Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems

John E. Mitchell
Mathematical Sciences,
Rensselaer Polytechnic Institute,
Troy, NY 12180.

mitchj@rpi.edu
http://www.math.rpi.edu/~mitchj

June 25, 1999

Abstract

Many combinatorial optimization problems have relaxations that are semidefinite programming problems. In principle, the combinatorial optimization problem can then be solved by using a branch-and-cut procedure, where the problems to be solved at the nodes of the tree are semidefinite programs. It is desirable that the solution to one node of the tree should be exploited at the child node in order to speed up the solution of the child. We show how the solution to the parent relaxation can be used as a warm start to construct an appropriate initial dual solution to the child problem. This restart method for SDP branch-and-cut can be regarded as analogous to the use of the dual simplex method in the branch-and-cut method for mixed integer linear programming problems.

Keywords: Semidefinite programming, MAX-CUT problems, branch-and-cut.

1 Introduction

Many combinatorial and quadratic optimization problems can be written as:

$$\begin{array}{llllll} \min & v^TCv & & & & \\ \text{subject to} & v^TA_iv & \sharp & b_i, & i=1,\ldots,m & (COP) \\ & v_j & = & \pm 1, & j=1,\ldots,n, & \end{array}$$

where v is an n-vector, C and $A_i, i = 1,\ldots,m$ are symmetric $n \times n$ matrices, b is an m-vector, and $\sharp$ denotes either $=$ or $\geq$ or $\leq$, depending on the particular ith constraint. In many applications, C is a sparse matrix and each A_i is a rank one matrix that may also be sparse. For example, the maximum cut problem in a graph with weighted edges requires dividing the vertices of a graph into two sets so that the total weight of the edges having one endpoint in each set is as large as possible. This can be expressed in the form (COP) by letting C_{ij} be the negative of the edge weight between vertices i and j, with no constraints of the form $v^TA_iv \,\sharp\, b_i$ necessary. The equipartition problem requires that the vertices of a graph be divided into two sets of equal cardinality with as few edges as possible having one endpoint in each set. This can be formulated by taking C to be the matrix of edge weights and taking $A_1 = ee^T$, where e denotes a vector of ones of appropriate dimension, and $b_1 = 0$. No other constraints are needed. For other examples, see [1, 4, 10, 11, 19].

The standard semidefinite programming relaxation of (COP) is

$$\begin{array}{llcll} \min & C \bullet V & & & \\ \text{subject to} & V_{jj} & = & 1, & j = 1, \ldots, n, \\ & A_i \bullet V & \sharp & b_i, & i = 1, \ldots, m \\ & V & \succeq & 0. & \end{array} \qquad (SDP)$$

Here, $\bullet$ denotes the Frobenius inner product, so $C \bullet V := \sum_{i=1}^n \sum_{j=1}^n C_{ij} V_{ij}$. Further, $V \succeq 0$ denotes that the matrix V must be positive semidefinite. (SDP) is obtained from (COP) by noticing that the quadratic form $v^T C v$ can be written as $C \bullet V$ if we equate vv^T and V. It follows that the diagonal elements of V must equal one, and that V must be positive semidefinite. The condition that V is a rank one matrix is omitted, so (SDP) is a relaxation of (COP). This formulation has been used, for example, in [18, 8, 11, 4, 13].

With the addition of slack variables, if necessary, (SDP) can be written equivalently as

$$\begin{array}{llcllll} \min & C \bullet V & + & c^T x & & & \\ \text{subject to} & V_{jj} & & & = & 1, & j = 1, \ldots, n \\ & A_i \bullet V & + & a_i^T x & = & b_i, & i = 1, \ldots, m \\ & V \succeq 0, & & 0 \le x \le u, & & & \end{array} \qquad (SDPE)$$

where c, x, u and $a_i, i = 1, \ldots, m$ are l-vectors. Since $V \succeq 0$ and the diagonal elements of V are all equal to one, we have that $-1 \le V_{ij} \le 1$ for any element. Therefore, it is easy to calculate an upper bound u_i for any slack variable x_i. The dual problem to $(SDPE)$ is

$$\begin{array}{llclclclcl} \max & \text{trace}(W) & + & b^T y & & & - & u^T w & & \\ \text{subject to} & W & + & \sum_{i=1}^m y_i A_i & + & S & & & = & C \\ & & & A^T y & + & z & - & w & = & c \\ & S \succeq 0, & & z \ge 0, & & w \ge 0, & & & & \end{array} \qquad (SDPD)$$

where A is an $m \times l$ matrix whose ith row is a_i, W is a diagonal matrix, y is an m-vector, S is an $n \times n$ matrix, and z and w are n-vectors.

There are many interior point algorithms for solving semidefinite programming problems; see, for example, [1, 16, 12, 17, 14, 19]. Benson *et al.* [3, 4] have proposed using a dual-scaling algorithm to solve SDP relaxations of combinatorial optimization problems. This approach is able to exploit sparsity in the objective function matrix C to solve far larger instances than previously reported with semidefinite programming approaches. In particular, if C is sparse and A_i are sparse or of low rank, then S is either sparse or a low rank modification of a sparse matrix. Therefore, sparse matrix techniques can be used if we only work with the dual variables. Primal variables can be calculated as necessary.

A branch-and-cut approach is described in §2. In §3 we show that an analogue of the dual simplex method can be used to solve semidefinite programming problems when using a branch-and-cut approach, because a strictly feasible dual solution can be found easily. A method for finding a bound for the child problems is described in §4, and the determinants of the dual slacks of the child nodes are discussed in §5; the results of these two sections suggest that branching on elements of V' that are close to zero (the elements that are hardest to fix) has some useful properties. It may be that there is no strictly feasible primal point after branching; it is shown in §6 how to handle some forms of redundant constraints and how to modify the primal problem in order to ensure the existence of a strictly feasible primal solution.

2 A branch-and-cut approach

Helmberg and Rendl [11] investigated a branch-and-cut approach to solving problems of the form $(SDPE)$. They propose several classes of cutting planes, including triangle inequalities:

$$\begin{aligned} V_{ij} + V_{jk} + V_{ik} &\geq -1, \\ -V_{ij} - V_{jk} + V_{ik} &\geq -1. \end{aligned}$$

These inequalities are valid for any three distinct indices. They arise from exploitation of the fact that $V_{ij} = v_i v_j$ and the elements of v are all ± 1. The corresponding constraint matrix A_Δ can be written as a rank one matrix dd^T, where d only has nonzeroes in positions i, j, and k. After adding cutting planes, the relaxations still have the form $(SDPE)$. The cutting planes should always be added using symmetric matrices A_i.

Note that it is easy to find a new feasible iterate for the dual problem: we can set the new components of y to be zero, and set $z_i = w_i = 1$ for the new components of z and w. Helmberg and Rendl [11] propose taking $V = I$ as the new primal iterate for the MAX-CUT problem, since this is strictly feasible and is a convex combination of cuts.

The branching scheme proposed by Helmberg and Rendl [11] corresponds to branching on whether v_i and v_j should be the same or different. For the MAX-CUT problem, this corresponds to deciding whether vertices i and j should be on the same side of the cut or on opposite sides. With this branching rule, V_{ki} and V_{kj} are also then constrained to be either the same or different (depending on the branch) for each index k. This means that the problem can be replaced by an equivalent semidefinite program of dimension one less. Without loss of generality, let us assume that we are branching on whether $v_{n-1} = v_n$ or $v_{n-1} \neq v_n$. If the objective function matrix C in the SDP formulation is written

$$C = \begin{bmatrix} \bar{C} & p_1 & p_2 \\ p_1^T & \alpha & \beta \\ p_2^T & \beta & \gamma \end{bmatrix}, \tag{1}$$

where $\bar{C}$ is an $(n-2) \times (n-2)$ matrix, p_1 and p_2 are $(n-2)$-vectors, and α, β, and γ are scalars, then we obtain two different matrices depending on the branch. If $v_{n-1} = v_n$, we obtain:

$$C_S = \begin{bmatrix} \bar{C} & p_1 + p_2 \\ p_1^T + p_2^T & \alpha + 2\beta + \gamma \end{bmatrix}. \tag{2}$$

If $v_{n-1} \neq v_n$, we obtain:

$$C_O = \begin{bmatrix} \bar{C} & p_1 - p_2 \\ p_1^T - p_2^T & \alpha - 2\beta + \gamma \end{bmatrix}. \tag{3}$$

Similarly, if the matrices A_i are written

$$A_i = \begin{bmatrix} \bar{A}_i & q_{i1} & q_{i2} \\ q_{i1}^T & \alpha_i & \beta_i \\ q_{i2}^T & \beta_i & \gamma_i \end{bmatrix}, \tag{4}$$

where $\bar{A}_i$ is an $(n-2) \times (n-2)$ matrix, q_{i1} and q_{i2} are $(n-2)$-vectors, and α_i, β_i, and γ_i are scalars, then we obtain two different matrices depending on the branch. If $v_{n-1} = v_n$, we obtain:

$$A_{iS} = \begin{bmatrix} \bar{A}_i & q_{i1} + q_{i2} \\ q_{i1}^T + q_{i2}^T & \alpha_i + 2\beta_i + \gamma_i \end{bmatrix}. \tag{5}$$

If $v_{n-1} \neq v_n$, we obtain:

$$A_{iO} = \begin{bmatrix} \bar{A}_i & q_{i1} - q_{i2} \\ q_{i1}^T - q_{i2}^T & \alpha_i - 2\beta_i + \gamma_i \end{bmatrix}. \tag{6}$$

With the dual scaling algorithm of Benson *et al.* [4], it is useful to work with low rank matrices, if possible. The transformation defined above leaves a rank one matrix as a rank one matrix, as the following lemma shows.

Lemma 1 *If A_i is a rank one matrix then A_{iS} and A_{iO} are both rank one matrices.*

Proof: If A_i is a rank one matrix, we can write

$$A_i = \begin{bmatrix} v_A \\ \Gamma \\ \Phi \end{bmatrix} \begin{bmatrix} v_A^T & \Gamma & \Phi \end{bmatrix},$$

where v_A is an $(n-2)$-vector, and Γ and Φ are scalars. It is can then be verified that A_{iS} and A_{iO} are given by

$$A_{iS} = \begin{bmatrix} v_A \\ \Gamma + \Phi \end{bmatrix} \begin{bmatrix} v_A^T & \Gamma + \Phi \end{bmatrix}, \qquad A_{iO} = \begin{bmatrix} v_A \\ \Gamma - \Phi \end{bmatrix} \begin{bmatrix} v_A^T & \Gamma - \Phi \end{bmatrix},$$

as required. $\square$

For example, for the equipartition problem, on the $v_{n-1} = v_n$ branch, the constraint matrix $A_i = ee^T$ becomes

$$A_{iS} = \begin{bmatrix} 1 & \dots & 1 & 2 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \dots & 1 & 2 \\ 2 & \dots & 2 & 4 \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 & \dots & 1 & 2 \end{bmatrix},$$

a rank one matrix.

After branching, the relaxation takes one of the following two forms, where V is now an $(n-1) \times (n-1)$ symmetric matrix:

$$\begin{array}{llcccll} \min & C_S \bullet V & + & c^T x & & & \\ \text{subject to} & V_{jj} & & & = & 1, & j = 1, \dots, n-1, \\ & A_{iS} \bullet V & + & a_i^T x & = & b_i, & i = 1, \dots, m \\ & V \succeq 0, & & 0 \leq x \leq u, & & & \end{array} \qquad (SDPES)$$

or

$$\begin{array}{llcccll} \min & C_O \bullet V & + & c^T x & & & \\ \text{subject to} & V_{jj} & & & = & 1, & j = 1, \dots, n-1, \\ & A_{iO} \bullet V & + & a_i^T x & = & b_i, & i = 1, \dots, m \\ & V \succeq 0, & & 0 \leq x \leq u. & & & \end{array} \qquad (SDPEO)$$

Theorem 1 *(Helmberg and Rendl [11].) Let v be a feasible solution to (COP). Define $v^{n-1} := [v_1, \dots, v_{n-1}]^T$ and $V^{n-1} := v^{n-1}(v^{n-1})^T$. If $v_{n-1} = v_n$ then V^{n-1} is feasible in $(SDPES)$ for some choice of $x \geq 0$; otherwise, V^{n-1} is feasible in $(SDPEO)$ with some choice of $x \geq 0$.*

Proof: The proof follows directly from the construction of $(SDPES)$ and $(SDPEO)$ and the fact that $V = vv^T$ is feasible in $(SDPE)$. We can use the same vector x as in $(SDPE)$. $\square$

3 Finding a new strictly feasible dual solution

In a branch-and-cut method for mixed integer linear programming problems, the subproblems obtained after branching are solved using the dual simplex method, because fixing a primal variable at zero or one corresponds to dropping a dual constraint, so the old dual solution is still feasible. Branch-and-bound interior point methods have been investigated for mixed integer linear programming problems, with moderate success for some instances; see [5, 6, 15].

It is not quite so simple in the SDP case. However, we show in this section that it is still possible to construct a dual strictly feasible solution to the new subproblems using the old solution in a straightforward manner. A dual-scaling algorithm [3, 4] could then be used to solve the resulting subproblems.

At each iteration of an interior point method for solving $(SDPE)$, the matrix of dual slacks S is positive definite. At optimality, this matrix will be positive semidefinite. There are four possible outcomes at a node of the branch-and-cut tree:

- The optimal solution to the relaxation corresponds to a feasible solution to (COP). In this case, we can fathom the node through feasibility.
- The problem $(SDPE)$ is infeasible, so again we can fathom the node.
- The optimal solution to $(SDPE)$ has value worse than a known feasible solution to (COP), so we can fathom by bounds.
- The optimal solution to $(SDPE)$ has value better than the best known feasible solution to (COP) and it does not correspond to a feasible solution to (COP). In this case, it is necessary to branch.

Notice that in every case except the first, an approximate optimal solution to $(SDPE)$ will suffice, enabling us to determine the status of the node. In the first case, there is no need to branch. Thus, we may assume that when we branch we know a feasible dual solution where the matrix of dual slacks is positive definite. If, for numerical reasons, it is desired to use an earlier dual iterate, that is also possible with the restart procedure we describe below, again provided that the matrix of dual slacks is positive definite.

The dual problems to $(SDPES)$ and $(SDPEO)$ are

$$\begin{array}{llcccccccl} \max & \operatorname{trace}(W) & + & b^T y & & & - & u^T w & & \\ \text{subject to} & W & + & \sum_{i=1}^m y_i A_{iS} & + & S & & & = & C_S \\ & & & A^T y & + & z & - & w & = & c \\ & & & \multicolumn{5}{c}{S \succeq 0, \quad z \geq 0, \quad w \geq 0} & & \end{array} \qquad (SDPDS)$$

and

$$\begin{array}{llcccccccl} \max & \operatorname{trace}(W) & + & b^T y & & & - & u^T w & & \\ \text{subject to} & W & + & \sum_{i=1}^m y_i A_{iO} & + & S & & & = & C_O \\ & & & A^T y & + & z & - & w & = & c \\ & & & \multicolumn{5}{c}{S \succeq 0, \quad z \geq 0, \quad w \geq 0,} & & \end{array} \qquad (SDPDO)$$

respectively, where now W is a $(n-1) \times (n-1)$ diagonal matrix, and S is a $(n-1) \times (n-1)$ symmetric matrix.

Let (W', y', S', z', w') be the known feasible solution at the parent node, where S' is a positive definite matrix. We will restart with:

$$
\begin{aligned}
W_{ii} &= \begin{cases} {W'}_{ii} & i = 1, \ldots, n-2 \\ {W'}_{(n-1),(n-1)} + {W'}_{nn} & i = n-1 \end{cases} && (7) \\
y &= y' && (8) \\
z &= z' && (9) \\
w &= w', && (10)
\end{aligned}
$$

and S defined to be the resulting dual slacks.

We write S' as:

$$
S' = \begin{bmatrix} M & r_1 & r_2 \\ r_1^T & \tau & \nu \\ r_2^T & \nu & \xi \end{bmatrix}, \tag{11}
$$

where M is an $(n-2) \times (n-2)$ matrix, r_1 and r_2 are $(n-2)$-vectors, and τ, ν, and ξ are scalars. Since this is a positive definite matrix, it has a Cholesky factorization:

$$
S' = L'L'^T := \begin{bmatrix} L & 0 & 0 \\ v_1^T & \phi & 0 \\ v_2^T & \zeta & \chi \end{bmatrix} \begin{bmatrix} L^T & v_1 & v_2 \\ 0 & \phi & \zeta \\ 0 & 0 & \chi \end{bmatrix}, \tag{12}
$$

where L is a lower triangular matrix, v_1 and v_2 are $(n-2)$-vectors, and ϕ, ζ, and χ are real numbers. Notice that

$$
\begin{aligned}
Lv_1 &= r_1 && (13) \\
Lv_2 &= r_2 && (14) \\
v_1^T v_1 + \phi^2 &= \tau && (15) \\
v_2^T v_2 + \zeta^2 + \chi^2 &= \xi && (16) \\
v_1^T v_2 + \phi\zeta &= \nu. && (17)
\end{aligned}
$$

Let W^{n-2} denote the top left $(n-2) \times (n-2)$ submatrix of W. Consider first the case that we fix $v_{n-1} = v_n$. The matrix of dual slacks for the subproblem is then given by:

$$
\begin{aligned}
S_S &= C_S - W - \sum_{i=1}^{m} y_i A_{iS} \\
&= \begin{bmatrix} \bar{C} - W^{n-2} - \sum_{i=1}^{m} y_i \bar{A}_i & p_1 + p_2 - \sum_{i=1}^{m} y_i (q_{i1} + q_{i2}) \\ (p_1 + p_2)^T - \sum_{i=1}^{m} y_i (q_{i1} + q_{i2})^T & \rho \end{bmatrix} \\
&\qquad \text{where } \rho = \alpha + 2\beta + \gamma - {W'}_{(n-1),(n-1)} - {W'}_{nn} - \sum_{i=1}^{m} y_i (\alpha_i + 2\beta_i + \gamma_i), \\
&\qquad \text{using (2) and (5)} \\
&= \begin{bmatrix} M & r_1 + r_2 \\ (r_1 + r_2)^T & \tau + 2\nu + \xi \end{bmatrix}, && (18)
\end{aligned}
$$

from $(SDPD)$ and (11). Similarly, if we fix $v_{n-1} \neq v_n$, we obtain

$$S_O \quad = \quad \begin{bmatrix} M & r_1 - r_2 \\ (r_1 - r_2)^T & \tau - 2\nu + \xi \end{bmatrix}. \tag{19}$$

We can now show that the updated dual solution given in equations (7–10) is strictly dual feasible, so it gives an appropriate starting point for solving the child subproblems..

Theorem 2 *If S' is positive definite then both S_S and S_O are positive definite.*

Proof: The matrix S_S can be factored as

$$S_S = \begin{bmatrix} L & 0 \\ (v_1 + v_2)^T & \sigma_S \end{bmatrix} \begin{bmatrix} L^T & v_1 + v_2 \\ 0 & \sigma_S \end{bmatrix} \tag{20}$$

where

$$\sigma_S^2 := \tau + 2\nu + \xi - (v_1 + v_2)^T (v_1 + v_2),$$

using equations (12), (13), (14), and (18). If σ_S is a real number, then this is a Cholesky factorization, so S_S is indeed positive definite. We have from equations (15), (16), and (17):

$$\begin{aligned} \sigma_S^2 &= \tau + 2\nu + \xi - (v_1 + v_2)^T (v_1 + v_2) \\ &= \phi^2 + \zeta^2 + \chi^2 + 2\phi\zeta \\ &= (\phi + \zeta)^2 + \chi^2 \\ &> 0, \end{aligned}$$

since $\chi \neq 0$ because S' is positive definite. It follows that S_S is positive definite.

Similarly, the matrix S_O can be factored as

$$S_O = \begin{bmatrix} L & 0 \\ (v_1 - v_2)^T & \sigma_O \end{bmatrix} \begin{bmatrix} L^T & v_1 - v_2 \\ 0 & \sigma_O \end{bmatrix} \tag{21}$$

where

$$\begin{aligned} \sigma_O^2 &:= \tau - 2\nu + \xi - (v_1 - v_2)^T (v_1 - v_2) \\ &= (\phi - \zeta)^2 + \chi^2 \\ &> 0, \end{aligned}$$

using equations (12), (13–17), and (19) and the observation that $\chi \neq 0$. It follows that S_O is positive definite. □

4 A lower bound for the new dual

Goemans [7] proposed a branch-and-bound scheme where the initial relaxation is solved and this is used to obtain a bound for nodes lower in the tree. At level l of the tree, node $n - l + 1$ is placed on one side of the cut or the other. The bound obtained depends on the Cholesky factorization of S', the matrix of slacks at the root node. No iterations are taken at any node other than the root node. In this section, we construct a lower bound for the new dual problem in our framework using an argument similar to that in [7].

The value of any feasible solution for the new dual subproblem $(SDPDS)$ or $(SDPDO)$ provides a lower bound on the optimal value of the corresponding subproblem. This can be used to prune the subproblem if the bound is large enough. The construction of equations (7–10) gives an initial point with the same value as the last iterate from $(SDPD)$. Define Ξ to be this value.

Theorem 3 *A valid lower bound for* $(SDPDS)$ *is* $\Xi + (\phi + \zeta)^2 + \chi^2$, *and a valid lower bound for* $(SDPDO)$ *is* $\Xi + (\phi - \zeta)^2 + \chi^2$.

Proof: Adding $(\phi + \zeta)^2 + \chi^2$ or $(\phi - \zeta)^2 + \chi^2$ to $W_{(n-1),(n-1)}$, as appropriate, will increase the dual objective value to the stated quantities. This change still leads to positive semidefinite matrices S_S and S_O: setting $\sigma_S = 0$ in equation (20) and $\sigma_O = 0$ in equation (21) gives the corresponding slack matrix, and in each case it is clearly positive semidefinite. □

This theorem can be used to help determine an appropriate branching variable. It may well be useful to look for a pair of vertices where the increase in the lower bound is as large as possible. Of course, the examination of any pair of vertices requires a Cholesky factorization of the matrix, after reordering the columns, so this search would have to be performed in a careful manner.

The bounds given in Theorem 3 are exactly those obtained after fixing the last two variables in the tree defined in [7]. In general, if we modified our algorithm to take no iterations at subsequent nodes of the tree and if at level k we fix the relationship of vertices $n-k$ and $n-k+1$, then we would again obtain the same bounds as in [7].

5 The determinants of the child slack matrices

For this section, we assume that the primal and dual solutions for the parent problem are perfectly centered, namely,

$$V'S' = \mu I \tag{22}$$

for some positive scalar μ, where V' is the primal iterate for the parent problem. This is equivalent to

$$L'^T V' L' = \mu I, \tag{23}$$

where L' is defined in equation (12). Examining the (n,n) position of this equality shows that

$$\chi^2 = \mu, \tag{24}$$

since ${V'}_{ii} = 1$ for each i.

In general, if

$$V' = \left[\begin{array}{c|cc} \ddots & & \vdots \\ \hline & 1 & \epsilon \\ \cdots & \epsilon & 1 \end{array}\right] \tag{25}$$

and

$$L' = \left[\begin{array}{c|cc} \ddots & & 0 \\ \hline & \phi & 0 \\ \cdots & \zeta & \chi \end{array}\right], \tag{26}$$

we have the following:

Lemma 2 *If equation (22) holds then* $\phi\epsilon + \zeta = 0$ *and* $\phi^2 + 2\phi\epsilon\zeta + \zeta^2 = \phi^2 - \zeta^2 = \mu$.

Proof: The proof follows from examination of the $(n-1, n)$ and $(n-1, n-1)$ entries of equation (23). □

At the optimal solution to the underlying integer programming problem, V should be a rank one matrix. This suggests that it may be good to branch on columns of V that are close to orthogonal to one another. One situation in which the final two columns of V will not be parallel is if $\epsilon = 0$. In this case, we immediately obtain a corollary to Lemma 2.

Corollary 1 . *If* $V'_{(n-1),n} = 0$ *then* $\phi^2 = \mu$ *and* $\zeta = 0$.

The next corollary relates to large values of ϵ.

Corollary 2 *If (22) holds then* $-1 < \epsilon < 1$.

Proof: Since V is positive definite, we must have $-1 \leq \epsilon \leq 1$. Further, in Lemma 2, $\epsilon = \pm 1$ leads to a contradiction. □

Further, we can say something about the determinants of S_S and S_O. This value will have an impact on the speed of convergence of the interior point method for the child subproblems.

Theorem 4 *Assume equation (22) is satisfied. Let* $\epsilon = V'_{(n-1),n}$.

1. *If* $\epsilon = 0$ *then* $\det(S_S) = \det(S_O) = \frac{2}{\mu}\det(S')$.

2. *For general* ϵ,

$$\det(S_O) = \frac{(1+\epsilon)^2 + 1 - \epsilon^2}{\mu}\det(S') \quad \textit{and} \quad \det(S_S) = \frac{(1-\epsilon)^2 + 1 - \epsilon^2}{\mu}\det(S').$$

Proof: From equation (12), we have

$$\det(S') = \det(LL^T)\phi^2\chi^2.$$

From equations (20) and (21), we obtain

$$\det(S_S) = \det(LL^T)\sigma_S^2 \quad \text{and} \quad \det(S_O) = \det(LL^T)\sigma_O^2.$$

Lemma 2 implies

$$\phi^2 = \frac{\mu}{1-\epsilon^2}, \qquad \phi - \zeta = \phi(1+\epsilon), \qquad \text{and} \quad \phi + \zeta = \phi(1-\epsilon).$$

The result follows from the proof of Theorem 2 and then examining the ratios

$$\frac{(\phi-\zeta)^2 + \chi^2}{\phi^2\chi^2} \quad \text{and} \quad \frac{(\phi+\zeta)^2 + \chi^2}{\phi^2\chi^2}.$$

□

If $\epsilon \approx \pm 1$ then V' is close to singular, so S' has a large determinant. It follows from the theorem that one of the child problems will have a far smaller determinant.

Comparing the proof of this theorem with Theorem 3 shows that the bound on one of the child subproblems can be increased dramatically if $\epsilon \approx \pm 1$. Branching in this manner resembles a depth first search, with it likely that one of the branches will lead to a feasible solution quickly, and it may be possible to prune the other branch effectively. Branching on values of ϵ close to zero resembles a breadth first search approach, where both child subproblems are improved approximately equally.

The results in this section also suggest a way to fix or eliminate variables, analogous with reduced cost fixing in the integer linear programming case. Let Υ be the value of the best known feasible solution for (COP). If one of the bounds given in Theorem 3 is larger than Υ then the corresponding variable can be fixed to take the opposite value. Notice that at least one of the bounds will be larger than $\Xi + \phi^2$. If (22) holds then $\phi^2 = \frac{\mu}{1-\epsilon^2}$, so if ϵ is close enough to ± 1 then one of the bounds will be larger than Υ. If $\epsilon \approx 1$ then we may fix $v_{n-1} = v_n$; if $\epsilon \approx -1$ then we may fix $v_{n-1} = -v_n$. Of course, (22) will not hold exactly, but it should hold approximately. Therefore, for any entry V_{ij} that is close enough to ± 1, it may be worthwhile performing a Cholesky factorization of S', with the columns ordered so that these two columns are last, in order to confirm that one of the bounds is large enough to fix the variables. A different method for fixing variables is proposed in Helmberg [9], which also uses the matrix S' of dual slacks.

6 The child primal problem

Duality properties of semidefinite programming problems are not as strong as those for linear programming problems. For example, it is possible that the primal and dual problems are both feasible but there is a positive duality gap between the optimal primal and dual solutions. However, this cannot happen if there exists a strictly feasible dual solution, as stated in the following well-known theorem (see, for example, [1, 2, 16, 19]):

Theorem 5 *If there exists a strictly feasible solution to* $(SDPD)$ *and if the optimal value of* $(SDPD)$ *is bounded then there exists an optimal solution to* $(SDPE)$ *and the optimal values of* $(SDPD)$ *and* $(SDPE)$ *agree.*

Note that Theorem 5 does not state whether the optimal value to $(SDPD)$ is attained.

It is difficult to determine whether there exist strictly feasible solutions to $(SDPES)$ and $(SDPEO)$ in general, without trying to solve the problems. There are two different reasons why the child problems might not contain strictly feasible primal solutions:

- The relaxation of the underlying combinatorial optimization problem for this branch does not have an appropriate solution. We discuss one method for handling this case in §6.1
- The particular representation that we have for the problem does not allow strictly feasible solutions, because some additional variables have now become fixed or some constraints have become redundant. We discuss this case further in §6.2.

The dual scaling method does not require a new strictly feasible primal iterate in order to be restarted. All that is required is an upper bound $\bar{z}$ on the value of the current relaxation. Such an upper bound can be obtained from any point v that is feasible in (COP) and which satisfies the branching restrictions. Thus, heuristics can be used to construct an upper bound, if an appropriate point is not known already.

6.1 Ensuring the primal relaxation has an interior

It is possible that the child problem has no feasible interior solutions even if the parent problem has such solutions. For example, consider an equipartition problem with n vertices where there are two subsets of the vertices that have already been fixed, with the vertices in each subset required to appear on the same side of the partition as each other, and suppose we branch to require that the two subsets appear on the same side of the partition. Assume each subset contains less than half the vertices. If more than half of the vertices are in the union of the two subsets then the subproblem will be infeasible. If exactly half of the vertices appear in the two subsets then the subproblem will have exactly one feasible solution, namely, that every other vertex should appear on the opposite side from the two subsets.

We relate the feasible regions of (COP) and (SDP) in the following lemma.

Lemma 3 *If the set of feasible solutions to* (COP) *is full-dimensional then the set of feasible solutions to* (SDP) *contains a positive definite matrix.*

Proof: Let $\{v^1, \ldots, v^n\}$ be a set of linearly independent feasible solutions to (COP). Take $\hat{V} := [v^1, ..., v^n]$. Let $V = \frac{1}{n}\hat{V}\hat{V}^T$. It follows that $V_{jj} = 1$ for $j = 1, \ldots, n$. Consider the constraint $A_i \bullet V \sharp b_i$. We have

$$\begin{aligned} A_i \bullet V &= \mathrm{trace}(A_i V) \\ &= \frac{1}{n}\mathrm{trace}(A_i \hat{V}\hat{V}^T) \\ &= \frac{1}{n}\mathrm{trace}(\hat{V}^T A_i \hat{V}) \\ &= \frac{1}{n}\sum_{k=1}^{n}(v^k)^T A_i v^k. \end{aligned}$$

Thus, if each v^k satisfies the constraint $(v^k)^T A_i v^k \sharp b_i$ then V satisfies $V^T A_i V \sharp b_i$. By construction, V is positive semidefinite and, further, rank(V)=rank$(\hat{V}) = n$. The result follows. □

The following corollary is an immediate consequence of this lemma.

Corollary 3 *If the set of solutions to (COP) with the additional restriction ($v_{n-1} \pm v_n = 0$) still has dimension $n-1$ then there is a positive definite feasible primal matrix for the child problem.*

The results obtained in §3 still apply even if the problem has a less than full-dimensional feasible region to (COP).

If the feasible region to (COP) is not full dimensional, there must be valid equality constraints for the problem. These may be explicit or implicit. One technique used in [4] for modeling combinatorial optimization problems with equality constraints is to include an artificial variable with a corresponding large objective function coefficient. This technique is used when modeling the equipartition problem, among others. When branching, an artificial variable could be added to all constraints that involved $V_{j,(n-1)}$ and/or V_{jn} for some $j \in \{1, \ldots, n\}$, and this would ensure that the resulting child primal problem was full-dimensional if the parent problem was full-dimensional. Even if the child problem is infeasible, the problem with the artificial variable will still be feasible, albeit with a large optimal value and the artificial variable nonzero at optimality. It should not be necessary to solve such a subproblem to optimality within a branch-and-cut approach, as it is likely to be pruned before optimality is reached. By placing upper and lower bounds on the artificial variable, a strictly feasible interior point for the dual problem can be found easily.

6.2 Rank one inequalities

Consider the triangle inequality

$$V_{j,(n-1)} - V_{j,n} - V_{(n-1),n} \geq -1.$$

This constraint can be written

$$dd^T \bullet V \geq 1 \tag{27}$$

where d is an n-vector given by

$$d_j = 1, \qquad d_{n-1} = 1, \qquad d_n = -1, \qquad d_k = 0 \ \text{ otherwise.}$$

On the branch where we fix $v_{n-1} = v_n$, this constraint becomes $V_{jj} \geq 1$, which forces the corresponding primal slack variable to equal zero. Thus, the primal problem no longer has a strictly feasible point. One remedy is to drop this constraint, which requires dropping the corresponding dual variable y_i. The matrix of dual slacks can be left unchanged if W_{jj} is increased by y_i; note that this leaves the dual objective value unchanged. The other constraint involving y_i is of the form $-y_i + z_i - w_i = c_i$, and this can also be dropped without affecting the dual problem, since z_i and w_i only appear in this constraint.

On the other branch, the triangle inequality becomes $V_{jj}+4V_{j,(n-1)}+4V_{(n-1),(n-1)} \geq 1$, or equivalently, $V_{j,(n-1)} \geq -1$. If $y_i \geq 0$ then this constraint can be dropped and the matrix of dual slacks will still be positive definite. If y_i is negative, it may be necessary to decrease W_{jj} and $W_{(n-1),(n-1)}$ in order to regain a positive definite matrix of dual slacks. The smallest total decrease in the components of W is to decrease W_{jj} by $3y_i$ and decrease $W_{(n-1),(n-1)}$ by $6y_i$. Note that the constraint $-y_i + z_i - w_i \leq 0$ can also be dropped; the upper bound u_i is 8 (=9-1), and $w_i \geq -y_i$. Thus, the dual objective function value is not decreased by these modifications.

Consider the general case when A_i is a rank one matrix, so $A_i = tt^T$ and $t^T = [\bar{t}^T, p_t, q_t]^T$, where $\bar{t}$ is an $(n-2)$-vector and p_t and q_t are scalars. If at least two components of $\bar{t}$ are nonzero or at most one of p_t and q_t is nonzero, then the modified constraint is unlikely to be redundant. If both p_t and q_t are nonzero and $\bar{t}$ is zero then the constraint can be deleted, with a possible adjustment in $W_{(n-1),(n-1)}$ and with the dropping of a constraint of the form $y_i + z_i - w_i = c_i$. If p_t and q_t are both nonzero and exactly one component of $\bar{t}$ is nonzero, then the constraint can be handled as indicated above for the triangle inequalities. It may be that the resulting constraint requires that $V_{i,(n-1)}$ take a particular value, either 1 or -1. In this case, we can fix this element of V by combining columns i and $n-1$, as described in §2.

7 Conclusions

Theorem 2 shows that an analogue of the dual simplex method can be used to solve semidefinite programming problems when using a branch-and-cut approach, because a strictly feasible dual solution can be found easily. Investigation of the properties of this dual solution suggest that it may be useful to branch on elements of V' that are close to zero, since the determinants of the new dual slack matrices are well behaved and the bound on both branches of the tree can be increased. The results in this paper should make it possible to exploit a warm start in an SDP branch-and-cut algorithm, especially one that uses a dual-scaling algorithm [3, 4].

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.

[2] A. Ben-Tal and A. Nemirovskii. Convex optimization in engineering: Modeling, analysis, algorithms. Technical report, Faculty of Industrial Engineering and Management, Technion — Israel Institute of Technology, Technion-City, Haifa 32000, Israel, 1997.

[3] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. Technical report, Department of Management Sciences, University of Iowa, Iowa City, Iowa 52242, September 1997.

[4] S. J. Benson, Y. Ye, and X. Zhang. Mixed linear and semidefinite programming for combinatorial and quadratic optimization. Technical report, Department of Management Sciences, University of Iowa, Iowa City, Iowa 52242, February 1998.

[5] B. Borchers and J. E. Mitchell. Using an interior point method in a branch and bound algorithm for integer programming. Technical Report 195, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, March 1991. Revised July 7, 1992.

[6] A. de Silva and D. Abramson. A parallel interior point method and its application to facility location problems. *Computational Optimization and Applications*, 9:249–273, 1998.

[7] M. X. Goemans. Semidefinite programming in combinatorial optimization. *Mathematical Programming*, 79:143–161, 1997.

[8] Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[9] C. Helmberg. Fixing variables in semidefinite relaxations. Technical Report SC-96-43, Konrad-Zuse-Zentrum fuer Informationstechnik, Berlin, December 1996.

[10] 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, Lecture Notes in Computer Science*, volume 920, pages 124–134. Springer, 1995.

[11] C. Helmberg and F. Rendl. Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82:291–315, 1998.

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

[13] S. Joy, J. E. Mitchell, and B. Borchers. Solving MAX-SAT and weighted MAX-SAT problems using branch-and-cut. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, 1998.

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

[15] E. K. Lee and J. E. Mitchell. Computational experience of an interior point SQP algorithm in a parallel branch-and-bound framework. Technical Report LEC 97–08, Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, September 1997. Revised: August, 1998. To appear in *Proceedings of HPOPT97*, Rotterdam, The Netherlands.

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

[17] Y. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22:1–42, 1997.

[18] S. Poljak and F. Rendl. Nonpolyhedral relaxations of graph-bisection problems. *SIAM Journal on Optimization*, 5:467–487, 1995.

[19] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.