x_i x_j \geq 0$, $x_i(1-x_j) \geq 0$, and $(1-x_i)(1-x_j) \geq 0$, respectively,

$$y_{ij} \geq 0 \tag{1}$$

$$y_{ii} \geq y_{ij} \tag{2}$$

$$1 + y_{ij} \geq y_{ii} + y_{jj}. \tag{3}$$

These linear constraints are straightforward, and are also used in e.g. [15]. A further strengthening is obtained by observing that $Y = xx^T$ is positive semidefinite, $Y \succeq 0$. In [3] Balas et al point out, that there is a much more general procedure to obtain semidefiniteness conditions. Since $(x+v)(x+v)^T$ is semidefinite for any vector v, Y can be constrained to satisfy

$$Y + \operatorname{diag}(Y)v^T + v\operatorname{diag}(Y)^T + vv^T \succeq 0,$$

for any vector v. The inequality can be rewritten as

$$Y + (\operatorname{diag}(Y) + v)(\operatorname{diag}(Y) + v)^T - \operatorname{diag}(Y)\operatorname{diag}(Y)^T \succeq 0.$$

The left hand side is minimized for $v = -\operatorname{diag}(Y)$. Thus, the intersection over all v is characterized by

$$Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T \succeq 0.$$

This condition is nonlinear in Y, but fortunately, see e.g. [15], via Schur complement it is equivalent to

$$\mathcal{A}(Y) + E_{n+1} \succeq 0, \tag{4}$$

where the linear operator $\mathcal{A}(Y)$ is defined by

$$\mathcal{A}(Y) := \begin{pmatrix} Y & \operatorname{diag}(Y) \\ \operatorname{diag}(Y)^T & 0 \end{pmatrix} \tag{5}$$

and $E_{n+1} := e_{n+1}e_{n+1}^T$ is the diagonal matrix of size $n+1$ formed from the canonical unit vector e_{n+1}. Therefore a good way to derive a semidefinite model is to combine the linear inequalities on Y with the semidefiniteness condition $\mathcal{A}(Y) + E_{n+1} \succeq 0$.

3 (0,1) Variables Versus (-1,1) Variables

Up to now we only looked at integer programs in (0,1) variables. Recently, relaxations involving (-1,1) variables turned out to be very successful, see e.g.[6, 10, 8]. We will now show how these relaxations relate to each other.

Let C be a symmetric $n \times n$ matrix.

$$(P) \quad \max\ y^T C y \quad \text{such that} \quad y \in \{0,1\}^n.$$

The semidefinite relaxation resulting from the previous section leads to

$$(R) \quad \max\ \operatorname{tr}(CY) \quad \text{such that} \quad W := \mathcal{A}(Y) + E_{n+1} \succeq 0.$$

Now let us transform the problem into (-1,1) variables x.

$$x = 2y - e, \quad y = \frac{1}{2}(x+e)$$

with e the vector of all ones. The cost function translates into $y^T C y = x^T Q x - 2c^T x + d$, where $4Q = C$, $4c = -Ce$, $4d = e^T C e$. Define the $(n+1) \times (n+1)$ matrix

$$D := \begin{pmatrix} Q & -c \\ -c^T & d \end{pmatrix}.$$

Then (P) is equivalent to

$$(P') \quad \max\ z^T D z \quad \text{such that} \quad z \in \{-1,1\}^{n+1}.$$

Note that without loss of generality we can require $z_{n+1} = 1$ because the function is symmetric, $f(z) = f(-z)$. This transformation was already used in in [5] to solve quadratic (0,1) minimization using the max-cut problem. A semidefinite relaxation for (P') is

$$(R') \quad \max\ \operatorname{tr} DZ \quad \text{such that} \quad \operatorname{diag}(Z) = e, \quad Z \succeq 0.$$

It was pointed out by [4] that the two relaxations (R), (R') are actually equivalent as well. The following proof and other important connections between both models are also contained in [13].

We will now show in addition to this equivalence, how linear constraints in one model can be translated into equivalent constraints in the other model.

In particular, we will see that the trivial constraints (1), (2), (3) translate into "triangle inequalities" through node $n+1$. Moreover, we show how general linear inequalities for the (-1,1) version can be translated into linear inequalities for the (0,1) model, and vice versa.

The main observation consists in the following relation between W from (R) and Z from (R'). Let us partition the matrices as follows:

$$Z := \begin{pmatrix} X & x_0 \\ x_0^T & 1 \end{pmatrix} \qquad \text{and} \qquad W := \begin{pmatrix} Y & y_0 \\ y_0^T & 1 \end{pmatrix}.$$

Feasibility implies $\operatorname{diag}(X) = e, \operatorname{diag}(Y) = y_0$. Suppose Z and Y are related by

$$x_0 = 2y_0 - e \tag{6}$$

$$X = 4Y + (2y_0 - e)(2y_0 - e)^T - 4y_0 y_0^T \tag{7}$$

Note that these relations show that (y_0, Y) uniquely determines (x_0, X). Moreover, given (x_0, X), the equations (6), (7) can be solved for (y_0, Y) by

$$\begin{aligned} 2y_0 &= x_0 + e \\ 4Y &= X + (x_0 + e)(x_0 + e)^T - x_0 x_0^T \end{aligned}$$

The last equation can also be written as

$$4(Y - y_0 y_0^T) = X - x_0 x_0^T,$$

showing that $Z \succeq 0$ if and only if $W \succeq 0$. Similarly we have $\operatorname{diag}(X) = e$ if and only if $\operatorname{diag}(4Y - x_0 e^T - e x_0^T) = 2e$ which is equivalent to $\operatorname{diag}(Y) = y_0$. Finally, $\operatorname{tr}(DZ) = \operatorname{tr}(CY)$ holds if Z and Y are related as above.

The linear map (6), (7) is therefore a bijection between the feasible sets of (R) and (R'), preserving the objective function value. Moreover, (0,1) solutions are mapped into (-1,1) solutions, i.e. $X = x_0 x_0^T$ is equivalent to $Y = y_0 y_0^T$. Indeed, De Simone proved in [7] that this map is a bijection between the cut-polytope and the boolean quadric polytope. We can therefore use this map to transform linear inequalities from one model into the other. Writing out the map in detail we have

$$4y_{ij} = x_{ij} + x_{n+1,i} + x_{n+1,j} + 1,$$

which simplifies to

$$2y_{ii} = x_{n+1,i} + 1$$

in case $i = j$. Now note that for (1)–(3),

$$\begin{aligned} y_{ij} \geq 0 \quad &\iff \quad x_{ij} + x_{n+1,i} + x_{n+1,j} + 1 \geq 0 \\ y_{ii} - y_{ij} \geq 0 \quad &\iff \quad -x_{ij} + x_{n+1,i} - x_{n+1,j} + 1 \geq 0 \\ 1 + y_{ij} - y_{ii} - y_{jj} \geq 0 \quad &\iff \quad x_{ij} - x_{n+1,i} - x_{n+1,j} + 1 \geq 0. \end{aligned}$$

Thus, (1)–(3) in the (0,1) model correspond precisely to "triangle inequalities" through node $n+1$ in the (-1,1) model. Going the other direction, it is now straighforward to see that for general triangle inequalities in the (-1,1) model,

$$\begin{aligned} x_{ij} + x_{ik} + x_{jk} + 1 \geq 0 \quad &\iff \quad y_{ij} + y_{ik} + y_{jk} - (y_{ii} + y_{jj} + y_{kk}) + 1 \geq 0 \\ x_{ij} - x_{ik} - x_{jk} + 1 \geq 0 \quad &\iff \quad y_{ij} - y_{ik} - y_{jk} + y_{kk} \geq 0. \end{aligned}$$

The last relation can be interpreted as $(y_i - y_k)(y_j - y_k) \geq 0$, which is true for all (0,1) vectors y. In principle we can employ this transformation for any facet defining (or valid) inequality of the cut–polytope to obtain facet defining (or valid) inequalities for the quadratic case. Let us collect the linear inequalities such as (1), which are generic for (0,1) problems, into

$$\mathcal{G}(Y) - g \geq 0.$$

As a consequence of the mapping, both models are equivalent not only for the relaxations (R) and (R'), but also if additional linear inequalities are added. We can choose whatever model seems to be favorable from an implementational point of view.

4 Modeling Linear (In)Equalitites

So far we have investigated the problem of modeling the integrality conditions. Next we look at ways to combine Y with additional inequalities, which may possibly be present in (CQP).

It is only natural to represent the quadratic constraints $x^T A_i x \leq a_i$ as linear inequalities $\mathrm{tr}(A_i Y) \leq a_i$ for $i = 1 \ldots k$. To avoid excessive indexing we collect all constraints of this form in

$$\mathcal{Q}(Y) - a \leq 0.$$

Linear inequalities of the form $a_i^T x - b_i \leq 0$ can be embedded in different ways. The first and obvious possibility is to model them on the diagonal of Y,

$$\mathrm{tr}(\mathrm{Diag}(a_i)Y) \leq b_i.$$

Balas et al. [3] propose to multiply the linear inequalities by the nonnegative quantities x_j and $1 - x_j$. This leads to

$$(a_i^T x)x_j - b_i x_j \leq 0, \quad (a_i^T x)(1 - x_j) - b_i(1 - x_j) \leq 0, \;\; \forall i, j.$$

Let e_j denote column j of the identity matrix I_n. We define the following symmetric matrices that will be used to represent the above conditions as linear constraints on Y.

$$M_{ij} := \frac{1}{2}(a_i e_j^T + e_j a_i^T) - b_i e_j e_j^T,$$

$$N_{ij} := \mathrm{Diag}(a_i) - \frac{1}{2}(a_i e_j^T + e_j a_i^T) + b_i e_j e_j^T.$$

Then the linear constraints on x translate into linear constraints on Y. We get

$$\mathrm{tr}(M_{ij}Y) \leq 0, \quad \mathrm{tr}(N_{ij}Y) - b_i \leq 0, \quad \forall i, j.$$

A third possibility, mentioned in [15], is to multiply the linear inequalities with one another,

$$(a_i^T x - b_i)(a_j^T x - b_j) \geq 0.$$

Again we can represent this product as a linear constraint on Y. To simplify notation let us again collect all the different representations (or some subset of them) in the condition

$$\mathcal{B}(Y) - b \leq 0.$$

We finally obtain the following semidefinite program as relaxation for (CQP)

$$\text{(SCQP)} \qquad \begin{array}{rl} \text{Maximize} & \operatorname{tr}(CY) \\ \text{subject to} & \mathcal{B}(Y) - b \leq 0 \\ & \mathcal{Q}(Y) - a \leq 0 \\ & \mathcal{G}(Y) - g \geq 0 \\ & \mathcal{A}(Y) + E_{n+1} \succeq 0. \end{array}$$

The (weak) dual semidefinite program can easily be derived and reads

$$\text{(DSCQP)} \qquad \begin{array}{rl} \text{Minimize} & u^T b + v^T a - w^T g + \operatorname{tr}(V E_{n+1}) \\ \text{subject to} & C - \mathcal{B}^T(u) - \mathcal{Q}^T(v) + \mathcal{G}^T(w) + \mathcal{A}^T(V) \preceq 0 \\ & u \geq 0, \ v \geq 0, \ w \geq 0, \ V \succeq 0. \end{array}$$

There is some degree of freedom which of the generic constraints of $\mathcal{G}$ are included, and also, it may be computationally more efficient, not to include all of the constraints from $\mathcal{B}$ at once. We show on two examples the effect of this relaxation.

5 Examples

Example 1. We first look at the problem of finding the largest clique in a graph, (MCL). According to our model we represent the constraints $x_i x_j = 0$ for $(ij) \notin E$ by $y_{ij} + y_{ji} = 0$ or, equivalently, $\operatorname{tr}(E_{ij} Y) = 0$ with $E_{ij} = e_i e_j^T + e_j e_i^T$. The cost function is modeled as $\operatorname{tr}(\operatorname{Diag}(w) Y)$ where $\operatorname{Diag}(\cdot)$ transforms a vector to a diagonal matrix. The "easiest" semidefinite relaxation is now

$$\text{(SMCL)} \qquad \begin{array}{rl} \text{Maximize} & \operatorname{tr}(\operatorname{Diag}(w) Y) \\ \text{subject to} & \operatorname{tr}(E_{ij} Y) = 0 \quad \text{for } (ij) \notin E \\ & \mathcal{A}(Y) + E_{n+1} \succeq 0, \end{array}$$

and is equivalent to the weighted ϑ–function by Lemma 2.17 of [15]. Note that the condition $y_{ij} = 0$ corresponds to requiring $x_{ij} + x_{n+1,i} + x_{n+1,j} + 1 = 0$ in the (-1,1) model. i.e. the corresponding triangle inequality must be satisfied with equality, see also [13]. The obvious interpretation for the (-1,1) model is that not both vertices, i and j, may be in the same set with vertex $n+1$.

We obtain a better relaxation if we add all possible triangle inequalities. Since there are exactly $4\binom{n+1}{3}$ inequalities of this type, this is still a tractable relaxation. Checking all possible triangles is of the same order of magnitude as inverting an $n \times n$–matrix, a basic operation carried out several times within each iteration of the interior point code.

Since $Y = \frac{1}{n+1} I$ is always an "interior point" of (SMCL) strong duality is satisfied, i.e. the optimal solutions of the primal and the dual are identical.

However, $\frac{1}{n+1}I$ is only the convex combination of $n+1$ feasible (0,1) solutions, and we cannot expect that this solution is in the interior with respect to all feasible inequalities. If we start with the arithmetic mean of all possible one cliques, two cliques, and the zero clique, then the point is in the relative interior of the convex hull of $n+|E|+1$ affinely independent feasible solutions. Thus, this point is in the relative interior of the convex hull of all feasible solutions and can be used as starting point for any setting, i.e. no matter which facet defining inequalities we start with, we will always have a starting point in the relative interior of the feasible set with $\mathcal{A}(Y)+E_{n+1}\succ 0$.

We illustrate the effect of the triangle inequalities on an example of size 70 with 224 non–edges. For efficiency reasons we do not add all triangle inequalities at once but include them succesively dependent on the amount of violation. We use the interior point algorithm of [10], the practical implementation follows [9]. The most expensive operation within each iteration is the factorization of a positive definite matrix of size $|V|$ + #non-edges + #ineq. The computational results are displayed in Table 1. The largest clique found so far is of size 23. % gives the relative error ($= bound/23-1$) in percent. Note that the value of the ϑ–function (27.0292) is radically improved by the first few triangle inequalities (the number is given in column *ineq*). Afterwards we do not get much improvement by further reducing the maximal violation (given in column *viol*) of the triangle inequalities.

iter	bound	%	hh:mm:ss	ineq	viol
17	27.0292	17.51	1:07	0	0.1094
78	25.6311	11.44	26:29	381	0.0495
135	25.4984	10.86	1:52:45	669	0.0251
194	25.4707	10.74	4:40:24	839	0.0150

Table 1. Development of upper bound for maximum clique.

In Table 2 we give some more results for random graphs with fixed edge probability. $|V|$ gives the number of nodes of the graph. *prob* is the probability with which each edge is contained in the graph. *non-edges* is the number of edges not contained in the graph. *lbnd* gives the size of the largest clique found, which is not necessarily optimal. ϑ is the value of the theta function for this graph, the relative error with respect to *lbnd* is given in brackets. *ubnd* is the upper bound obtained after one phase of succesively adding violated triangle inequalities. Finally, *ineq* gives the number of triangle inequalities added.

Example 2. Next we look at the quadratic knapsack problem (QK). W.l.o.g. we assume $a\geq 0$. Therefore we can add the additional constraints $\mathrm{tr}(E_{ij}Y)=0$ for $a_i+a_j>b$. Again, the arithmetic mean of all feasible one item, two items, and zero items solutions is in the relative interior of the convex hull of all feasible

$\|V\|$	prob	non-edges	lbnd	ϑ (%)	ubnd (%)	ineq
250	0.98	659	104	110.67 (6.41)	108.39 (4.22)	953
250	0.98	637	99	110.22 (11.33)	107.66 (8.75)	940
250	0.98	625	110	112.84 (2.58)	111.03 (0.94)	939
200	0.97	607	80	84.80 (6.00)	82.92 (3.65)	758
200	0.97	605	80	85.37 (6.71)	83.47 (4.34)	747
200	0.97	588	77	83.63 (8.61)	81.60 (5.97)	757
150	0.95	548	49	57.09 (16.51)	55.67 (13.61)	499
150	0.95	588	48	56.50 (17.71)	54.96 (14.50)	515
150	0.95	561	49	56.85 (16.02)	55.22 (12.69)	576
100	0.90	475	26	32.66 (25.62)	31.83 (22.42)	366
100	0.90	464	31	33.87 (9.26)	32.97 (6.35)	358
100	0.90	465	29	33.43 (15.28)	32.43 (11.83)	330

Table 2. improvement for random graphs by adding some triangles.

0-1 solutions.

To illustrate the benefit of modeling the constraints as described in Section 4 we first model the inequality constraint as a constraint on the main diagonal of Y,

$$\text{(SQK1)} \qquad \begin{array}{rl} \text{Maximize} & \operatorname{tr}(CY) \\ \text{subject to} & \operatorname{tr}(\operatorname{Diag}(a)Y) \leq b \\ & \operatorname{tr}(E_{ij}Y) = 0 \quad \text{for } a_i + a_j > b \\ & \mathcal{A}(Y) + E_{n+1} \succeq 0. \end{array}$$

We give all numerical results in Table 3 for an example of size 30. The optimal solution of the problem is 1580, % gives the relative error ($= bound/1580 - 1$) in percent. The upper bound obtained by (SQK1) is displayed on the first line of Table 3. The next two lines give the development of the bound as violated triangle inequalities are added successively. The last column displays the maximal violation of the triangle inequalities.

Next we add the constraint by squaring it, $a^T x x^T a \leq b^2$. This makes use of the fact that $a \geq 0$.

$$\text{(SQK2)} \qquad \begin{array}{rl} \text{Maximize} & \operatorname{tr}(CY) \\ \text{subject to} & \operatorname{tr}(aa^TY) \leq b^2 \\ & \operatorname{tr}(E_{ij}Y) = 0 \quad \text{for } a_i + a_j > b \\ & \mathcal{A}(Y) + E_{n+1} \succeq 0. \end{array}$$

In Table 3 we can observe considerable improvements to (SQK1). Even with all triangle inequalities satisfied the bound of (SQK1) cannot compete with the simple bound of (SQK2). By adding violated triangle inequalities the gap can now be reduced to reasonable size.

Finally we model the linear constraint by all possible variants M_j and N_j

(see Section 4),

$$\begin{array}{llll} (\mathrm{SQK3}) & \text{Maximize} & \mathrm{tr}(CY) & \\ & \text{subject to} & \mathrm{tr}(M_j Y) \leq 0 & \text{for } j = 1, \ldots, n \\ & & \mathrm{tr}(N_j Y) \leq b & \text{for } j = 1, \ldots, n \\ & & \mathrm{tr}(E_{ij} Y) = 0 & \text{for } a_i + a_j > b \\ & & \mathcal{A}(Y) + E_{n+1} \succeq 0. & \end{array}$$

The numerical result for our running example is again given in Table 3 and is indeed an improvement with respect to $(SQK2)$, but this time the bound is definitely more expensive to compute. Once more, the triangle inequalities tighten the gap considerably. The final bound is still a few units off from the optimal solution 1580. Helmberg, Rendl, and Weismantel are currently investigating possibilities to close this gap even further by adding knapsack cutting planes. The first results are incouraging [11].

relaxation	iter	bound	%	mm:ss	ineq	viol
(SQK1)	14	2227.58	40.98	2	1	0.1203
	57	2073.04	31.21	28	131	0.02413
	99	2070.24	31.03	2:26	257	5.5e-06
(SQK2)	14	1853.65	17.28	3	1	0.117
	58	1615.14	2.22	28	131	0.005227
	102	1615.05	2.22	2:01	231	2.2e-06
(SQK3)	13	1793.31	13.48	17	60	0.124
	57	1594.20	0.89	2:04	190	0.04752
	101	1585.60	0.35	7:24	320	0.0006229

Table 3. different relaxations.

Conclusions. With respect to computing times we are currently not able to compete with state of the art special purpose codes for either problem. However, the examples illustrate that the approach is not only computationally feasible for very diverse problems but indeed leads to serious improvements of the bounds. Any further progress in understanding the max-cut problem is of great importance for integer programming in general. But max-cut facets alone are certainly not sufficient. General separation procedures which take care of additional linear inequalities will be needed. Semidefinite programming offers new possibilities for the design of these separation procedures. Therefore we believe that this is only a first step into exploring the use of semidefinite programming for integer programs.

References

1. F. ALIZADEH. *Interior point methods in semidefinite programming with applications to combinatorial optimization.* Technical Report TR–93–050, International Computer Science Institute, Berkley, Sept. 1993.
2. E. BALAS, S. CERIA and G. CORNUEJOLS. A lift-and-project cutting plane algorithm for mixed 0-1 programs, *Mathematical Programming* 58:295-324, 1993.
3. E. BALAS, S. CERIA, G. CORNUEJOLS and G. PATAKI. *Polyhedral methods for the maximum clique problem.* Technical Report, Carnegie Mellon University, Pittsburgh, USA, 1994.
4. E. BALAS, S. CERIA, G. CORNUEJOLS and G. PATAKI. *Updated semi-definite constraints.* Technical Report, Carnegie Mellon University, Pittsburgh, USA, 1994.
5. F. BARAHONA, M. JÜNGER and G. REINELT. Experiments in quadratic 0-1 programming, *Mathematical Programming* 44:127-137, 1989.
6. C. DELORME and S. POLJAK. Laplacian eigenvalues and the max-cut problem, *Mathematical Programming*, 62(3):557-574, 1993.
7. C. DE SIMONE. The cut polytope and the boolean quadric polytope. *Discrete Mathematics*, 79:71–75, 1989.
8. M.X. GOEMANS and D.P. WILLIAMSON. *.878-approximation algorithm for max–cut and max–2sat.* Technical Report, MIT, 1993.
9. C. HELMBERG. *An interior point method for semidefinite programming and max–cut bounds.* PhD thesis, Graz University of Technology, Rechbauerstraße 12, A-8010 Graz, Austria, Oct. 1994.
10. C. HELMBERG, F. RENDL, R. J. VANDERBEI, and H. WOLKOWICZ. An interior–point method for semidefinite programming. *SIAM Journal on Optimization.* To appear.
11. C. HELMBERG, F. RENDL, and R. WEISMANTEL. *Quadratic knapsack relaxations using cutting planes and semidefinited programming.* Working paper.
12. M. LAURENT and S. POLJAK. *On a positive semidefinite relaxation of the cut polytope.* Rapport de Recherche 27, Laboratoire d'Informatique, URA 1327 du CNRS, Département de Mathématiques et d'Informatique, Ecole Normale Supérieure, Dec. 1993.
13. M. LAURENT, S. POLJAK, and F. RENDL. Connections between semidefinite relaxations of the max-cut and stable set problems. Technical Report BS-R9502, Centrum voor Wiskunde en Informatica, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands, Jan. 1995.
14. L. LOVÁSZ. On the Shannon Capacity of a Graph. *IEEE Trans. Inform. Theory,* IT–25, 1979.
15. L. LOVÁSZ and A. SCHRIJVER. Cones of matrices and set functions and 0-1 optimization. *SIAM J. Optimization,* 1(2):166–190, 1991.

This article was processed using the LaTeX macro package with LLNCS style