



Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7, D-14195 Berlin

Christoph Helmberg

Fixing Variables in Semidefinite Relaxations

Fixing Variables in Semidefinite Relaxations

Christoph Helmberg*

December 3, 1996

Abstract

The standard technique of reduced cost fixing from linear programming is not trivially extensible to semidefinite relaxations as the corresponding Lagrange multipliers are usually not available. We propose a general technique for computing reasonable Lagrange multipliers to constraints which are not part of the problem description. Its specialization to the semidefinite $\{-1, 1\}$ relaxation of quadratic 0-1 programming yields an efficient routine for fixing variables. The routine offers the possibility to exploit problem structure. We extend the traditional bijective map between $\{0, 1\}$ and $\{-1, 1\}$ formulations to the constraints such that the dual variables remain the same and structural properties are preserved. In consequence the fixing routine can efficiently be applied to optimal solutions of the semidefinite $\{0, 1\}$ relaxation of constrained quadratic 0-1 programming, as well. We provide numerical results showing the efficacy of the approach.

1 Introduction

The power of semidefinite relaxations of combinatorial problems has been recognized already in the seventies [21]. At that time it was not considered likely that practical algorithms for computing the associated bounds would ever be available. Research was primarily of theoretic nature [12]. A new rush of theoretical results in the early nineties [22, 6, 26] and the development of interior point algorithms for semidefinite programming [17, 24, 1, 28, 15, 19, 25] spurred interest for the field. Within short time several results in approximation theory [10, 23, 9, 5, 8] were published giving further evidence for the high quality of semidefinite programming bounds. Although a general framework for designing semidefinite relaxations of linear and quadratic 0-1 programming problems is available [22, 13, 16], only few papers presenting computational experience are published so far [14, 18, 31, 30] (all are based on interior point codes). The bounds prove to be of good quality in practice, but implementations suffer from the high computational cost involved in solving semidefinite programs.

The task of solving hard combinatorial problems to optimality leads naturally to branch and bound (or branch and cut). In this setting the efficiency of an expensive bound hinges on the tradeoff between the number of branch and bound nodes and the computation time needed for each node. Indeed, in spite of its high cost the semidefinite programming relaxation outperforms any other approach in the case of unconstrained 0-1 quadratic programming with dense cost matrices [14]. Yet for problems of more than a hundred 0-1 variables the approach must be considered impractical. Even though the number of branch and bound nodes may seem reasonably small the computation time needed to solve the semidefinite relaxation

*Konrad Zuse Zentrum für Informationstechnik Berlin, Takustraße 7, D-14195 Berlin, Germany. e-mail: helmberg@zib.de, URL: <http://www.zib.de/helmberg>.

becomes prohibitive. The sharp increase in computation time with growing dimension calls for routines that fix the “easy” variables quickly, thereby reducing the dimension of remaining problems in the branch and bound tree (not necessarily the number of nodes). In [14] we did not know how to exploit the dual to fix variables. Even in the case that the relaxation displayed its preferences for some variables at the root node we had to run through all dimensions till the overall bound was good enough.

In linear programming relaxations fixing variables by reduced costs is a standard procedure. For quadratic 0-1 programming problems the initial linear relaxations usually include the box constraints $0 \leq y_{ij} \leq 1$ (y_{ij} is to be understood as the linearization of $y_i \cdot y_j$). If the optimal solution of the current linear relaxation yields $y_{ij} = 1$, say, then a bound for the problem with $y_{ij} = 0$ is obtained via the Lagrange multiplier or dual variable corresponding to the constraint $y_{ij} \leq 1$.

The standard semidefinite programming relaxation for quadratic 0-1 programming already implies some of the box-constraints. Consequently these are not included in the relaxation and the corresponding Lagrange multipliers are unknown. Yet if the optimal solution of the semidefinite relaxation displays some $y_{ij} = 1$ then there must be some corresponding active constraint buried in the semidefiniteness constraint. It is the goal of this paper to present a practical method for extracting this information. It is worth noting that the considerations to come are completely independent of the actual algorithm used to solve the semidefinite relaxation.

There are two standard models for quadratic 0-1 programming, one formulated in $\{0, 1\}$ variables, the other in $\{-1, 1\}$ variables. Both lead, in a canonical way, to semidefinite relaxations that are slightly different in appearance. In particular the dual of the $\{-1, 1\}$ relaxation allows for a very efficient routine for fixing variables. It is well known that both problems and their primal relaxations are equivalent [7, 13, 20]. The dual variables, however, will differ for varying representations of the same primal set. We present a canonical transformation between the constraints of both formulations such that the dual variables are the same for both. If the $n + 1$ fundamental constraints of the $\{0, 1\}$ relaxation are modeled correctly this enables us to use the fixing procedure of the $\{-1, 1\}$ formulation even for optimal solutions computed in the $\{0, 1\}$ setting.

In Section 2 we introduce the semidefinite relaxation of quadratic 0-1 programming in $\{-1, 1\}$ variables which motivated the considerations to follow. Section 3 provides the theoretical framework for extracting information from the dual in the general setting of semidefinite programming. Section 4 explains the practical difficulties in implementing the theoretic approach and presents an efficient alternative within the $\{-1, 1\}$ setting. In Section 5 the equivalence transformation between $\{0, 1\}$ and $\{-1, 1\}$ formulations is extended to the constraints such that dual variables and structural properties are preserved. Section 6 presents numerical results underlining the efficacy of the fixing routine. We conclude the paper in Section 7.

Notation

\mathbb{R}, \mathbb{R}^n	real numbers, real column vector of dimension n
$M_{m,n}, M_n$	$m \times n, n \times n$ real matrices
S_n	$n \times n$ symmetric real matrices
$A \succeq 0, A \succ 0$	A is (symmetric) positive semidefinite, positive definite
I, I_n	identity of appropriate size or of size n
e_i	i -th column of I
e	vector of all ones of appropriate dimension

$\lambda_i(A)$	i -th eigenvalue of $A \in M_n$, usually $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$
$\lambda_{\min}(A), \lambda_{\max}(A)$	minimal and maximal eigenvalue of A
Λ_A	diagonal matrix with $(\Lambda_A)_{ii} = \lambda_i(A)$
$\text{tr}(A)$	trace of $A \in M_n$, $\text{tr}(A) = \sum_{i=1}^n a_{ii} = \sum_{i=1}^n \lambda_i(A)$
$\langle A, B \rangle$	inner product in $M_{m,n}$, $\langle A, B \rangle = \text{tr}(B^T A)$
$\text{rank}(A)$	rank of a A
$\text{diag}(A)$	$\text{diag}(A) = [a_{11}, \dots, a_{nn}]^T$
$\text{Diag}(v)$	diagonal matrix with diagonal v
$\mathcal{A}(X)$	$= [\langle A_1, X \rangle, \dots, \langle A_m, X \rangle]^T$ for given $A_i \in S_n$
$\mathcal{A}^T(y)$	$= \sum_{i=1}^m y_i A_i$

Unless explicitly stated otherwise all matrices considered are symmetric and vectors are columns.

2 Quadratic 0-1 Programming in $\{-1, 1\}$ Variables

Two canonical formulations of quadratic 0-1 programming appear in the literature, one in terms of $\{0, 1\}$ variables and one in $\{-1, 1\}$ variables. For our purposes the semidefinite relaxation of the $\{-1, 1\}$ formulation (which is better known as the semidefinite relaxation of max-cut) is more convenient. We will return to the $\{0, 1\}$ formulation and the equivalence of both in Section 5.

The natural interpretation for a vector $x \in \{-1, 1\}^n$ is that of a partition vector. Indices having the same sign belong to the same set. Formulated with respect to the product $x_i x_j$, i and j belong to the same set if $x_i x_j$ is equal to one, and i and j belong to opposite sets if $x_i x_j$ is minus one. The combinatorial problem to be investigated reads

$$(MC) \quad \max x^T C x \quad \text{s.t.} \quad x \in \{-1, 1\}^n$$

The standard semidefinite relaxation is derived by observing that $x^T C x = \langle C, x x^T \rangle$. For all $\{-1, 1\}^n$ vectors, $x x^T$ is a positive semidefinite matrix with all diagonal elements equal to one. We relax $x x^T$ to $X \succeq 0$ and $\text{diag}(X) = e$ and obtain the following primal dual pair of semidefinite programs,

$$(PMC) \quad \begin{array}{ll} \max & \langle C, X \rangle \\ \text{s.t.} & \text{diag}(X) = e \\ & X \succeq 0 \end{array} \quad (DMC) \quad \begin{array}{ll} \min & e^T u \\ \text{s.t.} & C + Z - \text{Diag}(u) = 0 \\ & Z \succeq 0. \end{array}$$

The relaxation can be strengthened by adding a few of the so called triangle inequalities,

$$\begin{aligned} x_{ij} + x_{ik} + x_{jk} &\geq -1 \\ x_{ij} - x_{ik} - x_{jk} &\geq -1 \\ -x_{ij} + x_{ik} - x_{jk} &\geq -1 \\ -x_{ij} - x_{ik} + x_{jk} &\geq -1 \end{aligned}$$

for $i < j < k$ from $\{1, \dots, n\}$. For later reference we point out that, by exploiting the ones on the diagonal of X , these inequalities can be written in the form

$$(1) \quad v^T X v \geq 1,$$

with $v \in \mathbb{R}^n$ having only three non-zero entries, each either +1 or -1.

A bound of this kind is used in [14] in a branch and cut scheme. This runs as follows. The relaxation is solved for the initial problem and a good integral solution is generated using the solution of the relaxed problem. In case the bound is close

enough to the best integral solution found, stop (in the recursive step proceed with an open problem in the branch and bound tree). Otherwise select two indices i and j to generate two subproblems, one with i and j in the same set, one with i and j in opposite sets. Both subproblems can be expressed as quadratic $\{-1, 1\}^{n-1}$ problems. Proceed recursively.

If the bound is not good enough to fathom the node, but $x_{ij} = 1$ ($x_{ij} = -1$) for some $i \neq j$ in the optimal solution of the relaxation we can expect that indeed i and j belong together (apart). If we force the opposite a drop in the bound is to be expected. Can we prove that this drop will be large enough without recomputing the bound for this case?

In a linear cutting plane algorithm for the $\{-1, 1\}$ model the constraints $-1 \leq x_{ij} \leq 1$ are typically included in the initial relaxation. If the optimal solution of the linear relaxation exhibits $|x_{ij}| = 1$ then the dual variable of the corresponding active constraint yields a lower bound on the change of the objective value that would result from forcing x_{ij} to the opposite sign. This bound may suffice to prove that the current value of x_{ij} is correct for all optimal solutions of (MC).

In the semidefinite relaxation (PMC) the constraints $-1 \leq x_{ij} \leq 1$ are already implied by the diagonal constraints and the semidefiniteness of X . Therefore they are not included in the semidefinite relaxation and the corresponding dual variables are not available.

However, we can associate with each active constraint $x_{ij} \geq -1$ or $x_{ij} \leq 1$ an active constraint $v^T X v \geq 0$ from the set of constraints ensuring the positive semidefiniteness of X as follows. Let $|x_{ij}| = 1$ for some $i \neq j$ in the optimal solution (X^*, u^*, Z^*) of the current relaxation. Then the vector $v \in \mathbb{R}^n$ with

$$(2) \quad v_k = \begin{cases} 1 & k = i \\ -\text{sgn}(x_{ij}) & k = j \\ 0 & \text{otherwise} \end{cases},$$

is in the null space of X^* . Although this does not yet yield the Lagrange multiplier corresponding to the constraint $\langle v v^T, X \rangle \geq 0$ it suggests to look for it in the dual slack matrix Z^* . We will do so in the general setting of semidefinite programming.

3 The Theoretical Framework

Consider a standard primal dual pair of semidefinite programs,

$$(P) \quad \begin{array}{ll} \min & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \succeq 0 \end{array} \quad (D) \quad \begin{array}{ll} \max & \langle b, u \rangle \\ \text{s.t.} & \mathcal{A}^T(u) + Z = C \\ & Z \succeq 0. \end{array}$$

$\mathcal{A} : S_n \rightarrow \mathbb{R}^m$ is a linear operator of the form

$$\mathcal{A}(X) = \begin{bmatrix} \langle A_1, X \rangle \\ \vdots \\ \langle A_m, X \rangle \end{bmatrix},$$

with $A_i \in S_n$, $i = 1, \dots, m$. Its adjoint operator $\mathcal{A}^T : \mathbb{R}^m \rightarrow S_n$, satisfying $\langle \mathcal{A}(X), u \rangle = \langle X, \mathcal{A}^T(u) \rangle$ for all $X \in S_n$ and $u \in \mathbb{R}^m$, is

$$\mathcal{A}^T(u) = \sum_{i=1}^m u_i A_i.$$

We examine possibilities to extract duality information for equality or inequality constraints that are not explicitly given in the problem description. Assume that

optimal solutions X^* of (P) and (u^*, Z^*) of (D) are given. By $p^* = \langle C, X^* \rangle = \langle b, u^* \rangle$ we denote the optimal objective value.

We are interested in the following question: How much does the optimal value of (P) increase if an additional constraint $\langle A_0, X \rangle = b_0$ is added to the problem? We would like to bound this quantity without actually computing the optimal solution of the new problem.

Let u_0 denote the new dual variable associated with the new constraint. The corresponding primal dual pair reads

$$\begin{array}{ll}
 \min & \langle C, X \rangle \\
 \text{s.t.} & \langle A_0, X \rangle = b_0 \\
 & \mathcal{A}(X) = b \\
 & X \succeq 0
 \end{array}
 \quad
 \begin{array}{ll}
 \max & b_0 u_0 + \langle b, u \rangle \\
 \text{s.t.} & u_0 A_0 + \mathcal{A}^T(u) + Z = C \\
 & Z \succeq 0.
 \end{array}
 \tag{P_0} \tag{D_0}$$

Computing the optimal solution is as hard as solving the original problem. However, we do already know a “good” dual feasible solution for (D₀), namely $(u_0 = 0, u^*, Z^*)$. To improve this solution with reasonable effort we restrict ourselves to a line search along an ascent direction $(\Delta u_0, \Delta u, \Delta Z)$ with

$$\begin{aligned}
 b_0 \Delta u_0 + \langle b, \Delta u \rangle &> 0 \\
 \Delta u_0 A_0 + \mathcal{A}^T(\Delta u) + \Delta Z &= 0 \\
 Z + t \Delta Z &\succeq 0 \quad \text{for some } t \geq 0.
 \end{aligned}$$

To determine the best search direction is again as difficult as the problem itself. The choice of a good direction will depend on our understanding of the problem at hand.

Having fixed an ascent direction $(\Delta u_0, \Delta u, \Delta Z)$ it remains to compute the maximal step size t such that $Z + t \Delta Z$ is still positive semidefinite, because the objective function is linear. With

$$\begin{aligned}
 S &= C - \mathcal{A}^T(u^*) = Z^* \succeq 0 \\
 B &= \Delta u_0 A_0 + \mathcal{A}^T(\Delta u)
 \end{aligned}$$

the problem reduces to

$$(LS) \quad \max t \quad \text{s.t.} \quad S - tB \succeq 0.$$

Problems of this form appear as matrix pencils in the literature (see e.g. [11], Chapters 7.7, 8.7, and references therein). Indeed, the optimal t can be computed explicitly. To keep the paper self contained we include the main steps.

Let $P \Lambda_S P^T = S$ denote an eigenvalue decomposition of S with P an orthonormal matrix and Λ_S a diagonal matrix having the eigenvalues $\lambda_1(S) \geq \dots \geq \lambda_n(S)$ on its diagonal in this order. Then $S - tB \succeq 0$ is equivalent to

$$\Lambda_S - tP^T B P \succeq 0.$$

If the rank of S is k then $\lambda_i(S) = 0$ for $i = k + 1, \dots, n$. Multiplying the equation above by $D = \text{Diag}(\lambda_1(S)^{-\frac{1}{2}}, \dots, \lambda_k(S)^{-\frac{1}{2}}, 1, \dots, 1)$ from left and right we obtain

$$\begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} - tDP^T B P D \succeq 0.$$

Assuming that $t > 0$ exists we divide by t and impose the same block structure on $DP^T B P D$,

$$\begin{bmatrix} \frac{1}{t} I_k & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix} \succeq 0,$$

with $B_{11} \in M_k$, $B_{22} \in M_{n-k}$, $B_{12} \in M_{k,n-k}$ such that

$$\begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix} = DP^T BPD.$$

In case B_{12} and B_{22} are both zero, $\frac{1}{t} \geq \lambda_{\max}(B_{11})$ is the best choice (for $S \succ 0$ this specializes to $\frac{1}{t} = \lambda_{\max}(S^{-1}B)$, see [19]). Note, that for $\lambda_{\max}(B_{11}) \leq 0$ the problem is unbounded. If $-B_{22}$ is non-zero it must be positive semidefinite, otherwise $t = 0$ is the only feasible solution. If $-B_{22}$ is positive semidefinite with rank h we can apply a similar sequence of steps to obtain a condition

$$\begin{bmatrix} \frac{1}{t}I_k - B_{11} & \bar{B}_{12} & \bar{B}_{13} \\ \bar{B}_{12}^T & I_h & 0 \\ \bar{B}_{13}^T & 0 & 0 \end{bmatrix} \succeq 0.$$

If \bar{B}_{13} is non-zero then again t must be zero. Otherwise we can apply the Schur complement Theorem to obtain the condition

$$\frac{1}{t}I_k - B_{11} \succeq \bar{B}_{12}\bar{B}_{12}^T.$$

This yields $\frac{1}{t} \geq \lambda_{\max}(B_{11} + \bar{B}_{12}\bar{B}_{12}^T)$.

We specialize this general procedure to a case of particular importance in semi-definite programming. For the purpose of explanation assume that X^* and (u^*, Z^*) are a strictly complementary pair of optimal solutions, i.e. , $\text{rank}(X^*) + \text{rank}(Z^*) = n$ (these do not necessarily exist, see e.g. [2]). Furthermore let A_0 be a dyadic product vv^T for some $v \in \mathbb{R}^n$ with $\langle vv^T, X^* \rangle = 0$, i.e. , v is in the null space of X^* . vv^T may be interpreted as one of the active constraints ensuring the positive definiteness of X . The right hand side b_0 of the new constraint must be greater than zero, otherwise there is certainly no feasible primal solution for the new problem. As ascent direction we choose $\Delta u_0 = 1$ and $\Delta u = 0$. This yields the following line search problem,

$$\max t \quad \text{s.t.} \quad Z^* - tvv^T \succeq 0.$$

Because X^* and Z^* are strictly complementary solutions and v is in the null space of X we conclude that v lies in the span of the eigenvectors to non-zero eigenvalues of Z^* . Assume that $\text{rank}(Z^*) = k$ and let $P\Lambda_{Z^*}P = Z^*$ denote the eigenvalue decomposition of Z^* with $P \in M_{n,k}$, $P^T P = I_k$, and the spectrum of non-zero eigenvalues $\Lambda_{Z^*} \in S_k$. Then the maximal t is given by

$$t^* = \frac{1}{v^T P \Lambda_{Z^*}^{-1} P^T v}.$$

If in particular v happens to be an eigenvector of Z^* then t^* is the corresponding eigenvalue of Z^* . Relating this to linear programming we might formulate, the dual slack matrix Z^* subsumes the dual variables to the constraints generating the primal cone $X \succeq 0$.

This interpretation can be extended to the case that X^* and Z^* are not strictly complementary. For any vector v in the null space of X^* but not in the span of the non-zero eigenvectors of Z^* the optimal t is zero.

With respect to the semidefinite relaxation (PMC) the formula above suggests a convenient procedure to construct Lagrange multipliers for the constraints of form (2). Assuming that the eigenvalue decomposition of Z^* into $P\Lambda_{Z^*}P^T$ is available ($k = \text{rank}(Z^*)$, $P \in M_{n,k}$, $P^T P = I_k$), it is easy to check whether v is in the span of the eigenvectors P . If it is not, then $t^* = 0$, otherwise $t^* = -1/(v^T P \Lambda_{Z^*}^{-1} P^T v)$ is the best Lagrange multiplier for u fixed to u^* . The bound corresponding to forcing

i and j into opposite sets can be modeled by changing the right hand side of the (currently active) constraint $v^T X v = 0$ to $v^T X v = 4$ in the current relaxation. Therefore the bound obtained from the relaxation with i and j in opposite sets is less than or equal to $e^T u^* + 4t^*$. In theory this yields a very efficient algorithm for checking several pairs (i, j) . The eigenvalue decomposition has to be computed only once for all pairs, the evaluation for a single pair requires roughly $O(nk)$ arithmetic operations. However, in the next section we will see that practical implementations require a different approach.

4 A Practical Algorithm

In implementing the approach suggested in the previous section several difficulties are encountered. Indeed, we cannot expect any real world algorithm to deliver the true optimal solution (X^*, u^*, Z^*) of (PMC) for arbitrary cost matrices. For a computed solution $(\hat{X}, \hat{u}, \hat{Z})$ both, \hat{X} and \hat{Z} , will be (rather ill conditioned) full rank matrices. Even in case the gap $\langle \hat{X}, \hat{Z} \rangle$ between primal and dual solution is almost zero, it is difficult to decide which of the eigenvalues of \hat{X} and \hat{Z} will eventually converge to zero. The space spanned by the eigenvectors corresponding to the “non-zero” eigenvalues of \hat{X} and \hat{Z} may still differ substantially from the true eigenspaces of X^* and Z^* . Eigenvalue decompositions are difficult to compute because the eigenvalues tend to cluster at 0. The vectors v of (2) will neither be contained in the null space of \hat{X} nor in the space spanned by the “non-zero” eigenvectors of \hat{Z} because no $|x_{ij}|$ will be strictly one. In consequence the line search will allow for a very short step only and the approach fails.

However, in the case of (PMC) there is an obvious way to get around these difficulties. We mention, that the framework can be applied in the presence of additional primal constraints, as well, but as these have no influence on the considerations to follow, we ignore them here.

Within the branch and bound scenario let $(\hat{X}, \hat{u}, \hat{Z})$ be the solution computed for the relaxation of the current branch and bound node yielding the upper bound $e^T \hat{u}$ and let c^* denote the lower bound. Let i, j and v as in (2) with $v^T \hat{X} v$ almost zero. How much does the bound improve if we add the constraint $v^T \hat{X} v = 4$ to the current relaxation? We denote the Lagrange multiplier for the new constraint by u_0 . We would like to compute an upper bound, ideally smaller than c^* , for the problem

$$\begin{aligned} \min \quad & 4u_0 + e^T u \\ \text{s.t.} \quad & Z = u_0 v v^T + \text{Diag}(u) - C \\ & Z \succeq 0. \end{aligned}$$

Consider the situation of setting u_0 to some (negative) value required for achieving $4u_0 + e^T \hat{u} < c^*$. If $\hat{Z} + u_0 v v^T$ is still positive semidefinite then we are done. If not, we add $-\lambda_{\min}(\hat{Z} + u_0 v v^T)e$ to \hat{u} ,

$$u = \hat{u} - \lambda_{\min}(\hat{Z} + u_0 v v^T)e.$$

This worsens the original bound of $e^T \hat{u}$ by $-n\lambda_{\min}$ but the new Z is feasible again. Thus we are looking for an u_0 such that

$$4u_0 + e^T \hat{u} - n\lambda_{\min}(\hat{Z} + u_0 v v^T) < c^*.$$

We have more freedom to compensate the negative eigenvalue than we exploit by adding $-\lambda_{\min} \cdot I$. In particular the addition of $u_0 v v^T$ leads to negative eigenvectors with strong components in indices i and j . In practice it proved much better to work with $A_0 = v v^T - \text{Diag}(e_i + e_j)$, which is $v v^T$ with zeros on the diagonal. This

change can be compensated by adding $-u_0$ to \hat{u}_i and \hat{u}_j . Equivalently it can be modeled by a cost coefficient of 2 for u_0 . Observe that $\langle A_0, X \rangle = 2$ is a natural way to model the constraint $x_{ij} = -\text{sgn}(\hat{x}_{ij})$. The support of this representation is disjoint from the diagonal constraints.

Summing up we specialize the semidefinite program above to

$$\min_{u_0 \in \mathbb{R}} 2u_0 + e^T \hat{u} - n\lambda_{\min}(u_0 A_0 + \hat{Z}).$$

The minimal eigenvalue is a concave function, so the problem is convex. The function is differentiable if and only if the minimal eigenvalue has multiplicity one. In this case the gradient is determined by

$$\nabla_{u_0}(2u_0 - n\lambda_{\min}(u_0 A_0 + \hat{Z})) = 2 - n q(u_0)^T A_0 q(u_0)$$

with $q(u_0)$ denoting the (normalized) eigenvector to the minimal eigenvalue of $u_0 A_0 + \hat{Z}$. As explained above, it can be expected that \hat{Z} has eigenvalue zero with high multiplicity. Therefore the function is not differentiable for $u_0 = 0$. This complicates the recognition of good candidates i and j . It seems appropriate to choose the starting value u_0 with respect to the gap $c^* - e^T \hat{u}$, e.g. $u_0 = 1.2(c^* - e^T \hat{u})$. For reasonably large $|u_0|$ the minimal eigenvalue will be well separated and we can use the gradient to decide whether it is worth to increase $|u_0|$ even further or not. If because of the gradient it seems possible to beat c^* we do another step slightly overestimating the remaining gap. We repeat this procedure for at most three times.

The computation of the gradient requires the computation of the eigenvector to the minimal eigenvalue of $u_0 A_0 + \hat{Z}$. Extremal eigenvalues and eigenvectors are best determined via iterative methods such as the Lanczos method, which can exploit problem structure (see e.g. [11]). In particular these methods are very fast if a good starting vector is known. For the first computation we suggest the vector v , for all further iterations the last eigenvector computed is the natural choice. We expect that this method is efficiently applicable even in case approximate solutions of rather large sparse problems are given.

In Section 6 we will present some experimental results indicating the efficacy of this approach. Note, that the algorithm trivially extends to arbitrary matrices A_0 and other semidefinite relaxations exhibiting the possibility to shift eigenvalues directly.

5 Quadratic 0-1 Programming

It is well known that quadratic 0-1 programming in n variables is equivalent to quadratic $\{-1, 1\}$ programming in $n + 1$ variables [7] and this equivalence also extends to the canonical semidefinite relaxations [13, 20]. In general the $\{0, 1\}$ formulation is considered more intuitive and usually chosen for modelling combinatorial problems. In fact most articles dealing with constrained quadratic 0-1 programming work within this setting [22, 3, 13, 16]. From a theoretical point of view the equivalence of both primal problems is sufficient to observe that the previous considerations can be applied (indirectly) to the $\{0, 1\}$ setting. From a practical point of view two other aspects are important. Problem transformations tend to destroy structure inherent in the natural problem formulation. Therefore transformations should be avoided or they should be designed such as to preserve as much structure as possible. On the other hand dual variables usually have a natural interpretation in the original formulation. It is difficult to translate this interpretation into a transformed model and typically it is even more difficult to construct dual variables for the original problem from the dual variables of the transformed problem. Astonishingly, there is a transformation between $\{0, 1\}$ and $\{-1, 1\}$ formulation that

achieves both, problem structure is largely preserved and the dual variables are the same. In fact, it is based on the same transformation used in [7] and [13, 20].

Quadratic 0-1 programming in $\{0, 1\}$ variables asks for the optimal solution of

$$(QP) \quad \max y^T B y \quad \text{s.t. } y \in \{0, 1\}^n$$

The canonical semidefinite relaxation for quadratic 0-1 programming is derived by adding an additional component 1 (with index 0) to the vector y and by looking at

$$\begin{bmatrix} 1 \\ y \end{bmatrix} \begin{bmatrix} 1 & y^T \end{bmatrix} = \begin{bmatrix} 1 & y^T \\ y & yy^T \end{bmatrix}.$$

The latter matrix is positive semidefinite and its diagonal is equal to the first column and the first row for all $y \in \{0, 1\}^n$. An intuitive way to write the semidefinite relaxation is

$$(PQ) \quad \begin{array}{l} \max \quad \langle B, Y \rangle \\ \text{s.t.} \quad \bar{Y} = \begin{bmatrix} 1 & \text{diag}(Y)^T \\ \text{diag}(Y) & Y \end{bmatrix} \succeq 0. \end{array}$$

There are several possibilities to model linear constraints ensuring the diagonal property of \bar{Y} . We will construct a representation ensuring that the dual variables are the same as those of the equivalent problem (PMC) in $n + 1$ variables.

To this end we present some well known facts about transformations of the type $W = QXQ^T$ for nonsingular $Q \in M_n$ in the general setting of the primal dual pair (P) and (D). These transformations belong to the automorphism group of the semidefinite cone (the set of all bijective linear maps leaving the semidefinite cone invariant) and appear several times in the interior point literature in connection with scaling issues (see e.g. [25, 27]). Clearly, $W = QXQ^T$ is positive semidefinite if and only if X is. How do we have to change the constraints of (P) such that we get the same semidefinite program in terms of W ? Since $X = Q^{-1}WQ^{-T}$ and, for arbitrary $A \in S_n$,

$$\langle A, X \rangle = \langle A, Q^{-1}WQ^{-T} \rangle = \langle Q^{-T}AQ^{-1}, W \rangle$$

the correct transformation of a coefficient matrix A is $Q^{-T}AQ^{-1}$. Note, that this is the adjoint to the inverse transformation of QXQ^T . With

$$\bar{C} = Q^{-T}CQ^{-1}, \quad \bar{A}_i = Q^{-T}A_iQ^{-1} \quad i = 1, \dots, m,$$

and the linear operators $\bar{\mathcal{A}}$ and $\bar{\mathcal{A}}^T$ formed by the \bar{A}_i , we obtain the transformed primal dual pair

$$(P_Q) \quad \begin{array}{l} \min \quad \langle \bar{C}, W \rangle \\ \text{s.t.} \quad \bar{\mathcal{A}}(W) = b \\ W \succeq 0 \end{array} \quad (D_Q) \quad \begin{array}{l} \max \quad \langle b, \bar{u} \rangle \\ \text{s.t.} \quad \bar{\mathcal{A}}^T(\bar{u}) + \bar{Z} = \bar{C} \\ \bar{Z} \succeq 0. \end{array}$$

Proposition 5.1 *X is a feasible solution of (P) if and only if the associated $W = QXQ^T$ is a feasible solution of (P_Q) . Furthermore X and W satisfy $\langle C, X \rangle = \langle \bar{C}, W \rangle$. (u, Z) is a feasible solution of (D) if and only if the associated $(\bar{u}, \bar{Z}) = (u, Q^{-T}ZQ^{-1})$ is a feasible solution of (D_Q) . Trivially, $\langle b, u \rangle = \langle b, \bar{u} \rangle$.*

Proof. Clear by construction. ■

In particular this implies that given an optimal primal dual solution for one of the problems we can construct an optimal primal dual solution for the other.

We apply this approach to the transformation between $\{-1, 1\}$ and $\{0, 1\}$ representation of 0-1 quadratic programming

Proposition 5.2 *Let $Q \in M_{n+1}$ be the matrix*

$$Q = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}e & \frac{1}{2}I_n \end{bmatrix},$$

then $\varphi : S_{n+1} \rightarrow S_{n+1}, X \mapsto \bar{Y} = QXQ^T$ bijectively maps feasible solutions of (PMC) (for $n+1$ variables) to feasible solutions of (PQ).

Proof. Q is nonsingular, therefore X is positive definite if and only if $\varphi(X)$ is. The properties concerning the diagonals are verified by direct computation. \blacksquare

This is a slight simplification with respect to earlier proofs of this fact ([13, 20]). However, the advantage of this approach is that by Proposition 5.1 we know how to formulate the constraints such that we can go back and forth between both models without changing the dual variables.

To make this even simpler we provide a table of the most important transformations. In order to introduce the necessary notation observe that

$$Q^{-1} = \begin{bmatrix} 1 & 0 \\ -e & 2I_n \end{bmatrix}.$$

As the first row and column play a special role in (PQ) we give all transformations for partitioned matrices

$$\bar{A} = \begin{bmatrix} a_0 & a^T \\ a & A \end{bmatrix}$$

with $a_0 \in R$, $a \in \mathbb{R}^n$, and $A \in S_n$. The correct transformation of the coefficient matrices is achieved by the adjoint operator to φ^{-1} (with φ as in Proposition 5.2),

$$(\varphi^{-1})^* : S_{n+1} \rightarrow S_{n+1}, \bar{A} \mapsto \bar{B} = Q^{-T} \bar{A} Q^{-1}.$$

For implementational purposes constraint matrices of the form $\bar{A} = \bar{v}\bar{v}^T$ or $\bar{A} = \bar{v}\bar{v}^T + \bar{v}'\bar{v}'^T$ are of special importance [22, 13, 16] and conveniently transformed by the linear bijective map

$$\sigma : \mathbb{R}^{n+1} \mapsto \mathbb{R}^{n+1}, \bar{v} \mapsto \bar{w} = Q^{-T} \bar{v}.$$

Obviously,

$$(\varphi^{-1})^*(\bar{v}\bar{v}^T) = \sigma(\bar{v})\sigma(\bar{v})^T \quad \text{and} \quad (\varphi^{-1})^*(\bar{v}\bar{v}^T + \bar{v}'\bar{v}'^T) = \sigma(\bar{v})\sigma(\bar{v})^T + \sigma(\bar{v}')\sigma(\bar{v}')^T.$$

The explicit formulas are given in Table 1. Note, that these transformations preserve most of the structure (sparsity and low rank representations) which is of high practical importance.

To translate (PMC) to (PQ) we observe that $\text{diag}(\bar{X}) = \bar{e}$ can be modeled by

$$\bar{e}_i^T X \bar{e}_i = 1 \quad i = 0, \dots, n.$$

Using σ these translate to

$$\begin{aligned} \bar{e}_0 &\rightarrow \bar{w}^{(0)} = \bar{e}_0 \\ \bar{e}_i &\rightarrow \bar{w}^{(i)} \text{ with } \bar{w}_j^{(i)} = \begin{cases} -1 & j = 0 \\ 2 & j = i \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \dots, n. \end{aligned}$$

Collecting these $n+1$ constraints in an operator \mathcal{A} with $\bar{A}_i = \bar{w}^{(i)}\bar{w}^{(i)T}$ ($i = 0, \dots, n$) we obtain a formulation of (PQ) having the same dual variables as (PMC),

$$\begin{array}{ll} \max & \langle \bar{B}, \bar{Y} \rangle \\ \text{(PQ')} & \text{s.t. } \mathcal{A}(\bar{Y}) = e \\ & \bar{Y} \succeq 0. \end{array} \quad \begin{array}{ll} \min & e^T u \\ \text{(DQ')} & \text{s.t. } \bar{B} + S - \mathcal{A}^T(u) = 0 \\ & S \succeq 0. \end{array}$$

MC \rightarrow QP	QP \rightarrow MC
$\varphi(\bar{X}) = Q\bar{X}Q^T = \begin{bmatrix} y_0 & y^T \\ y & Y \end{bmatrix}$ $y_0 = x_0$ $y = \frac{1}{2}(x + x_0e)$ $Y = \frac{1}{4}(X + xe^T + ex^T + x_0ee^T)$	$\varphi^{-1}(\bar{Y}) = Q^{-1}\bar{Y}Q^{-T} = \begin{bmatrix} x_0 & x^T \\ x & X \end{bmatrix}$ $x_0 = y_0$ $x = 2y - y_0e$ $X = 4Y - 2y_0e^T - 2ey_0^T + y_0ee^T$
$(\varphi^{-1})^*(\bar{A}) = Q^{-T}\bar{A}Q^{-1} = \begin{bmatrix} b_0 & b^T \\ b & B \end{bmatrix}$ $b_0 = a_0 - 2e^T a + e^T Ae$ $b = 2(a - Ae)$ $B = 4A$	$\varphi^*(\bar{B}) = Q^T\bar{B}Q = \begin{bmatrix} a_0 & a^T \\ a & A \end{bmatrix}$ $a_0 = b_0 + e^T b + \frac{1}{4}e^T Be$ $a = \frac{1}{2}b + \frac{1}{4}Be$ $A = \frac{1}{4}B$
$\sigma(\bar{v}) = Q^{-T}\bar{v} = \begin{bmatrix} w_0 \\ w \end{bmatrix}$ $w_0 = v_0 - e^T v$ $w = 2v$	$\sigma^{-1}(\bar{w}) = Q^T\bar{w} = \begin{bmatrix} v_0 \\ v \end{bmatrix}$ $v_0 = w_0 + \frac{1}{2}e^T w$ $v = \frac{1}{2}w$
$Q = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}e & \frac{1}{2}I_n \end{bmatrix}$	$Q^{-1} = \begin{bmatrix} 1 & 0 \\ -e & 2I_n \end{bmatrix}$

Table 1: Transformations between the $\{-1, 1\}$ and the $\{0, 1\}$ model.

Returning to the fixing procedure for (PMC) we mention that in the $\{0, 1\}$ model it is not obvious how to guarantee the positive semidefiniteness of S in (DQ') by a similar approach. However, for given optimal solutions of the $\{0, 1\}$ model we can switch to the $\{-1, 1\}$ setting without changing the dual variables and compute appropriate Lagrange multipliers. These are also correct multipliers in the $\{0, 1\}$ model.

We now interpret the fixing of x_{ij} to either $+1$ or -1 in the $\{0, 1\}$ setting. Using σ again on v from (2) we obtain for $\bar{w} = \sigma(\bar{v})$

$$\begin{aligned}
i = 0, 1 \leq j \leq n, x_{ij} = 1 &\quad \rightarrow \quad w_0 = 2, w_j = -2 \\
i = 0, 1 \leq j \leq n, x_{ij} = -1 &\quad \rightarrow \quad w_0 = 0, w_j = 2 \\
1 \leq i < j \leq n, x_{ij} = 1 &\quad \rightarrow \quad w_0 = 0, w_i = 2, w_j = -2 \\
1 \leq i < j \leq n, x_{ij} = -1 &\quad \rightarrow \quad w_0 = -2, w_i = 2, w_j = 2
\end{aligned}$$

Reinterpreted in $\{0, 1\}$ variables y_i and y_j the equations $\bar{w}^T \bar{Y} \bar{w} = 0$ correspond to

$$\begin{aligned}
y_j &= 1 \\
y_j &= 0 \\
(y_i - y_j)^2 &= 0 \\
(y_i - y_j)^2 &= 1.
\end{aligned}$$

The third equation states that both, i and j , must be zero or both must be one. The fourth states that exactly one of both must be one. Using the analogous procedure as for (PMC) we can try to verify the validity of such an equation for the optimal solution of a particular problem by constructing the corresponding Lagrange multipliers with respect to an optimal solution (\bar{Y}^*, u^*, S^*) of the relaxation (PQ’).

For completeness we include the interpretation of the box constraints $0 \leq y_{ij} \leq 1$ arising naturally in linear relaxations. Observe that they do not appear in the list above. $y_{ij} \leq 1$ is guaranteed by the feasibility of \bar{Y} . The natural interpretation for $y_{ij} = 1$ is that both, i and j , must be one. In the $\{-1, 1\}$ setting this corresponds to requiring that indices 0, i , and j belong to the same set.

$y_{ij} \geq 0$ is not implied by the feasibility of \bar{Y} . In fact, this constraint is well known to correspond to a triangle inequality (in (1) take $v_0 = 1$, $v_i = 1$, and $v_j = 1$). The interpretation of $y_{ij} = 0$ is that at most one of i and j may attain the value 1. In the $\{-1, 1\}$ setting not all three, 0, i , and j , may belong to the same set.

6 Implementation

We have implemented the algorithm for fixing variables of Section 4 within our branch and cut code for solving (MC) as described in [14]. Here, we improve the semidefinite relaxation (PMC) with triangle inequalities only¹. Eigenvalues and eigenvectors are computed by the EISPACK routines `tred2` and `imtq12` (translated from Fortran to C with `f2c`). The fixing procedure is applied whenever a variable x_{ij} of the current optimal solution satisfies $|x_{ij}| > .98$. This leads to literally no additional cost for problems in which no variables satisfy this bound. Whenever variables of this size appeared then usually some of them could be fixed. We tested the code on the same classes of problems as in [14], $G_{.5}$, $G_{-1/0/1}$, Q_{100} , and $Q_{100,.2}$.

$G_{.5}$ consists of unweighted graphs with edge probability $1/2$, $G_{-1/0/1}$ of weighted (complete) graphs with edge weights chosen uniformly from $\{-1, 0, 1\}$. Q_{100} and $Q_{100,.2}$ were used in [29, 4]. Formulating Q_{100} with respect to (QP) the lower triangle of B is set to zero, the upper triangle (including the diagonal) is chosen uniformly from $\{-100, \dots, 100\}$. The diagonal takes the role of the linear term. $Q_{100,.2}$ represents instances with a density of 20%.

It was observed in [14] that in practice $G_{.5}$ and $G_{-1/0/1}$ are substantially more difficult to solve than Q_{100} and $Q_{100,.2}$. Indeed, for these classes the fixing routine was hardly ever called, because no variables satisfied $|x_{ij}| > .98$. Accordingly the additional cost of the routine was neglectable. However, for the “easy” classes of problems Q_{100} and $Q_{100,.2}$ the fixing routine was very successful and we present the results in Table 2.

Column n gives the dimension of the problem within the $\{-1, 1\}$ setting (the additional one is due to the transformation) and nr refers to the number of instances solved. The average computation time² and number of branch and bound nodes follow. Clearly, fixing variables leads to large savings in most cases. In fact, $n = 101$ of Q_{100} is the only case with substantial increase in computation time, even though the number of branch and bound nodes is significantly reduced. Analyzing this case closely we find that the fixing routine is called many times without success, each call corresponding to the computation of the full spectrum of a dense symmetric matrix of dimension 101. This seems too expensive in this case.

¹The reader familiar with [14] may note some changes in the results without fixing of variables with respect to the previously published results. These are due to a slower machine and some slight changes in the implementation.

²All times were computed on a Sun SPARCstation-4 with a 110 MHz microSPARC II CPU.

n	nr	no fixing		with fixing	
		h:mm:ss	nodes	h:mm:ss	nodes
Q_{100}					
41	10	00:01:42	23	00:01:08	10
51	10	00:05:24	39	00:03:01	14
61	5	00:10:40	51	00:06:12	16
71	3	00:26:52	71	00:20:01	35
81	2	00:49:33	86	00:45:26	45
91	1	01:17:44	79	00:23:21	3
101	1	02:00:31	137	02:11:00	104
$Q_{100,.2}$					
41	10	00:00:09	1	00:00:09	1
51	10	00:00:32	1	00:00:33	1
61	5	00:04:11	15	00:03:25	9
71	3	00:32:57	71	00:18:55	32
81	2	00:32:10	33	00:22:29	11
91	1	01:15:16	73	00:17:57	7
101	1	02:05:00	113	00:32:50	7

Table 2: Average branch and bound results

Summing up, the experimental results show that the fixing procedure is an important addition to branch and cut algorithms. Full implementations for larger problems will have to employ Lanczos methods for eigenvalue computations. It should be possible to narrow the number of candidates by analyzing the relation of the respective vectors v to the spectrum and the eigenvectors of Z^* , the latter would only have to be computed once. Finally it remains to investigate other branching schemes, e.g. branching with respect to triangle inequalities.

7 Conclusions

We propose to compute Lagrange multipliers for constraints which are not included in the problem description by means of a line search. The optimal step size can be computed explicitly for any given direction. An open problem in practical implementations is the (fast) determination of a good search direction. Applied to constraints of the form $v^T X v \geq 0$ this approach suggests the interpretation of the dual slack matrix Z as a variable subsuming all Lagrange multipliers corresponding to the active constraints $v^T X v \geq 0$ that ensure the positive semidefiniteness of X .

In the special case of the $\{-1, 1\}$ semidefinite relaxation of 0-1 quadratic programming the diagonal variables can be used to guarantee dual feasibility. This leads to an efficient and comparatively robust procedure for fixing variables which offers the possibility to exploit structure. In practice first implementations show considerable savings in computation time whenever candidates for fixing appear. Yet more sophisticated routines and more efficient implementations seem desirable and possible.

We extended the traditional equivalence transformation between $\{0, 1\}$ and $\{-1, 1\}$ representations of the semidefinite relaxation for 0-1 quadratic programming to the constraints such that the dual variables and the structural properties of the constraints are preserved. This transformation allows to apply the fixing procedure of the $\{-1, 1\}$ formulation to optimal solutions of the corresponding $\{0, 1\}$ relaxation, as well.

Although the most important ingredients for a general constrained quadratic 0-1

programming solver seem to be available by now, the solution of real world problems is still out of reach. The main obstacle is the high computational cost involved in solving the semidefinite relaxations by interior point methods because these cannot fully exploit problem structure. The fixing routine proposed here allows to exploit structural properties and depends solely on the availability of an approximate dual solution. Thus it may turn out to be a useful tool for any semidefinite programming solver to come.

I would like to thank Kurt Anstreicher for encouraging me to work on this topic and Stefan E. Karisch and Franz Rendl for pointing out some missing references and for their constructive criticism with respect to the presentation.

References

- [1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optimization*, 5(1):13–51, 1995.
- [2] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Coplementarity and non-degeneracy in semidefinite programming. Technical report, Computer Science Department, Courant Institute of Mathematical Sciences, New York University, New York, New York, NY, Mar. 1995. Submitted to Mathematical Programming.
- [3] 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.
- [4] F. Barahona, M. Jünger, and G. Reinelt. Experiments in quadratic 0-1 programming. *Mathematical Programming*, 44:127–137, 1989.
- [5] B. Chor and M. Sudan. A geometric approach to betweenness. In *ESA '95 Proceedings*, volume 979 of *Lecture Notes in Computer Science*, pages 227–237. Springer, 1995.
- [6] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62:557–547, 1993.
- [7] C. De Simone. The cut polytope and the boolean quadric polytope. *Discrete Applied Mathematics*, 79:71–75, 1989.
- [8] U. Feige and M. X. Goemans. Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT. In *Proceedings of the Third Israel Symposium on Theory of Computing and Systems*, pages 182–189, Tel Aviv, Israel, 1995.
- [9] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. In E. Balas and J. Clausen, editors, *Integer Programming and Combinatorial Optimization*, volume 920 of *Lecture Notes in Computer Science*, pages 1–13. Springer, May 1995.
- [10] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.
- [11] G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2nd edition, 1989.

- [12] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 2nd edition, 1988.
- [13] C. Helmberg, S. Poljak, F. Rendl, and H. Wolkowicz. Combining semidefinite and polyhedral relaxations for integer programs. In E. Balas and J. Clausen, editors, *Integer Programming and Combinatorial Optimization*, volume 920 of *Lecture Notes in Computer Science*, pages 124–134. Springer, May 1995.
- [14] C. Helmberg and F. Rendl. Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. ZIB Preprint SC-95-35, Konrad Zuse Zentrum für Informationstechnik Berlin, Takustraße 7, D-14195 Dahlem, Germany, Nov. 1995.
- [15] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior–point method for semidefinite programming. *SIAM J. Optimization*, 6(2):342–361, May 1996.
- [16] C. Helmberg, F. Rendl, and R. Weismantel. Quadratic Knapsack Relaxations Using Cutting Planes and Semidefinite Programming. In W. H. Cunningham, S. T. McCormick, and M. Queyranne, editors, *Integer Programming and Combinatorial Optimization*, volume 1084 of *Lecture Notes in Computer Science*, pages 175–189. Springer, June 1996.
- [17] F. Jarre. An interior–point method for minimizing the maximum eigenvalue of a linear combination of matrices. *Siam J. Control and Optimization*, 31(5):1360–1377, Sept. 1991.
- [18] S. E. Karisch and F. Rendl. Semidefinite programming and graph equipartition. Technical Report 302, Department of Mathematics, Graz University of Technology, Graz, Austria, Dec. 1995.
- [19] M. Kojima, S. Shindoh, and S. Hara. Interior–point methods for the monotone linear complementarity problem in symmetric matrices. Research Report B–282, Department of Information Sciences, Tokyo Institute of Technology, Apr. 1994. Revised April 1995.
- [20] M. Laurent, S. Poljak, and F. Rendl. Connections between semidefinite relaxations of the max-cut and stable set problems. CWI Report BS-R9502, CWI Amsterdam, The Netherlands, Jan. 1995.
- [21] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, IT-25(1):1–7, Jan. 1979.
- [22] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optimization*, 1(2):166–190, May 1991.
- [23] D. K. R. Motwani and M. Sudan. Approximate graph coloring by semidefinite programming. In *FOCS 94*, pages 2–13, 1994.
- [24] Y. Nesterov and A. Nemirovskii. *Interior–Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics, Philadelphia, 1993.
- [25] Y. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. Technical Report TR 1091, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853, Apr. 1994. Revised June 1995, to appear in *Mathematics of Operations Research*.

- [26] S. Poljak and F. Rendl. Nonpolyhedral relaxations of graph-bisection problems. *SIAM J. Optimization*, 5(3):467–487, 1995.
- [27] L. Tunçel. Primal-dual symmetry and scale invariance of interior-point algorithms for convex optimization. CORR Report 96–18, Univeristy of Waterloo, Ontario, Canada, Nov. 1996.
- [28] L. Vandenberghe and S. Boyd. A primal–dual potential reduction method for problems involving matrix inequalities. *Mathematical Programming, Series B*, 69(1):205–236, 1995.
- [29] A.C. Williams. *Quadratic 0-1 programming using the roof dual with computational results*. RUTCOR Research Report 8-85, Rutgers Unversity, 1985.
- [30] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. CORR Report, University of Waterloo, Ontario, Canada, Oct. 1996.
- [31] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. CORR Report 95/27, University of Waterloo, Ontario, Canada, Sept. 1996.