

Concave Quadratic Cuts for Mixed-Integer Quadratic Problems

Jaehyun Park Stephen Boyd

October 20, 2015

Abstract

The technique of semidefinite programming (SDP) relaxation can be used to obtain a nontrivial bound on the optimal value of a nonconvex quadratically constrained quadratic program (QCQP). We explore concave quadratic inequalities that hold for any vector in the integer lattice \mathbf{Z}^n , and show that adding these inequalities to a mixed-integer nonconvex QCQP can improve the SDP-based bound on the optimal value. This scheme is tested using several numerical problem instances of the max-cut problem and the integer least squares problem.

1 Introduction

We consider mixed-integer indefinite quadratic optimization problems of the form

$$\begin{aligned} & \text{minimize} && f_0(x) = x^T P_0 x + q_0^T x + r_0 \\ & \text{subject to} && f_i(x) = x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m \\ & && x \in \mathcal{S}, \end{aligned} \tag{1}$$

with variable $x \in \mathbf{R}^n$, where $\mathcal{S} = \{x \mid x_1, \dots, x_p \in \mathbf{Z}\}$ is the *mixed-integer set*, *i.e.*, the set of real-valued vectors whose first p components are integer-valued. The problem data are $P_i \in \mathbf{S}^n$, $q_i \in \mathbf{R}^n$, and $r_i \in \mathbf{R}$. Here, \mathbf{S}^n denotes the set of $n \times n$ real-valued, symmetric, possibly indefinite, matrices. Quadratic equality constraints of the form $x^T F x + g^T x + h = 0$ can also be handled by expressing them as two inequalities,

$$x^T F x + g^T x + h \leq 0, \quad -x^T F x - g^T x - h \leq 0.$$

The class of problems that can be written in the form of (1) is very broad; it includes other problem classes such as mixed-integer linear programs (MILPs) and mixed-integer quadratic programs (MIQPs). Discrete constraints such as Boolean variables can be easily encoded as well, which makes many NP-hard combinatorial optimization problems special cases of (1). For example, $x_i^2 = 1$ encodes the Boolean constraint that x_i is either +1 or -1. The max-cut problem is a well-known NP-hard problem with Boolean constraints only, which can be formulated as the following:

$$\begin{aligned} & \text{maximize} && -(1/4)x^T W x + (1/4)\mathbf{1}^T W \mathbf{1} = (1/2) \sum_{i < j} W_{ij}(1 - x_i x_j) \\ & \text{subject to} && x_i^2 = 1, \quad i = 1, \dots, n. \end{aligned} \tag{2}$$

Here, $\mathbf{1}$ represents a vector with all components equal to one. The matrix $W \in \mathbf{S}^n$ stores the weights of the edges in the graph: W_{ij} is the weight of the edge between node i and j . Other combinatorial problems in the form of (1) include the maximum clique problem, graph bisection problem, and satisfiability problem (SAT). In fact, any problem that is known to be an instance of the quadratic assignment problem (QAP) fits into our framework [KB57].

Since (1) is an integer problem at its root, classical number theoretic problems such as linear and quadratic Diophantine equations are also special cases of mixed-integer indefinite QCQP [Nag51, §6]. The integer least squares problem is another simple example of an integer quadratic problem:

$$\begin{aligned} & \text{minimize} && \|Ax - b\|_2^2 \\ & \text{subject to} && x \in \mathbf{Z}^n, \end{aligned} \tag{3}$$

with variable x and data $A \in \mathbf{R}^{r \times n}$ and $b \in \mathbf{R}^r$. The integer least squares problem captures the essence of the phase ambiguity estimation problem arising in the global positioning systems (GPS) [HB98].

There are other interesting constraints that can be encoded in problems of the form (1). For example, the rank constraint $\mathbf{Rank}(X) \leq k$ can be handled by introducing auxiliary matrix variables U and V of appropriate dimensions, and adding a constraint $X = UV$. Constraints involving the Euclidean distance between two points are encoded naturally as well. The sphere packing problem [CS13] and its variants are examples of problems involving (nonconvex) distance constraints.

Generic methods such as branch-and-bound [LD60] or branch-and-cut [PR91] can be used to solve (1) globally, but they all have an exponential time complexity. A more practical approach is to find an approximate solution, or to compute lower and upper bounds on the optimal value. The focus of this paper is on attaining a lower bound on the optimal value of (1) by forming a *semidefinite relaxation* that is solvable in polynomial time. Semidefinite programming (SDP) is a generalization of linear programming to symmetric positive semidefinite matrices. The idea of semidefinite relaxation can be traced back to [Lov79]. Semidefinite relaxation is a powerful tool that has been used not just within the domain of combinatorial problems [LMS⁺10]; a notable example is sum-of-squares (SOS) optimization in control theory [Nes00, Par00]. (For a thorough discussion of semidefinite programming and applications, readers are directed to [VB96].) A well-known application of semidefinite relaxation is to the max-cut problem; Goemans and Williamson constructed a randomized algorithm for the max-cut problem that attains a data-independent approximation factor of 0.87856, using the solution of a semidefinite relaxation [GW95].

Our main idea resembles that of cutting-plane method [Kel60], and is closely related to hierarchies of linear and semidefinite programs suggested by Lovász and Schrijver [LS91], Sherali and Adams [SA90], Gomory [Gom58], Chvátal [Chv73], and Lasserre [Las01a]. However, to the best of our knowledge, tightening SDP relaxation using true indefinite quadratic inequalities outside the domain of Boolean problems is a novel approach.

2 Semidefinite relaxation

A nontrivial lower bound on the optimal value of a (possibly nonconvex) QCQP can be obtained by “lifting” it to a higher-dimensional space, and solving the resulting problem. Consider a QCQP:

$$\begin{aligned} & \text{minimize} && f_0(x) = x^T P_0 x + q_0^T x + r_0 \\ & \text{subject to} && f_i(x) = x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m, \end{aligned} \tag{4}$$

with variable $x \in \mathbf{R}^n$. Let f^* denote its optimal value (which can be $-\infty$). By introducing a new variable $X = xx^T$, we can reformulate (4) as:

$$\begin{aligned} & \text{minimize} && F_0(X, x) = \mathbf{Tr}(P_0 X) + q_0^T x + r_0 \\ & \text{subject to} && F_i(X, x) = \mathbf{Tr}(P_i X) + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m \\ & && X = xx^T, \end{aligned}$$

with variables $X \in \mathbf{S}^n$ and $x \in \mathbf{R}^n$.

Then, we relax the nonconvex constraint $X = xx^T$ into a convex constraint $X \succeq xx^T$ (where the operator \succeq is with respect to the positive semidefinite cone) and write it using a Schur complement to obtain a convex relaxation:

$$\begin{aligned} & \text{minimize} && F_0(X, x) \\ & \text{subject to} && F_i(X, x) \leq 0, \quad i = 1, \dots, m \\ & && \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0. \end{aligned} \tag{5}$$

This is now a convex problem in the “lifted” space, in fact an SDP since the objective function is affine in X and x , and its optimal value is a lower bound on f^* .

This SDP can be solved using an interior point method in polynomial time, and in practice, the number of iterations required is constant and insensitive to the problem size, despite its worst-case complexity [VB96]. A detailed analysis of the running time is beyond the scope of this paper.

2.1 Lagrangian dual problem

Here, we derive the Lagrangian dual problem [BV04, §5] of the SDP (5). The Lagrangian of (5) is

$$L(X, x, \lambda, Y, y, \alpha) = F_0(X, x) + \sum_{i=1}^m \lambda_i F_i(X, x) - \mathbf{Tr} \begin{bmatrix} Y & y \\ y^T & \alpha \end{bmatrix} \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix}.$$

Let $g(\lambda, Y, y, \alpha)$ be the corresponding dual function, defined by

$$g(\lambda, Y, y, \alpha) = \inf_{X, x} L(X, x, \lambda, Y, y, \alpha).$$

The Lagrangian is affine in both X and x , and thus minimizing it over X and x gives $-\infty$, unless the coefficients of X and x are both zero. Therefore,

$$g(\lambda, Y, y, \alpha) = \begin{cases} r_0 + \sum_{i=1}^m \lambda_i r_i - \alpha & Y = P_0 + \sum_{i=1}^m \lambda_i P_i, \quad y = (1/2)(q_0 + \sum_{i=1}^m \lambda_i q_i) \\ -\infty & \text{otherwise.} \end{cases}$$

The dual problem is then

$$\begin{aligned} & \text{maximize} && r_0 + \sum_{i=1}^m \lambda_i r_i - \alpha \\ & \text{subject to} && Y = P_0 + \sum_{i=1}^m \lambda_i P_i \\ & && y = (1/2)(q_0 + \sum_{i=1}^m \lambda_i q_i) \\ & && \lambda_i \geq 0, \quad i = 1, \dots, m \\ & && \begin{bmatrix} Y & y \\ y^T & \alpha \end{bmatrix} \succeq 0, \end{aligned}$$

with variables $Y \in \mathbf{S}^n$, $y \in \mathbf{R}^n$, $\lambda \in \mathbf{R}^m$, $\alpha \in \mathbf{R}$. Alternatively, by eliminating Y and y , we get an equivalent formulation

$$\begin{aligned} & \text{maximize} && r_0 + \sum_{i=1}^m \lambda_i r_i - \alpha \\ & \text{subject to} && \lambda_i \geq 0, \quad i = 1, \dots, m \\ & && \begin{bmatrix} P_0 + \sum_{i=1}^m \lambda_i P_i & (1/2)(q_0 + \sum_{i=1}^m \lambda_i q_i) \\ (1/2)(q_0 + \sum_{i=1}^m \lambda_i q_i)^T & \alpha \end{bmatrix} \succeq 0, \end{aligned} \tag{6}$$

with variables $\lambda \in \mathbf{R}^m$, $\alpha \in \mathbf{R}$. Notice that the dual problem (6) is also a semidefinite program. Under mild assumptions (*e.g.*, feasibility of the primal problem), strong duality holds and both (5) and (6) yield the same optimal value.

An advantage of considering this dual problem is that unlike (5), any feasible point of (6) yields a lower bound on f^* . This observation could be particularly useful if a dual feasible solution with high objective value can be obtained relatively quickly.

3 Concave quadratic cuts for mixed-integer vectors

In this section, we present a simple method of relaxing the mixed-integrality constraint $x \in \mathcal{S}$ into a set of concave quadratic inequalities. Let $a \in \mathbf{Z}^n$ and $b \in \mathbf{Z}$ such that $a_{p+1} = \dots = a_n = 0$. The concave quadratic inequality

$$(a^T x - b)(a^T x - (b + 1)) \geq 0, \tag{7}$$

or equivalently,

$$-x^T (aa^T) x + (2b + 1)a^T x - b(b + 1) \leq 0,$$

holds if and only if $a^T x - b \leq 0$ or $a^T x - b \geq 1$. In particular, (7) holds for *every* vector $x \in \mathcal{S}$, since then $a^T x - b$ is integer-valued, which (trivially) satisfies $a^T x - b \leq 0$ or $a^T x - b \geq 1$. Figure 1 shows an example of an inequality of the form (7). It follows from this observation

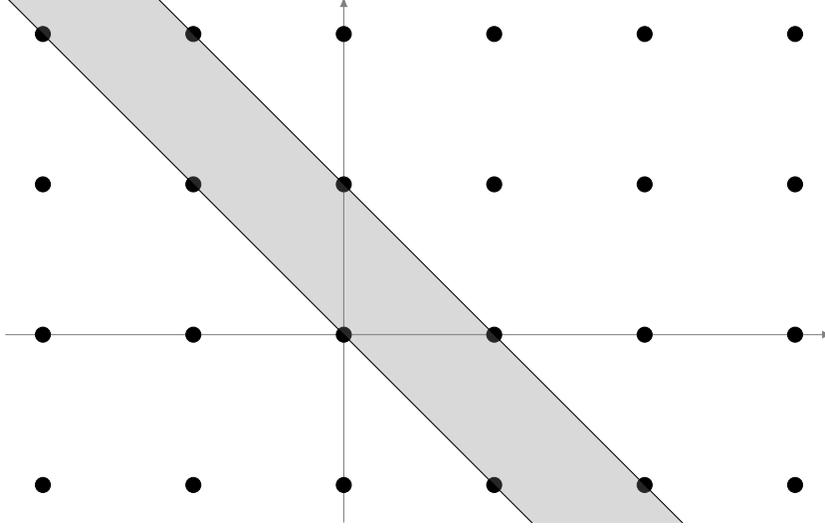


Figure 1: A concave quadratic inequality $(x_1 + x_2)(x_1 + x_2 - 1) \geq 0$. Notice that the shaded area, which does not satisfy the inequality, contains no lattice point.

that any number of such inequalities can be added to (1) without changing the optimal solution. That is, the following problem is equivalent to (1):

$$\begin{aligned}
& \text{minimize} && f_0(x) \\
& \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, m \\
& && -x^T(a_i a_i^T)x + (2b_i + 1)a_i^T x - b_i(b_i + 1) \leq 0, \quad i = 1, \dots, r \\
& && x \in \mathcal{S},
\end{aligned} \tag{8}$$

with $a_i \in \mathbf{Z}^n$, $b_i \in \mathbf{Z}$, $(a_i)_{p+1} = \dots = (a_i)_n = 0$ for $i = 1, \dots, r$. Then, simply dropping the mixed-integrality constraint from (8) gives a nonconvex QCQP over \mathbf{R}^n , and now the semidefinite relaxation technique in §2 is readily applicable. The resulting SDP is:

$$\begin{aligned}
& \text{minimize} && F_0(X, x) \\
& \text{subject to} && F_i(X, x) \leq 0, \quad i = 1, \dots, m \\
& && -\mathbf{Tr}(a_i a_i^T X) + (2b_i + 1)a_i^T x - b_i(b_i + 1) \leq 0, \quad i = 1, \dots, r \\
& && \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0,
\end{aligned} \tag{9}$$

where each F_i is defined as in §2. The optimal value f^{sdp} of (9) is a lower bound on f^* . Moreover, adding more true inequalities of the form (7) can only increase f^{sdp} , hence tightening the bound.

Inequalities of the form (7) are satisfied by every point in \mathcal{S} , but additionally, the mixed-integrality constraint $x \in \mathcal{S}$ itself can be written as a set of countably many such inequalities, *i.e.*, the set

$$\{x \mid (a^T x - b)(a^T x - (b + 1)) \geq 0 \text{ for all } a \in \mathbf{Z}^n, b \in \mathbf{Z}, a_{p+1} = \dots = a_n = 0\}$$

is precisely \mathcal{S} . Thus, adding concave quadratic cuts can also be interpreted as relaxing the mixed-integrality constraint, not by removing it completely, but by replacing it with infinitely many concave quadratic constraints, then dropping all but finitely many of them.

Although \mathcal{S} can be written as a set of countably many concave quadratic inequalities, when these inequalities are relaxed and rewritten in the lifted space as in (9), the resulting set of feasible points has no relationship with \mathcal{S} , in general. Consequently, the solution or the optimal value of (9) do not have any relationship with x^* or f^* . To demonstrate this, consider the following problem in \mathbf{Z}^2 :

$$\begin{aligned} & \text{minimize} && -\|x\|_2^2 \\ & \text{subject to} && \|x\|_2^2 \leq 1.2 \\ & && x \in \mathbf{Z}^2, \end{aligned} \tag{10}$$

which clearly has four optimal points $(\pm 1, 0)$ and $(0, \pm 1)$ with objective value $f^* = -1$. The SDP relaxation of this nonconvex problem (without any additional concave quadratic cut) is

$$\begin{aligned} & \text{minimize} && -\text{Tr } X \\ & \text{subject to} && \text{Tr } X \leq 1.2 \\ & && \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0, \end{aligned}$$

with optimal value $f^{\text{sdp}} = -1.2$. Take $\hat{X} = 0.6I$ and $\hat{x} = 0$, which attain this objective value. Then, for all $a \in \mathbf{Z}^n$ and $b \in \mathbf{Z}$,

$$-\text{Tr}(aa^T \hat{X}) + (2b + 1)a^T \hat{x} - b(b + 1) = -0.6\|a\|_2^2 - b(b + 1) \leq 0$$

holds. The inequality follows from the fact that b is integer-valued. This shows that adding any number of inequalities in the form of (7) to (10) will not increase f^{sdp} .

The example above shows that adding all possible concave quadratic cuts to an integer problem and subsequently solving the relaxation, in general, does not solve the original integer problem. In the special case of the integer least squares problem, however, it is not clear whether this relaxation is tight or not. When $n = 1$, *i.e.*, when the problem reduces to minimizing a convex quadratic function over the integers, it is easy to show that the relaxation is indeed tight. On the other hand, even when $n = 2$, we have failed to either prove the tightness of relaxation, or disprove it by producing any numerical instance whose relaxation is not tight.

Finally, we discuss simple extensions of concave quadratic cuts. The structure of the integer lattice, inherently, was not what made the construction of (7) possible. Rather, it was the property that every feasible point of (1) satisfies *exactly one* of the two affine inequalities (namely $a^T x - b \leq 0$ and $a^T x - (b + 1) \geq 0$). In other words, it does not matter whether the inequalities are on $a^T x - b$ or not; as long as there are two inequalities such that only one of them holds for every feasible point, they can be multiplied together to produce a true (and possibly nonconvex) inequality constraint. This construction resembles that of [Sho87, Ans09] in that affine constraints are combined to quadratic constraints, and subsequently

lifted to a higher-dimensional space. However, the key difference is that (7) encodes *exclusive disjunction*, *i.e.*, the two inequalities that we combine do not hold individually. There are other types of concave quadratic inequalities that utilize the structure of the integer lattice more. Take, for example, the constraint

$$\|x - (1/2)\mathbf{1}\|_2^2 \geq n/4.$$

This constraint holds for every $x \in \mathbf{Z}^n$, but is not representable as exclusive disjunction of any two affine inequalities. It is still a concave quadratic cut, and thus can be used in our framework without any modification. It is an open question as to whether these extensions result in a noticeable improvement of our SDP bound.

3.1 Choosing suitable cuts

Let (\hat{X}, \hat{x}) be a solution of (9), and consider adding an additional inequality

$$-\mathbf{Tr}(aa^T X) + (2b + 1)a^T x - b(b + 1) \leq 0 \quad (11)$$

to (9). Suppose that we want to choose integer-valued a and b so that adding (11) increases the SDP-based lower bound f^{sdp} . Then, we need (\hat{X}, \hat{x}) to violate this inequality, *i.e.*,

$$-\mathbf{Tr}(aa^T \hat{X}) + (2b + 1)a^T \hat{x} - b(b + 1) > 0. \quad (12)$$

Note that if a is given, then choosing b is easy, as we can maximize the lefthand side with respect to b , and round it to the nearest integer value. That is, we choose $b = \lfloor a^T \hat{x} \rfloor$.

Choosing a suitable vector a , unfortunately, is a difficult integer problem itself, and we need to use a heuristic to find a . One very simple method is to try all possible a with at most k nonzero elements that are either $+1$ or -1 . For a given such vector a , it takes $O(k^2)$ time to check whether (12) holds or not. Checking all possible such vectors then takes $O(n^k 2^k k^2)$ time. In particular, if $k = 2$, then checking all possible vectors can be done in $O(n^2)$ time.

A more involved heuristic uses an eigendecomposition of a certain matrix. To derive the heuristic, we first bound the lefthand side of (12) from below:

$$\begin{aligned} -\mathbf{Tr}(aa^T \hat{X}) + (2b + 1)a^T \hat{x} - b(b + 1) &\geq \inf_{b \in \mathbf{Z}} -\mathbf{Tr}(aa^T \hat{X}) + (2b + 1)a^T \hat{x} - b(b + 1) \\ &\geq \inf_{b \in \mathbf{R}} -\mathbf{Tr}(aa^T \hat{X}) + (2b + 1)a^T \hat{x} - b(b + 1) \\ &= -\mathbf{Tr}(aa^T \hat{X}) + 2(a^T \hat{x})^2 - ((a^T \hat{x})^2 - 1/4) \\ &= -\mathbf{Tr}(aa^T \hat{X}) + (a^T \hat{x})^2 + 1/4 \\ &= -a^T (\hat{X} - \hat{x} \hat{x}^T) a + 1/4. \end{aligned}$$

Our heuristic is to find $a \in \mathbf{Z}^n$ that maximizes the last line. Recall that $M = \hat{X} - \hat{x} \hat{x}^T$ is positive semidefinite, and thus $a = 0$ is a trivial maximizer of the expression. However, choosing $a = 0$ is not an option, since then (12) can never be satisfied, regardless of b . Therefore, instead of choosing $a = 0$, we choose an integer vector a that is “close” to the

eigenvector corresponding to the smallest eigenvalue v of M . At the same time, we want $\|a\|_2$ to be small (but nonzero), because scaling a by a factor of t also scales $a^T M a$ by a factor of t^2 . Note that once a is chosen this way, b is set as $\lfloor a^T \hat{x} \rfloor$ (which is the minimizer of the lefthand side of (12) over $b \in \mathbf{Z}$), instead of $a^T \hat{x} - 1/2$ (which is the minimizer of the lefthand side of (12) over $b \in \mathbf{R}$). Therefore, (12) may not hold even when $-a^T M a + 1/4 > 0$. Conversely, if $-a^T M a + 1/4 \leq 0$, then (12) is guaranteed not to hold. In particular, if λ_{\min} , the smallest eigenvalue of M , is larger than or equal to $1/4$, then there exists no inequality of the form (11) that can increase f^{sdp} . It is not hard to justify this:

$$a^T M a \geq \|a\|_2^2 \lambda_{\min} \geq \lambda_{\min}.$$

There are a number of reasonable ways to find a “short” integer vector a that is (approximately) aligned with a given vector v . For example, one can take an arbitrary scaling factor $t > 0$ and round each entry of tv to find a . Alternatively, we can fix some small k , take k indices i_1, \dots, i_k that correspond to the entries of v with the largest magnitudes, and set $a_{i_j} = \text{sign}(v_{i_j})$ for $j = 1, \dots, k$, while leaving the other entries of a as zeros. In particular, when $k = 1$, it means that a will be of the form $a = \pm e_i$ for some i . Without loss of generality, assume that $a = e_i$. According to this heuristic, we choose $b = \lfloor a^T \hat{x} \rfloor = \lfloor \hat{x}_i \rfloor$, and thus the inequality (11) we add in is

$$-X_{ii} + (2 \lfloor \hat{x}_i \rfloor + 1)x_i - \lfloor \hat{x}_i \rfloor (\lfloor \hat{x}_i \rfloor + 1) \leq 0.$$

The corresponding concave quadratic inequality of the form (7) is

$$(x_i - \lfloor \hat{x}_i \rfloor)(x_i - (\lfloor \hat{x}_i \rfloor + 1)) \geq 0.$$

Similarly, when $k = 2$, the corresponding inequalities would be on $x_i \pm x_j$ for some $i \neq j$.

Finally, we note that the heuristics described above can be applied in an iterative manner, as in the well-known cutting-plane method [Kel60]. That is, once some number of additional cuts are introduced, we can solve the resulting SDP and find more cuts using the new solution.

3.2 Application to branch-and-cut

In this section, we restrict ourselves to pure integer problems only, *i.e.*, $p = n$. Then, the concave quadratic cuts (7) can be used in a general branch-and-cut scheme to obtain the global solution. The branch-and-cut framework is depicted in Algorithm 3.1.

Algorithm 3.1 *Branch-and-cut algorithm.*

given an optimization problem \mathcal{P} of the form (1) with $p = n$.

1. *Initialize.* Add \mathcal{P} to \mathcal{T} , the list of *active problems*. Let $f^* := \infty$.

while \mathcal{T} is nonempty

 2. *Select an active problem.* Remove \mathcal{P}' from \mathcal{T} .

3. Solve the SDP relaxation of \mathcal{P}' to get its solution (\hat{X}, \hat{x}) with optimal value f^{sdp} .
 4. **if** $f^{\text{sdp}} \geq f^*$, go back to 2.
 5. **if** $\hat{x} \in \mathbf{Z}^n$, set $f^* := f^{\text{sdp}}$ and go back to 2.
 6. *Cut.* If any $a \in \mathbf{Z}^n$ and $b \in \mathbf{Z}$ satisfying (12) is found, add (11) to \mathcal{P}' and go back to 3.
 7. Create two problem instances \mathcal{P}_1 and \mathcal{P}_2 , both identical to \mathcal{P}' .
 8. Take some $c \in \mathbf{Z}^n$, $d \in \mathbf{Z}$, then add constraint $c^T x \leq d$ to \mathcal{P}_1 , and $c^T x \geq d + 1$ to \mathcal{P}_2 .
 9. *Branch.* Add \mathcal{P}_1 and \mathcal{P}_2 to \mathcal{T} .
-

There are a number of technical conditions that need to be met in order for Algorithm 3.1 to even terminate. We omit these details, as most problems of practical interest meet these conditions, such as a bounded domain, or the existence of the global solution (which would be implied by the former condition). For example, any 0-1 program satisfies these requirements.

The two crucial steps that affect the overall performance of the branch-and-cut algorithm are Steps 6 and 8. In §3.1, we have discussed a heuristic for finding a suitable cut that can be used in Step 6. Step 8, which is called the *branching step*, is another important step in the algorithm. Algorithm 3.1 may not even terminate if the branching step is not implemented carefully; for example, if the algorithm takes the same c and d at Step 7 every time, then it produces redundant branches, making no progress as a result. A commonly used branching strategy is to branch on a particular variable x_i . That is, for some index i such that \hat{x}_i is not integer-valued, we add $x_i \leq \lfloor \hat{x}_i \rfloor$ and $x_i \geq \lfloor \hat{x}_i \rfloor + 1$ as the branching inequalities. This would correspond to choosing $c = e_i$ and $d = \lfloor c^T \hat{x} \rfloor$ in Step 8 of Algorithm 3.1.

While branching on a single variable is an intuitive and simple strategy that is commonly used, we generalize this idea and find a “natural” branching inequality that can be easily obtained from the solution of the SDP relaxation (5), with almost no additional computation. The main idea comes from sensitivity analysis of convex problems; the optimal dual variables of (5) gives information about the sensitivity of f^{sdp} , with respect to perturbations of the corresponding constraints. Roughly speaking, if the magnitude of λ_i —the optimal dual variable corresponding to the constraint $F_i(X, x) \leq 0$ of (5)—is big, then the constraint is “tight,” and further tightening the constraint would lead to a big increase in f^{sdp} . To be precise, let $f^{\text{sdp}}(u)$ denote the optimal value of problem (5), when the inequality constraint $F_i(X, x) \leq 0$ is replaced with $F_i(X, x) \leq u$. Let λ_i be the optimal dual variable corresponding to the constraint of the unperturbed problem (9). Then, for all u , we have

$$f^{\text{sdp}}(u) \geq f^{\text{sdp}}(0) - \lambda_i u.$$

In other words, tightening an inequality (*i.e.*, $u < 0$) by $|u|$ increases the SDP bound by at least $\lambda_i |u|$. According to this interpretation, natural branching inequalities come from the concave quadratic cut (in the lifted space) with the largest dual variable. That is, if the constraint

$$-\text{Tr}(aa^T X) + (2b + 1)a^T x - b(b + 1) \leq 0 \tag{13}$$

has the largest value of the dual variable (*i.e.*, is the tightest), then we add $a^T x \leq b$ and $a^T x \geq b + 1$ as branching inequalities. The intuition behind this choice is that these inequalities

are tighter versions of (13); recall that (13) is a relaxation of (7), which is satisfied exactly when $a^T x \leq b$ or $a^T x \geq b + 1$. Therefore, we expect f^{sdp} to go up by adding any of these inequalities. After a branching inequality is added, (13) can be removed, as it is implied by the newly added branching inequality. When there is no concave quadratic cut with strictly positive dual variable, then we may branch on a single variable.

We note that scaling a constraint by a factor of $t > 0$ scales the corresponding optimal dual variable by a factor of $1/t$. Therefore, to correctly compare the dual variables and add branching inequalities, we have to apply a proper scaling to each constraint of the form (11). It is difficult to determine the scaling factor directly from perturbation and sensitivity analysis, as the branching inequalities we add, despite being tighter than (13), are not direct perturbations of it. However, experiments suggest that (11) is already properly scaled, and thus directly comparing the optimal dual variables gives good branching inequalities.

4 Examples

In this section, we consider numerical instances of the integer least squares problem and max-cut problem to show the effectiveness of the quadratic concave cuts in terms of the SDP-based lower bound. We emphasize that these problems were chosen because they are simple to describe, and showed qualitatively different results. They are by no means representative problems of the entire class of problems that can be written as (1).

4.1 Computational details

The SDP (5) was solved using CVX [GB14, GB08] with the MOSEK 7.1 solver [MOS], on a 3.40 GHz Intel Xeon machine. In order to obtain the solution in a reasonable amount of time, we only considered small-sized problems of $n \sim 100$.

4.2 Integer least squares

Problem formulation. We consider the following problem formulation that is equivalent to (3):

$$\begin{aligned} & \text{minimize} && \|A(x - x^{\text{cts}})\|_2^2 \\ & \text{subject to} && x \in \mathbf{Z}^n. \end{aligned} \tag{14}$$

Here, $x^{\text{cts}} \in \mathbf{R}^n$ is a given point at which the objective value becomes zero, which is a simple lower bound on the optimal value f^* .

Problem instances. We use random problem instances of the integer least squares problem (14), generated in the same way as in [PB15]: entries of $A \in \mathbf{R}^{r \times n}$ are sampled independently from $\mathcal{N}(0, 1)$, with the number of rows r set as $r = 2n$. The point x^{cts} was drawn from the uniform distribution on the box $[0, 1]^n$.

n	SDP1	SDP2	# of cuts
40	0.378	1.984	516.2
50	0.338	3.542	749.3
60	0.374	6.584	1033
70	0.443	11.82	1380
80	0.543	19.12	1732
100	0.830	47.51	2579

Table 1: Running time of SDPs by number of variables, and the average number of additional cuts added to the second SDP.

Method. We compare three lower bounds and one upper bound on the optimal value f^* . The first lower bound is the simple lower bound of the continuous relaxation: $f^{\text{cts}} = 0$. The second lower bound, which we denote by f_1^{sdp} , is the SDP-based lower bound explored in [PB15]. This was obtained by relaxing the integer constraint to a set of n quadratic concave inequalities $x_i(x_i - 1) \geq 0$ for all i , followed by solving the SDP relaxation. The third lower bound, f_2^{sdp} , was obtained by generalizing this approach, as described in §3; in addition to the inequalities $x_i(x_i - 1) \geq 0$, we considered $O(n^2)$ additional cuts with $\|a\|_2 = \sqrt{2}$, and added only those satisfied (12). Adding more inequalities (with $\|a\|_2 \geq \sqrt{3}$) gave little improvement, and thus they were not considered for our experiments. The upper bound, \hat{f} , was found by running the randomized algorithm constructed from the solution of the SDP (see [GW95, PB15]). In [PB15], it was shown empirically that this upper bound is very close to the optimal value for problems of small enough size ($n \leq 60$) that the optimal value was obtainable.

Results. For each problem size, we generated 100 problem instances and collected the lower and upper bounds from the SDP relaxation. In Table 1, we show the average running time of the two SDPs used to obtain f_1^{sdp} and f_2^{sdp} , respectively, and the average number of additional cuts added to the second SDP. The trade-off between the number of cuts (which was roughly $n^2/4$) and the running time of the SDP is clear from the table.

In Table 2, we compare the two lower bounds f_1^{sdp} and f_2^{sdp} , along with an upper bound \hat{f} . The simple lower bound $f^{\text{cts}} = 0$ was omitted from the table. Note that $f_2^{\text{sdp}} \geq f_1^{\text{sdp}}$ holds not only on average, but for every problem instance, because the second SDP is more constrained than the first. The ratio between the two optimality gaps, namely

$$\alpha = \frac{\hat{f} - f_2^{\text{sdp}}}{\hat{f} - f_1^{\text{sdp}}},$$

is also shown in the same table. We obtained a significant reduction for all problem sizes ($\alpha = 0.61$ for $n = 100$), though we expect α to be higher for larger problems. Note, however, that \hat{f} is not the optimal value, and thus the true reduction in the optimality gap is larger than what is reported. This is a notable improvement, because [PB15], to the best of our

n	f_1^{sdp}	f_2^{sdp}	\hat{f}	α
40	88.21	131.9	165.6	0.43
50	135.0	198.8	259.4	0.48
60	186.0	272.2	369.2	0.53
70	245.1	356.7	493.4	0.55
80	310.3	451.1	646.0	0.58
100	469.9	675.7	1001	0.61

Table 2: Average lower and upper bounds by number of variables, along with the average ratio between the optimality gaps.

knowledge, is a state of the art method for obtaining a lower bound on the integer least squares problem (in polynomial time).

4.3 Max-cut problem

Problem formulation. We reformulate (2) in terms of 0-1 variables $z_i = (1/2)(x_i + 1)$, so that the cuts introduced in §3 are tighter:

$$\begin{aligned} & \text{maximize} && (W\mathbf{1})^T z - z^T W z \\ & \text{subject to} && z_i(z_i - 1) = 0, \quad i = 1, \dots, n. \end{aligned} \tag{15}$$

Note that (15) is a maximization problem, and hence we get an upper bound on the optimal value by solving its relaxation.

Problem instances. We used the set of 10 small-sized problems ($n = 125$) with ± 1 edge weights, which was used in [FPRR02]. Note that due to the integral edge weights, the optimal value also is integer-valued. It follows that if f' is an upper bound on the optimal value f^* , then $\lfloor f' \rfloor$ is also an upper bound on f^* . In particular, if we have some feasible point z and some relaxation of the max-cut problem has an optimal value f^{sdp} such that $f_0(z) \leq f^{\text{sdp}} < f_0(z) + 1$, then the optimal solution to the SDP provides a *certificate of optimality* of the point z , *i.e.*, $f_0(z) = f^*$.

Method. In this application, we compare two upper bounds, and the best known lower bound to each of the 10 problem instances. The first upper bound f^{sw} is the classical SDP bound explored in [GW95]. This bound was obtained by relaxing (15) according to §2, without adding any concave quadratic cut. The second lower bound, which we denote by f^{sdp} , was found by adding concave quadratic cuts satisfying (12), just as in the integer least squares problem. However, for the max-cut problem, we found effectively no improvement in the SDP bound by adding inequalities with $\|a\|_2 = \sqrt{2}$, even when all such inequalities were added. Instead, we generated 10,000 random vectors a with $\|a\|_2 = \sqrt{3}$ (*i.e.*, having exactly three nonzero entries that are ± 1), and added the corresponding cuts to the SDP relaxation only when (12) was satisfied.

Instance	GW	SDP	# of cuts
G54100	1.92	17.9	1072
G54200	1.07	16.4	997
G54300	1.07	13.1	875
G54400	0.86	18.1	1117
G54500	0.86	16.4	987
G54600	1.17	14.1	915
G54700	0.82	20.1	1115
G54800	1.07	15.8	969
G54900	0.91	23.0	1325
G541000	0.89	22.9	1343

Table 3: Running time of two SDP relaxations, and the number of additional cuts added to the second SDP.

Results. In Table 3, we show the solve time of the two SDPs used to obtain f^{gw} and f^{sdp} , respectively, and the number of additional cuts added to the second SDP. As seen in the previous application, the trade-off between the number of cuts and the running time is clear.

In Table 4, we compare the two upper bounds f^{gw} and f^{sdp} , along with the best known lower bound \hat{f} . The upper bounds were rounded down to the nearest integer, using the observation made above. The ratio between the optimality gaps

$$\alpha = \frac{f^{\text{sdp}} - \hat{f}}{f^{\text{gw}} - \hat{f}},$$

is also shown in the same table. The value of α ranged from 0.73 to 0.88, which is larger than what we obtained in the case of integer least squares.

It should be noted that the objective of this experiment is not to compete with state of the art methods for solving the max-cut problem such as [MDL09] (which are shown to find near-optimal solutions very efficiently), but to demonstrate that our method generates a nontrivial upper bound for Boolean problems that is better than the Goemans-Williamson bound. Whether this method has any direct relationship with the (3rd) Lasserre hierarchy is an outstanding question.

References

- [ABSS93] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 724–733. IEEE Computer Society Press, 1993.

Instance	f^{gw}	f^{sdp}	\hat{f}	α
G54100	126	123	110	0.81
G54200	128	125	112	0.81
G54300	123	121	106	0.88
G54400	128	125	114	0.79
G54500	127	123	112	0.73
G54600	126	124	110	0.88
G54700	126	124	112	0.86
G54800	125	122	108	0.82
G54900	126	123	110	0.81
G541000	127	124	112	0.80

Table 4: Best known lower bound \hat{f} , Goemans-Williamson SDP bound f^{gw} , our SDP-based upper bound f^{sdp} , and the reduction in optimality gap for each problem instance.

- [AEVZ02] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, 2002.
- [Ajt96] M. Ajtai. Generating hard instances of lattice problems. In *Proceedings of the 28th Annual ACM Symposium on Theory of computing*, pages 99–108. ACM, 1996.
- [Ajt98] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proceedings of the 30th Annual ACM Symposium on Theory of computing*, pages 10–19. ACM, 1998.
- [AKS01] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33th Annual ACM symposium on Theory of computing*, pages 601–610. ACM, 2001.
- [Ans09] K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2-3):471–484, 2009.
- [AW02] M. F. Anjos and H. Wolkowicz. Strengthened semidefinite relaxations via a second lifting for the max-cut problem. *Discrete Applied Mathematics*, 119(1):79–106, 2002.
- [BCL12] C. Buchheim, A. Caprara, and A. Lodi. An effective branch-and-bound algorithm for convex quadratic integer programming. *Mathematical programming*, 135(1-2):369–395, 2012.

- [BDSPP13] C. Buchheim, M. De Santis, L. Palagi, and M. Piacentini. An exact algorithm for nonconvex quadratic integer minimization using ellipsoidal relaxations. *SIAM Journal on Optimization*, 23(3):1867–1889, 2013.
- [BDSRT15] C. Buchheim, M. De Santis, F. Rinaldi, and L. Trieu. A Frank-Wolfe based branch-and-bound algorithm for mixed-integer portfolio optimization. *arXiv preprint, arXiv:1507.05914*, 2015.
- [BEL12] A. Billionnet, S. Elloumi, and A. Lambert. Extending the QCR method to general mixed-integer programs. *Mathematical programming*, 131(1-2):381–401, 2012.
- [BEL13] A. Billionnet, S. Elloumi, and A. Lambert. An efficient compact quadratic convex reformulation for general integer quadratic programs. *Computational Optimization and Applications*, 54(1):141–162, 2013.
- [BEL15] A. Billionnet, S. Elloumi, and A. Lambert. Exact quadratic convex reformulations of mixed-integer quadratically constrained problems. *Mathematical Programming*, pages 1–32, 2015.
- [BHS15] C. Buchheim, R. Hubner, and A. Schobel. Ellipsoid bounds for convex quadratic integer programming. *SIAM Journal on Optimization*, 25(2):741–769, 2015.
- [Bie96] D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical programming*, 74(2):121–140, 1996.
- [Bie10] D. Bienstock. Eigenvalue techniques for convex objective, nonconvex optimization problems. In *Integer Programming and Combinatorial Optimization*, pages 29–42. Springer, 2010.
- [BL12] S. Burer and A. N. Letchford. Non-convex mixed-integer nonlinear programming: a survey. *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012.
- [BM94] B. Borchers and J. E. Mitchell. An improved branch and bound algorithm for mixed integer nonlinear programs. *Computers & Operations Research*, 21(4):359–367, 1994.
- [BMZ02] S. Burer, R. D. C. Monteiro, and Y. Zhang. Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, 12(2):503–521, 2002.
- [BPC⁺11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

- [Bre11] S. Breen. *Integer Least Squares Search and Reduction Strategies*. PhD thesis, McGill University, 2011.
- [BS12] S. Burer and A. Saxena. The MILP road to MIQCP. In *Mixed Integer Non-linear Programming*, pages 373–405. Springer, 2012.
- [BV97] S. Boyd and L. Vandenberghe. Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization. In *Communications, Computation, Control, and Signal Processing*, pages 279–287. Springer, 1997.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [BV08] S. Burer and D. Vandembussche. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Mathematical Programming*, 113(2):259–282, 2008.
- [BW05] D. Bertsimas and R. Weismantel. *Optimization over integers*, volume 13. Dynamic Ideas Belmont, 2005.
- [BW13] C. Buchheim and A. Wiegele. Semidefinite relaxations for non-convex quadratic mixed-integer programming. *Mathematical Programming*, 141(1-2):435–452, 2013.
- [CG09] X. W. Chang and G. H. Golub. Solving ellipsoid-constrained integer least squares problems. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1071–1089, 2009.
- [CH08] X. W. Chang and Q. Han. Solving box-constrained integer least squares problems. *IEEE Transactions on Wireless Communications*, 7(1):277–287, 2008.
- [Chv73] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete mathematics*, 4(4):305–337, 1973.
- [ÇI05] M. T. Çezik and G. Iyengar. Cuts for mixed 0-1 conic programming. *Mathematical Programming*, 104(1):179–202, 2005.
- [CS13] J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups*, volume 290. Springer Science & Business Media, 2013.
- [CYZ05] X. W. Chang, X. Yang, and T. Zhou. MLAMBDA: a modified LAMBDA method for integer least-squares estimation. *Journal of Geodesy*, 79(9):552–565, 2005.
- [CZ07] X. W. Chang and T. Zhou. MILES: MATLAB package for solving mixed integer least squares problems. *GPS Solutions*, 11(4):289–294, 2007.

- [DDA09] I. Dillig, T. Dillig, and A. Aiken. Cuts from proofs: A complete and practical technique for solving linear inequalities over integers. In *Computer Aided Verification*, pages 233–247. Springer, 2009.
- [DDA10] I. Dillig, T. Dillig, and A. Aiken. Small formulas for large programs: On-line constraint simplification in scalable static analysis. In *Static Analysis*, pages 236–252. Springer, 2010.
- [DDMA12] I. Dillig, T. Dillig, K. L. McMillan, and A. Aiken. Minimum satisfying assignments for SMT. In *Computer Aided Verification*, pages 394–409. Springer, 2012.
- [DG86] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36(3):307–339, 1986.
- [DKL11] E. De Klerk and M. Laurent. On the Lasserre hierarchy of semidefinite programming relaxations of convex polynomial optimization problems. *SIAM Journal on Optimization*, 21(3):824–832, 2011.
- [DKRS03] I. Dinur, G. Kindler, R. Raz, and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003.
- [DS90] C. De Simone. The cut polytope and the Boolean quadric polytope. *Discrete Mathematics*, 79(1):71–75, 1990.
- [DY15] C. Dang and Y. Ye. A fixed point iterative approach to integer programming and its distributed computation. *Fixed Point Theory and Applications*, 2015(1):1–15, 2015.
- [EB92] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.
- [Eck94] J. Eckstein. Parallel alternating direction multiplier decomposition of convex programs. *Journal of Optimization Theory and Applications*, 80(1):39–62, 1994.
- [EF98] J. Eckstein and M. C. Ferris. Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control. *INFORMS Journal on Computing*, 10(2):218–235, 1998.
- [FAEBAP15] Y. Fadlallah, A. Aïssa-El-Bey, K. Amis, and D. Pastor. Low-complexity detector for very large and massive MIMO transmission. In *IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 251–255. IEEE, 2015.

- [Fer00] E. Feron. Nonconvex quadratic programming, semidefinite relaxations and randomization algorithms in information and decision systems. In *System Theory*, pages 255–274. Springer, 2000.
- [FP85] U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of computation*, 44(170):463–471, 1985.
- [FPRR02] P. Festa, P. M. Pardalos, M. G. C. Resende, and C. C. Ribeiro. Randomized heuristics for the MAX-CUT problem. *Optimization methods and software*, 17(6):1033–1058, 2002.
- [GB08] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.
- [GB14] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.
- [GLS] M. Grötschel, L. Lovász, and A. Schrijver. Geometric algorithms and combinatorial optimization. 1988.
- [GMSS99] O. Goldreich, D. Micciancio, S. Safra, and J. P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.
- [Gom58] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958.
- [Gro02] I. E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3(3):227–252, 2002.
- [GT92] C. C. Gonzaga and M. J. Todd. An $O(\sqrt{n}L)$ -iteration large-step primal-dual affine algorithm for linear programming. *SIAM Journal on Optimization*, 2(3):349–359, 1992.
- [GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.
- [HB98] A. Hassibi and S. Boyd. Integer parameter estimation in linear models with applications to GPS. *IEEE Transactions on Signal Processing*, 46(11):2938–2952, 1998.

- [HR00] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.
- [HS11] R. Hübner and A. Schöbel. When is rounding allowed? A new approach to integer nonlinear optimization. Technical report, Institut für Numerische und Angewandte Mathematik, 2011.
- [JO04] J. Jaldén and B. Ottersten. An exponential lower bound on the expected complexity of sphere decoding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages iv–393. IEEE, 2004.
- [JO05] J. Jaldén and B. Ottersten. On the complexity of sphere decoding in digital communications. *IEEE Transactions on Signal Processing*, 53(4):1474–1484, 2005.
- [Kan83] R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the 15th Annual ACM Symposium on Theory of computing*, pages 193–206. ACM, 1983.
- [KB57] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, pages 53–76, 1957.
- [Kel60] J. E. J. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, pages 703–712, 1960.
- [Kho05] S. Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM (JACM)*, 52(5):789–808, 2005.
- [Knu97] D. E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, 1997.
- [Kon76] H. Konno. Maximization of a convex quadratic function under linear constraints. *Mathematical Programming*, 11(1):117–127, 1976.
- [Laa14] T. Laarhoven. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. Technical report, Cryptology ePrint Archive, Report 2014/744, 2014.
- [Las01a] J. B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *Integer Programming and Combinatorial Optimization*, pages 293–303. Springer, 2001.
- [Las01b] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

- [Lau03] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research*, 28(3):470–496, 2003.
- [LB15] R. Louca and E. Bitar. Acyclic semidefinite approximations of quadratically constrained quadratic programs. In *American Control Conference (ACC)*, pages 5925–5930. IEEE, 2015.
- [LD60] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520, 1960.
- [LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
- [LMS⁺10] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010.
- [LO⁺99] C. Lemaréchal, F. Oustry, et al. Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization. *Rapport de recherche 3710*, 1999.
- [Lov79] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(1):1–7, 1979.
- [LRMM13] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. 2013.
- [LS91] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.
- [LS06] D. Li and X. Sun. *Nonlinear integer programming*, volume 84. Springer Science & Business Media, 2006.
- [LW66] E. L. Lawler and D. E. Wood. Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719, 1966.
- [LY08] D. G. Luenberger and Y. Ye. *Linear and nonlinear programming*, volume 116. Springer Science & Business Media, 2008.
- [MDL09] R. Martí, A. Duarte, and M. Laguna. Advanced scatter search for the max-cut problem. *INFORMS Journal on Computing*, 21(1):26–38, 2009.

- [MDW⁺02] W. K. Ma, T. N. Davidson, K. M. Wong, Z. Q. Luo, and P. C. Ching. Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA. *IEEE Transactions on Signal Processing*, 50(4):912–922, 2002.
- [MF13] R. Misener and C. A. Floudas. GloMIQO: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, 57(1):3–50, 2013.
- [Mit02] J. E. Mitchell. Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of Applied Optimization*, pages 65–77, 2002.
- [MJHG06] U. Malik, I. M. Jaimoukha, G. D. Halikias, and S. K. Gungah. On the gap between the quadratic integer programming problem and its semidefinite relaxation. *Mathematical programming*, 107(3):505–515, 2006.
- [MOS] MOSEK ApS. The MOSEK optimization toolbox for MATLAB manual, version 7.1 (revision 28). <http://docs.mosek.com/7.1/toolbox.pdf>.
- [MY80] R. D. McBride and J. S. Yormark. An implicit enumeration algorithm for quadratic integer programming. *Management Science*, 26(3):282–296, 1980.
- [Nag51] T. Nagell. *Introduction to number theory*. American Mathematical Society, 1951.
- [Nes98] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization methods and software*, 9(1-3):141–160, 1998.
- [Nes00] Y. Nesterov. Squared functional systems and optimization problems. In *High performance optimization*, pages 405–440. Springer, 2000.
- [NNY94] Y. Nesterov, A. Nemirovskii, and Y. Ye. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994.
- [NS01] P. Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Cryptography and lattices*, pages 146–180. Springer, 2001.
- [Pad89] M. Padberg. The boolean quadric polytope: some characteristics, facets and relatives. *Mathematical programming*, 45(1-3):139–172, 1989.
- [Par00] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.
- [PB15] J. Park and S. Boyd. A semidefinite programming method for integer convex quadratic minimization. *arXiv preprint, arXiv:1504.07672*, 2015.

- [Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the 41st Annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2009.
- [PR91] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991.
- [PRW95] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for (0, 1)-quadratic programming. *Journal of Global Optimization*, 7(1):51–73, 1995.
- [SA90] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.
- [SBL10] A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Mathematical Programming*, 124(1-2):383–411, 2010.
- [Sch87] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical computer science*, 53(2):201–224, 1987.
- [SE94] C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical programming*, 66(1-3):181–199, 1994.
- [Sho87] N. Z. Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25(6):1–11, 1987.
- [SLW03] B. Steingrimsson, Z. Q. Luo, and K. M. Wong. Soft quasi-maximum-likelihood detection for multiple-antenna wireless channels. *IEEE Transactions on Signal Processing*, 51(11):2710–2719, 2003.
- [SS14] M. Soltanalian and P. Stoica. Designing unimodular codes via quadratic optimization. *IEEE Transactions on Signal Processing*, 62(5):1221–1234, 2014.
- [SVH05a] M. Stojnic, H. Vikalo, and B. Hassibi. A branch and bound approach to speed up the sphere decoder. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 429–432. IEEE, 2005.
- [SVH05b] M. Stojnic, H. Vikalo, and B. Hassibi. An H -infinity based lower bound to speed up the sphere decoder. In *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, pages 751–755. IEEE, 2005.

- [Teu99] P. J. G. Teunissen. An optimality property of the integer least-squares estimator. *Journal of Geodesy*, 73(11):587–593, 1999.
- [TMBB15] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad. A simple effective heuristic for embedded mixed-integer quadratic programming. *arXiv preprint, arXiv:1509.08416*, 2015.
- [UR15] B. Ustun and C. Rudin. Supersparse linear integer models for optimized medical scoring systems. *arXiv preprint, arXiv:1502.04269*, 2015.
- [VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.
- [VT98] N. Van Thoai. Global optimization techniques for solving the general quadratic integer programming problem. *Computational Optimization and Applications*, 10(2):149–163, 1998.
- [WES05] A. Wiesel, Y. C. Eldar, and S. Shamai. Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels. *IEEE Signal Processing Letters*, 12(9):653–656, 2005.
- [WN14] L. A. Wolsey and G. L. Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 2014.
- [WR14] F. Wang and C. Rudin. Falling rule lists. *arXiv preprint, arXiv:1411.5899*, 2014.
- [WRD⁺15] T. Wang, C. Rudin, F. Doshi, Y. Liu, E. Klampfl, and P. MacNeille. Bayesian Or’s of And’s for interpretable classification with application to context aware recommender systems, 2015.
- [ZH06] S. Zhang and Y. Huang. Complex quadratic optimization and semidefinite programming. *SIAM Journal on Optimization*, 16(3):871–890, 2006.
- [Zha00] S. Zhang. Quadratic maximization and semidefinite relaxation. *Mathematical Programming*, 87(3):453–465, 2000.
- [ZSL11] X. J. Zheng, X. L. Sun, and D. Li. Nonconvex quadratically constrained quadratic programming: best DC decompositions and their SDP representations. *Journal of Global Optimization*, 50(4):695–712, 2011.