Lagrangian Decomposition of Mixed-Integer All-Quadratic Programs*

Ivo Nowak

May 31, 2002

Humboldt-Universität zu Berlin
Institut für Mathematik
Rudower Chaussee 25,
D-12489 Berlin, Germany
E-mail : ivo@mathematik.hu-berlin.de

Abstract

The purpose of this paper is threefold. First we show that the Lagrangian dual of a block-separable mixed-integer all-quadratic program (MIQQP) can be formulated as an eigenvalue optimization problem keeping the block-separable structure. Second we propose splitting schemes for reformulating non-separable problems as block-separable problems. Finally we report numerical results on solving the eigenvalue optimization problem by a proximal bundle algorithm applying Lagrangian decomposition. The results indicate that appropriate block-separable reformulations of MIQQPs could accelerate the running time of dual solution algorithms considerably.

Key words. semidefinite programming, quadratic programming, combinatorial optimization, non-convex programming, decomposition

AMS classifications. 90C22, 90C20, 90C27, 90C26, 90C59

*The work was supported by the German Research Foundation (DFG) under grant NO 421/2-1.

1 Introduction

A mixed-integer all-quadratic program (MIQQP) is defined by

$$
\text{(Q)} \qquad \begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_E(x) = 0, \\ & f_I(x) \leq 0, \\ & x \in [\underline{x}, \overline{x}], \\ & x_B \text{ binary}, \end{array}
$$

where $|I| + |E| = m$, $f = (f_E, f_I) = (f_i)_{i=1,\ldots,m}$, $f_i(x) = \langle x, A^i x\rangle + 2\langle a^i, x\rangle + d_i$, $A^i \in I\!R^{(n,n)}$ is symmetric, $a^i \in I\!R^n$, $d_i \in I\!R$, $i = 0, \ldots, m$. Furthermore, $\underline{x}, \overline{x} \in I\!R^n$, $B \subset \{1, \ldots, n\}$ and x_B binary means that $x_j \in \{\underline{x}_j, \overline{x}_j\}$ for $j \in B$.

There exist a vast number of MIQQP applications. Many hard combinatorial optimization problems are special cases of MIQQP, such as max-cut, max-clique or quadratic assignment. Further applications are all bilinear problems, for example pooling problems in petrochemistry [VF90], modularization of product sub-assemblies [RS71] and special cases of structured stochastic games [FS99]. Other applications are packing problems [CS93], minmax location problems [hH82], chance-constrained problems in portfolio optimization [DT92, hH82, WV91], fuel mixture problem [PTA94], placement and layout problems in integrated circuit design [AKLvV95, AKvV96]. Several mixed-integer nonlinear programs (MINLP) can be reformulated as an MIQQP, for example polynomial programs [PS01]. Under mild assumptions it can be shown that every MINLP can be approximated by a MIQQP with arbitrary precision [Neu01].

Since MIQQP is a difficult NP-hard [GJ79] optimization problem many researchers worked on tractable relaxations of the problem. These relaxation can be either used to define lower bounds of a branch-and-bound procedure or to provide valuable information for generating good local solutions via a heuristic.

It is not easy to find a good relaxation of MIQQP which can be computed in reasonable time. One possibility is to approximate/reformulate a MIQQP by a mixed-integer linear program (MILP) [Neu01], which can be solved in quite high dimensions [BFG$^+$00]. However, the number of variables and constraints of a MILP approximation/reformulation is usually much higher than of the original problem. For example, the traditional MILP reformulation of an unconstrained quadratic binary program with n variables needs $n^2/2$ variables and $3n^2/2$ constraints.

Relaxations based on LP (linear programming) [BCC93, Ada99, Tho00, Kc00] are fast, but often too weak. Semidefinite programming (SDP) [WSV00] provides stronger relaxations. However, in large dimensions the cost for solving SDPs is often too large to be practicable. Attempts to reduce this cost are based on exploiting sparsity [FKN97, HR00] and second-order cone programming [KK00].

A conceptual different approach is Lagrangian decomposition (LD). Originally utilized by Danzig for exploiting block-angular structure of LPs [DW60] it is now a main tool for tackling difficult optimization problems which can be reformulated to be block-separable. LD is mainly used in mixed-integer linear programming. It has also been applied to quadratic 0-1 programming [CS95]. If the overhead of a block-separable reformulation is reasonable, LD often speeds up the method considerably. In addition LD offers the possibility of parallelization.

In this paper, we study the application of LD to a general MIQQP problem. We introduce three dualization schemes for MIQQP in Section 2. The last one is equivalent to an eigenvalue optimization problem. Based on strong duality results of the trust-region problem we show that all schemes have the same optimal value in Section 3. In Section 4, we discuss splitting schemes for reformulating MIQQPs to have a block-structure. We report preliminary numerical results on random MIQQPs in Section 5 and finish with some conclusions in Section 6.

1.1 Notation

Let $V = \{1, \ldots, n\}$ be a set of nodes, where a node i represents a variable x_i. We denote by $\mathcal{P}$ a partition of V, i.e. $\bigcup_{J\in\mathcal{P}} J = V$ and $J \cap K \neq \emptyset$ if $J \neq K$. We assign to each partition element $J \in \mathcal{P}$ a number $\pi_J \in \{1, \ldots, |\mathcal{P}|\}$ such that $\pi_J \neq \pi_K$ if $J \neq K$. The subvector $(x_i)_{i\in J}$ of a vector x is denoted by x_J. The submatrix $(a_{jk})_{j\in J,k\in K}$ of a symmetric matrix $A \in I\!R^{(n,n)}$ is denoted by A_{JK}. For $J \in \mathcal{P}$, we denote by $r^J = (r^J_1, r^J_2) \in I\!R^m \times I\!R^n$ the characteristic vector of a partition element J defined by $r^J_1 = 0$ and $r^J_{2j} = \begin{cases} 1 & \text{for } j \in J \\ 0 & \text{else} \end{cases}$. We denote by $\mathrm{Diag}(x) \in I\!R^{(n,n)}$ the diagonal matrix with the diagonal $x \in I\!R^n$. The optimal value of an optimization problem (P) is denoted by $\mathrm{val}(P)$. The zero centered ball in $I\!R^n$ with radius $n^{1/2}$ is denoted by $I\!B(n) = \{x \in I\!R^n \mid \|x\|^2 \le n\}$. The vector $e \in I\!R^n$ denotes the vector of ones. The minimum eigenvalue of a matrix A is denoted by $\lambda_1(A)$.

2 Three Lagrangian dualization schemes

Problem (Q) is called block-separable if there exist a partition $\mathcal{P}$ of $\{1, \ldots, n\}$, such that

$$f_i(x) = d_i + \sum_{J\in\mathcal{P}} f_i^J(x_J), \tag{1}$$

where $f_i^J(x_J) = \langle x_J, A^i_{JJ} x_J\rangle + 2\langle a^i_J, x_J\rangle$ for $i = 0, \ldots, m$. In other words, the matrices A^i are block-diagonal with $A^i_{JK} = 0$ for $J \neq K$, $J, K \in \mathcal{P}$ and $i = 1, \ldots, m$. We introduce now three Lagrangian dualization schemes for block-separable MIQQPs.

2.1 Dualization I

We consider the following all-quadratic programming formulation of (Q)

$$
\text{(Q1)} \qquad \begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_E(x) = 0, \\ & f_I(x) \le 0, \\ & q_B(x) = 0, \\ & q_C(x) \le 0, \end{array}
$$

with $C = \{1, \dots, n\} \setminus B$ and

$$
q_j(x) = (x_j - \underline{x}_j)(x_j - \overline{x}_j) = 0.
$$

Here, we replaced the box and binary constraints $x_j \in [\underline{x}_j, \overline{x}_j]$, $j \in C$, and $x_j \in \{\underline{x}_j, \overline{x}_j\}$, $j \in B$, respectively by the above quadratic constraints. By introducing the Lagrangian function

$$
L(x; \mu) = f_0(x) + \langle \mu_1, f(x) \rangle + \langle \mu_2, q(x) \rangle
$$

and the Lagrangian multiplier set

$$
\mathcal{M} = \{\mu = (\mu_1, \mu_2) \in I\!R^m \times I\!R^n \mid \mu_{1i} \ge 0 \text{ for } i \in I, \mu_{2j} \ge 0 \text{ for } j \in C\}
$$

we formulate the Lagrangian dual of (Q1) by

$$
\text{(D1)} \qquad \begin{array}{ll} \max & D_1(\mu) \\ \text{s.t.} & \mu \in \mathcal{M} \end{array}
$$

where $D_1(\mu) = \inf\limits_{x \in I\!R^n} L(x; \mu)$ is the dual function. Assuming that (Q) is block-separable, i.e. (1) holds, we define the partial Lagrangian function

$$
\begin{aligned}
L_1^J(x_J; \mu) &= f_0^J(x_J) + \langle \mu_1, f^J(x_J) \rangle + \langle \mu_{2,J}, q_J(x) \rangle \\
&= \langle x_J, A^J(\mu) x_J \rangle + 2\langle a^J(\mu), x_J \rangle + \langle \mu_{2,J}, q_J(x) \rangle
\end{aligned}
$$

with $A^J(\mu) = A^0_{JJ} + \sum_{i=1}^m \mu_{1i} A^i_{JJ}$ and $a^J(\mu) = a^0_J + \sum_{i=1}^m \mu_{1i} a^i_J$ and the related partial dual function

$$
D_1^J(\mu) = \min_{x \in I\!R^{|J|}} L_1^J(x; \mu).
$$

Then the dual function D_1 decomposes into

$$
D_1(\mu) = d(\mu) + \sum_{J \in \mathcal{P}} D_1^J(\mu),
$$

where $d(\mu) = d_0 + \sum_{i=1}^m \mu_{1i} d_i$.

Remark 1 *Since $D_1(\mu) > -\infty$ if and only if $\nabla^2 L(\cdot; \mu)$ is positive semidefinite, the dual (D1) contains a hidden semidefinite constraint. This implies that for all $\hat{\mu} \in$* dom D_1 *the function $L(\cdot; \hat{\mu})$ is a convex underestimator of f_0 over the feasible set of (Q).*

Remark 2 *Let $\beta(S)$ be the optimal value of the dual problem (D1) related to an interval $S = [\underline{x}, \overline{x}]$. Consider a sequence of nested intervals $\{S_k\}_{k \in I\!N}$ with $S_{k+1} \subset S_k$ converging to a point $\{\hat{x}\}$. In [Now00] it is shown*

$$\lim_{k \to \infty} \beta(S_k) = \begin{cases} f_0(\hat{x}) & \textit{if } \hat{x} \in \Omega \\ \infty & \textit{else,} \end{cases}$$

where Ω is the feasible set of (Q). This shows that $\beta(S)$ is a consistent lower bounding method ensuring convergence of branch-and-bound algorithms with exhaustive subdivision strategies [HPT95].

2.2 Dualization II (Standardization)

We standardize the variables to be bounded by -1 and 1. Let $u = \frac{1}{2}(\overline{x} + \underline{x})$ be the center and $w = \overline{x} - \underline{x}$ be the diameter vector of the interval $[\underline{x}, \overline{x}]$ respectively. The affine transformation $\theta(x) = \mathrm{Diag}(w)x + u$ maps the interval $[-e, e]$ onto $[\underline{x}, \overline{x}]$. Under this map, problem (Q1) is transformed into

$$\text{(Q2)} \qquad \begin{array}{lll} \min & d_i + \sum_{J \in \mathcal{P}} \hat{f}_0^J(x_J) & \\ \text{s.t.} & d_E + \sum_{J \in \mathcal{P}} \hat{f}_E^J(x_J) = 0, & \\ & d_I + \sum_{J \in \mathcal{P}} \hat{f}_I^J(x_J) \leq 0, & \\ & x_j^2 \leq 1, & j \in C \\ & x_j^2 = 1, & j \in B, \end{array}$$

where $\hat{f}_i^J(x_J) = \langle x_J, B_{JJ}^i x_J \rangle + 2\langle b_J^i, x_J \rangle + \hat{d}_i^J$, $B_{JJ}^i = W^J A_{JJ}^i W^J$, $b_J^i = W^J(a_J^i + A_{JJ}^i u_J)$, $\hat{d}_i^J = \langle u_J, A_{JJ}^i u_J \rangle + 2\langle a_J^i, u_J \rangle$ and $W^J = \mathrm{Diag}(w_J)$. The partial Lagrangian related to (Q2) is

$$\begin{aligned} L_2^J(x_J; \mu) &= L_1^J(W^J x_J + u_J; \mu) \\ &= \langle x, B^J(\mu) x \rangle + 2\langle b^J(\mu), x \rangle + \hat{d}^J(\mu), \end{aligned}$$

with $B^J(\mu) = B_{JJ}^0 + \sum_{i=1}^m \mu_{1i} B_{JJ}^i + \mathrm{Diag}(\mu_{2,J})$, $b^J(\mu) = b_J^0 + \sum_{i=1}^m \mu_{1i} b_J^i$ and $\hat{d}^J(\mu) = \hat{d}_0^J + \sum_{i=1}^m \mu_{1i} \hat{d}_i^J - \langle e, \mu_{2,J} \rangle$. Since the feasible set of (Q2) is contained in the interval $[-e, e]$, we can dualize (Q2) with respect to the set

$$X = \{x \in I\!R^n \mid x_J \in I\!B(|J|),\ J \in \mathcal{P}\}.$$

This defines the partial dual function

$$D_2^J(\mu) = \min_{x \in I\!B(|J|)} L_2^J(x; \mu), \tag{2}$$

the dual function

$$D_2(\mu) = d(\mu) + \sum_{J\in\mathcal{P}} D_2^J(\mu)$$

and the dual problem

$$\text{(D2)} \qquad \begin{array}{ll} \max & D_2(\mu) \\ \text{s.t.} & \mu \in \mathcal{M}. \end{array}$$

2.3 Dualization III (Homogenization)

We homogenize now (Q2) by replacing linear terms $\langle b_J^i, x_J\rangle$ by quadratic terms $x_{n+\pi_J}\cdot\langle b_J^i, x_J\rangle$ and adding constraints $x_{n+\pi_J}^2 - 1 = 0$. This gives the problem

$$\text{(Q3)} \qquad \begin{array}{lll} \min & d_0 + \sum_{J\in\mathcal{P}} \tilde f_0^J(x_{\tilde J}) & \\ \text{s.t.} & d_E + \sum_{J\in\mathcal{P}} \tilde f_E^J(x_{\tilde J}) = 0, & \\ & d_I + \sum_{J\in\mathcal{P}} \tilde f_I^J(x_{\tilde J}) \le 0, & \\ & x_j^2 - 1 \le 0, & j \in C \\ & x_j^2 - 1 = 0, & j \in B \cup \{n+1,\dots,n+|\mathcal{P}|\} \end{array}$$

where $\tilde f_i^J(x) = \langle x_{\tilde J}, C_{\tilde J\tilde J}^i x_{\tilde J}\rangle + \hat d_i^J$, $\tilde J = J\cup\{n+\pi_J\}$ and

$$C_{\tilde J\tilde J}^i = \begin{pmatrix} B_{JJ}^i & b_J^i \\ (b_J^i)^T & 0 \end{pmatrix},$$

$i = 1,\dots,m$. Obviously, $\tilde f^J(x) = f^J(\hat x)$ if $x_{1:|J|} = \hat x$ and $x_{|J|+1} = 1$ or $x_{1:|J|} = -\hat x$ and $x_{|J|+1} = -1$. Therefore, the optimal values of (Q3) and (Q2) coincide. Since each additional variable can be 1 or -1, the number of solutions of (Q3) is $2^{|\mathcal{P}|}$ times larger than of (Q2). The partial Lagrangian related to (Q3) is

$$\begin{aligned} L_3^J(x_J;\mu) &= \tilde f_0^J(x_{\tilde J}) + \langle \mu_1, \tilde f^J(x_{\tilde J})\rangle + \sum_{j\in\tilde J} \mu_{2j}\cdot(x_j^2 - 1) \\ &= \langle x, C^J(\mu)x\rangle + \tilde d^J(\mu), \end{aligned}$$

where

$$C^J(\mu) = \begin{pmatrix} B^J(\mu) & b^J(\mu) \\ b^J(\mu)^T & \mu_{2,n+\pi_J} \end{pmatrix}$$

and $\tilde d^J(\mu) = \hat d^J(\mu) - \mu_{2,n+\pi_J}$. The related partial dual function is the following eigenvalue function

$$D_3^J(\mu) = \min_{x\in \mathbb{B}(|J|+1)} L_3^J(x;\mu) = (|J|+1)\cdot\min\{0, \lambda_1(C^J(\mu))\} + \tilde d^J(\mu).$$

The dual function related to (Q3) is

$$D_3(\mu) = d(\mu) + \sum_{J\in\mathcal{P}} D_3^J(\mu)$$

defining the eigenvalue optimization problem

$$\text{(D3)} \qquad \begin{array}{ll} \max & D_3(\mu) \\ \text{s.t.} & \mu \in \tilde{\mathcal{M}} \end{array}$$

with

$$\tilde{\mathcal{M}} = \{\mu \in I\!R^m \times I\!R^{n+|\mathcal{P}|} \mid \mu_{1i} \geq 0 \text{ for } i \in I, \mu_{2j} \geq 0 \text{ for } j \in C\}.$$

A similar transformation was used in [RW97] for solving the trust region problem and in [Hel00] for unconstrained quadratic 0-1 programming.

2.4 Supergradient formula

The dual problem (D3) is a convex non-differentiable optimization problem and can be solved by many methods (see [HUL93]). For maximizing D_3 we need supergradients. We show that supergradients can be easily computed by solving eigenvalue problems. A supergradient of a concave function $D\colon I\!R^m \to I\!R$ is a point $g \in I\!R^m$ satisfying

$$D(\mu) + \langle g, \lambda - \mu\rangle \geq D(\lambda)$$

for all $\lambda, \mu \in I\!R^m$. A supergradient of a dual function can be obtained by evaluating constraint functions at a Lagrangian solution point. More precisely, it holds [HUL93]

Lemma 1 *Let $L(x;\mu) = f(x) + \langle h, \mu\rangle$ be a continuous Lagrangian function related to an objective function $f\colon I\!R^n \to I\!R$ and a constraint function $h\colon I\!R^n \to I\!R^m$ and let $X \subset I\!R^n$ be a compact set. Then the dual function $D(\mu) = \min_{x\in X} L(x;\mu)$ is concave and $g(\mu) = h(x_\mu)$ with $x_\mu \in \operatorname{Argmin}_{x\in X} L(x;\mu)$ is a supergradient of $D(\mu)$ at $\mu \in$ dom D.*

We apply this result to problem (D3).

Lemma 2 *For a given $\mu \in \tilde{\mathcal{M}}$ let v^J be a (normalized) minimum eigenvector of $C^J(\mu)$, $J \in \mathcal{P}$. Define $x \in I\!R^{n+|\mathcal{P}|}$ by $x_{\tilde{J}} = \sqrt{|J|+1}\cdot v^J$ for $J \in \mathcal{P}$. Then the point $g = (g_1, g_2) \in I\!R^m \times I\!R^{n+|\mathcal{P}|}$ defined by $g_{1i} = \tilde{f}_i(x)$ for $i = 1,\ldots,m$ and $g_{2j} = x_j^2 - 1$ for $j = 1,\ldots,n+|\mathcal{P}|$ is a supergradient of $D_3(\mu)$ at μ.*

Proof. Let $L_3(x;\mu) = d(\mu) + \sum_{J\in\mathcal{P}} L_3^J(x_{\tilde{J}};\mu)$ be the Lagrangian related to (Q3) and $X = \{x \in I\!R^{n+|\mathcal{P}|} \mid x_{\tilde{J}} \in I\!B(|J|+1),\ J \in \mathcal{P}\}$. From the definition of x it follows $x_{\tilde{J}} \in \operatorname{Argmin}_{y\in I\!B(|J|+1)} L_3^J(y;\mu)$. Hence $x \in \operatorname{Argmin}_{y\in X} L_3(y;\mu)$, proving the statement according to Lemma 1. $\square$

3 Duality results

In this section we analyze the dual functions and the optimal values of the dualization schemes (D1), (D2) and (D3).

3.1 Dualization of the trust-region problem

We begin by considering a trust-region problem of the following form

$$\text{(T1)} \qquad \begin{array}{ll} \min & q(x) \\ \text{s.t.} & x \in I\!B(n) \end{array}$$

where $q(x) = \langle x, Bx\rangle + 2\langle b, x\rangle$, $B \in I\!R^{(n,n)}$ and $b \in I\!R^n$. The dual of (T1) is

$$\text{(DT1)} \qquad \max_{\sigma \in I\!R_+} \inf_{x \in I\!R^n} q(x) + \sigma(\|x\|^2 - n).$$

Problem (T1) is one of the few nonconvex all-quadratic optimization problems having a zero duality gap, i.e.

$$\operatorname{val}(T1) = \operatorname{val}(DT1) \tag{3}$$

where $\operatorname{val}(T1)$ and $\operatorname{val}(DT1)$ are the optimal values of (T1) and (DT1) respectively (see [SW95]). If $b = 0$, then (T1) is an eigenvalue problem and it holds $\operatorname{val}(T1) = n \cdot \min\{0, \lambda_1(B)\}$. We consider now the case $b \neq 0$. By replacing $\langle b, x\rangle$ by $x_{n+1} \cdot \langle b, x\rangle$, where $x_{n+1}^2 = 1$, we get the following homogenized formulation of (T1) with $n+1$ variables and an additional equality constraint

$$\text{(T2)} \qquad \begin{array}{ll} \min & \langle x, Cx\rangle \\ \text{s.t.} & \|x\|^2 \leq n+1 \\ & x_{n+1}^2 = 1, \end{array}$$

where

$$C = \begin{pmatrix} B & b \\ b^T & 0 \end{pmatrix}.$$

Clearly, it holds $\operatorname{val}(T1) = \operatorname{val}(T2)$. Dualization of (T2) with respect to the ball $I\!B(n+1)$ yields the dual problem

$$\text{(DT2)} \qquad \max_{\mu \in I\!R_+} (n+1) \cdot \min\{0, \lambda_1(C(\mu))\} - \mu,$$

where $C(\mu) = \begin{pmatrix} B & b \\ b^T & \mu \end{pmatrix}$.

Lemma 3 *It holds* $\operatorname{val}(T2) = \operatorname{val}(T1) = \operatorname{val}(DT2)$.

Proof. This was proved in [RW97]. We repeat the proof in oder to keep the paper self-contained.

$$
\begin{aligned}
\min_{\|x\|^2\le n} q(x) &= \max_{\mu\in I\!R} \min_{\substack{\|x\|^2\le n\\ y^2=1}} \langle x, Bx\rangle + 2y\langle b, x\rangle + \mu(y^2-1)\\
&\ge \max_{\mu\in I\!R} \min_{\|x\|^2+y^2\le n+1} \langle x, Bx\rangle + 2y\langle b, x\rangle + \mu(y^2-1)\\
&\ge \max_{\substack{\mu\in I\!R\\ \sigma\in I\!R_+}} \inf_{\substack{x\in I\!R^n\\ y\in I\!R}} \langle x, Bx\rangle + 2y\langle b, x\rangle + \mu(y^2-1) + \sigma(\|x\|^2+y^2-n-1)\\
&= \max_{\sigma\in I\!R_+} \inf_{\substack{x\in I\!R^n\\ y^2=1}} \langle x, Bx\rangle + 2y\langle b, x\rangle + \sigma(\|x\|^2-n)\\
&= \min_{\|x\|^2\le n} \langle x, Ax\rangle + 2\langle b, x\rangle.
\end{aligned}
$$

□

Lemma 4 *Let $\bar{\mu}$ be a solution of (DT2). Then $\sigma^* = -\min\{0, \lambda_1(C(\bar{\mu}))\}$ solves (DT1).*

Proof. Let

$$L_2(x; \sigma, \mu) = \langle x, (C(\mu) + \sigma I)x\rangle - \mu - (n+1)\sigma$$

be the Lagrangian of (T2) and

$$D_2(\sigma, \mu) = \inf_{x\in I\!R^{n+1}} L_2(x; \sigma, \mu)$$

be the corresponding dual function, which can be formulated in close form as

$$D_2(\sigma, \mu) = \begin{cases} -\mu - (n+1)\sigma & \text{if } C(\mu) + \sigma I \succcurlyeq 0 \\ -\infty & \text{else .} \end{cases}$$

For a dual solution point $(\bar{\sigma}, \bar{\mu}) \in \operatorname*{Argmax}_{\sigma\in I\!R_+, \mu\in I\!R} D_2(\sigma, \mu)$ it follows from the close form that $\bar{\sigma} = -\min\{0, \lambda_1(C(\bar{\mu}))\}$. Since $D_2(\bar{\sigma}, \bar{\mu}) = \text{val}(T2)$ from Lemma 3, the solution set of (T2) is in $\operatorname*{Argmin}_{x\in I\!R^{n+1}} L_2(x; \bar{\sigma}, \bar{\mu})$. This proves

$$\text{val}(T1) = \min_{x\in I\!R^{n+1}, x_{n+1}=1} L_2(x; \bar{\sigma}, \bar{\mu}) = \min_{x\in I\!R^n} L(x; \bar{\sigma}).$$

□

3.2 Dual equivalence

We show that the three dualization schemes have the same optimal value.

Proposition 1
(i) $\mathrm{val}(D1) = \mathrm{val}(D2)$.
(ii) $\mathrm{val}(D2) = \mathrm{val}(D3)$.

Proof. (i) From the strong duality result (3) it follows

$$\begin{aligned} D_2^J(\mu) &= \min_{x \in I\!B(|J|)} L_2^J(x;\mu) \\ &= \max_{t \in I\!R_+} \inf_{x \in I\!R^{|J|}} L_2^J(x;\mu) + t \cdot (\|x\|^2 - |J|) \\ &= \max_{t \in I\!R_+} \inf_{x \in I\!R^{|J|}} L_2^J(x;\mu + t \cdot r^J) \\ &= \max_{t \in I\!R_+} \inf_{x \in I\!R^{|J|}} L_1^J(W^J x + u_J;\mu + t \cdot r^J) \\ &= \max_{t \in I\!R_+} D_1^J(\mu + t \cdot r^J). \end{aligned}$$

Therefore,

$$\begin{aligned} \mathrm{val}(D2) &= \max_{\mu \in \mathcal{M}} d(\mu) + \sum_{J \in \mathcal{P}} D_2^J(\mu) \\ &= \max_{\mu \in \mathcal{M}} d(\mu) + \sum_{J \in \mathcal{P}} \max_{t \in I\!R_+} D_1^J(\mu + t \cdot r^J) \\ &= \mathrm{val}(D1). \end{aligned}$$

(ii) We apply Lemma 3 to show that

$$\begin{aligned} D_2^J(\mu) &= \min_{x \in I\!B(|J|)} \langle x, B^J(\mu)x\rangle + 2\langle b^J(\mu), x\rangle + \hat{d}^J(\mu) \\ &= \max_{\tilde{\mu}_{2,n+\pi_J}} \inf_{x \in I\!R^{|J|+1}} \langle x, C^J(\tilde{\mu})x\rangle + \tilde{d}^J(\tilde{\mu}) \\ &= \max_{\tilde{\mu}_{2,n+\pi_J}} D_3^J(\tilde{\mu}), \end{aligned}$$

where $\tilde{\mu}_1 = \mu_1$ and $\tilde{\mu}_{2,1:n} = \mu_2$. Hence,

$$\begin{aligned} \mathrm{val}(D2) &= \max_{\mu \in \mathcal{M}} d(\mu) + \sum_{J \in \mathcal{P}} D_2^J(\mu) \\ &= \max_{\mu \in \mathcal{M}} d(\mu) + \sum_{J \in \mathcal{P}} \max_{\tilde{\mu}_{2,n+\pi_J}} D_3^J(\tilde{\mu}) \\ &= \mathrm{val}(D3), \end{aligned}$$

where $\tilde{\mu}$ is defined as above. $\square$

3.3 Influence of decomposition on the dual function

Denote by D_1^0 the dual function D_1 defined with respect to the trivial partition $\mathcal{P}^0 = \{V\}$. Clearly, it holds $D_1^0(\mu) = D_1(\mu)$ for all $\mu \in \mathcal{M}$. In other words, decomposition does not change the dual function D_1. From Proposition 1 it follows that decomposition also does not change the opimal values of the dual problems (D2) and (D3).

However, decomposition can affect the dual function D_2 and D_3. Denote by D_2^0 the dual function D_2 related to $\mathcal{P}^0$. Then

$$D_2^0(\mu) = \min_{x \in I\!B(n)} L(x;\mu) \leq \min_{x \in X} L(x;\mu) = D_2(\mu),$$

where $X = \{x \in I\!R^n \mid x_J \in I\!B(|J|),\ J \in \mathcal{P}\}$, since $X \subset I\!B(n)$. The following example shows that the above inequality can be strict.

Example 1 *Consider the max-cut problem*

$$\min\{\langle x, Ax \rangle \mid x \in \{-1,1\}^n\},$$

where A is a block-diagonal matrix consisting of submatrices $A^k \in I\!R^{(n_k,n_k)}$, $k = 1,\dots,l$. Assuming $\lambda_1(A^1) < \lambda_1(A^j)$ for $j > 1$, it follows

$$D_2^0(0) = n \cdot \lambda_1(A) < \sum_{k=1}^{l} n_k \lambda_1(A^k) = D_2(0).$$

Since $D_2 = D_3$ and $D_2^0 = D_3^0$ in this case, it holds also $D_3^0(0) < D_3(0)$.

3.4 Lagrangian convexification

The next proposition shows how convex underestimators for (Q) can be obtained from feasible dual points of (D3).

Proposition 2 *Let $\tilde{\mu} \in \tilde{\mathcal{M}}$ be a feasible dual point of (D3) and define $\mu \in \mathcal{M}$ by $\mu_1 = \tilde{\mu}_1$ and $\mu_{2j} = \tilde{\mu}_{2j} + \min\{0, \lambda_1(C^J(\tilde{\mu}))\}$ for $j \in J$ and $J \in \mathcal{P}$. Then*
(i) $D_3(\tilde{\mu}) \leq D_1(\mu)$ and $L(\cdot;\mu)$ is convex,
(ii) if $\tilde{\mu}$ is a solution of (D3), then μ is a solution of (D1).

Proof.
(i) Let $t^J = \min\{0, \lambda_1(C^J(\tilde{\mu}))\}$. From Lemma 3 and 4 it follows

$$D_3^J(\tilde{\mu}) = \min_{x \in I\!R^{|J|+1}} L_3^J(x; \tilde{\mu} + t^J r^{\tilde{J}}) \leq \min_{x \in I\!R^{|J|}} L_2^J(x;\mu) = D_1^J(\mu).$$

Hence, $D_3(\tilde{\mu}) \leq D_1(\mu)$.
(ii) Follows from (i) and Proposition 1.

□

3.5 Modifications

Several simplifications of the dual problem (D3) are possible.

Remark 3 *If all variables of a block $J \in \mathcal{P}$ are binary, i.e. $J \subset B$, we can dualize the related partial Lagrangian function with respect to the sphere $\partial I\!B(|J|)$. This simplifies the dual problem (D3) since the number of dual constraints are reduced. We show that this modification does not change* $\mathrm{val}(D3)$. *To see this, we consider the modified partial dual function of D_3 defined by*

$$\tilde{D}_3^J(\mu) = (|J|+1)\cdot\lambda_1(C^J(\mu)) - \mu_{n+\pi_J}^2 + \hat{d}^J(\mu).$$

Since $\lambda_1(C^J(\mu+t\cdot r^J)) = \lambda_1(C^J(\mu))+t(|J|+1)$ and $\hat{d}^J(\mu+t\cdot r^J) = \hat{d}^J(\mu)-t(|J|+1)$ for all $t \in I\!R$, it holds

$$\tilde{D}_3^J(\mu) = \tilde{D}_3^J(\mu + t\cdot r^J).$$

For $t = \min\{0, -\lambda_1(C^J(\mu))\}$ it follows $\lambda_1(C^J(\mu+t\cdot r^J)) \geq 0$ and therefore

$$\tilde{D}_3^J(\mu + t\cdot r^J) = D_3^J(\mu + t\cdot r^J),$$

which implies that $\mathrm{val}(D3)$ *is not changed.*

Remark 4 *A further simplification can be made in the case $b^J(\mu) = 0$ for all $\mu \in \mathcal{M}$. In this case, the trust region problem (2) is an eigenvalue problem and it holds*

$$D_2^J(\mu) = |J|\cdot\min\{0, \lambda_1(B^J(\mu))\} + \hat{d}^J(\mu).$$

From Lemma 3 it follows that $D_3^J(\mu)$ can be replaced by $D_2^J(\mu)$ without changing $\mathrm{val}(D3)$.

Remark 5 *If $A^J(\mu)$ is zero for all $\mu \in \mathcal{M}$, the related Lagrangian problem is linear and therefore separable with respect to all variables of this block. Hence, we can assume $J = \{j^J\}$, i.e $I\!B(|J|) = [-1,1]$. Then*

$$\min_{x\in[-1,1]} L_2^J(x;\mu) = \min_{x\in[-1,1]} 2\langle b^J(\mu), x\rangle + \hat{d}^J(\mu) = 2\min\{b_{j^J}(\mu)\underline{x}_{j^J}, b_{j^J}(\mu)\overline{x}_{j^J}\}.$$

If (Q) is a MILP, this yields the so-called LP-relaxation.

4 Splitting schemes

We discuss now splitting schemes for transforming non-separable MIQQPs into block-separable MIQQPs. This technique goes back to 1956 [DR56] where it was used for partial differential equations. It is much used in stochastic programming [Rus97] and in combinatorial optimization [GK87].

4.1 Sparsity graph

Problem (Q) has a sparse structure if most of the entries a^k_{ij} of the matrices A^k, $k = 0, \ldots, m$, are zero. We define the sparsity pattern of problem (Q) by

$$E_Q = \{ij \in V^2 \mid a^k_{ij} \neq 0 \text{ for some } k \in \{0, \ldots, m\}\}.$$

The graph (V, E_Q) is called sparsity-graph of (Q). Consider a partition $\mathcal{P} = \{J_1, \ldots, J_l\}$ of V. We define the set of nodes in J_p, $p = k+1, \ldots, l$, connected to J_k by

$$V(J_k) = \{i \in \bigcup_{p=k+1}^{l} J_p \mid ij \in E_Q,\, j \in J_k\},$$

for $k = 1, \ldots, l$. The set $V(J_k)$ can be interpreted as the set of flows of a network problem connecting a component J_k with components J_p, $p > k$. If (Q) is block-separable with respect to the blocks $J \in \mathcal{P}$, then $V(J) = \emptyset$ for all $J \in \mathcal{P}$. Since

$$\langle x, A^i x\rangle = \sum_{k=1}^{l} \langle x_{J_k}, A^i_{J_k J_k} x_{J_k}\rangle + 2 \sum_{p=k+1}^{l} \langle x_{J_k}, A^i_{J_k, J_p} x_{J_p}\rangle,$$

it follows

$$\langle x, A^i x\rangle = \sum_{J\in\mathcal{P}} \langle x_J, A^i_{JJ} x_J\rangle + 2 \sum_{J\in\mathcal{P}} \langle x_J, A^i_{J,V(J)} x_{V(J)}\rangle \tag{4}$$

for $i = 0, \ldots, m$.

4.2 Splitting sparse MIQQPs

For a given partition $\mathcal{P}$ of V, the following MIQQP is a splitting scheme of (Q)

$$\text{(S)} \qquad \begin{array}{lll} \min & \tilde{f}_0(x, y) & \\ \text{s.t.} & \tilde{f}_E(x, y) = 0, & \\ & \tilde{f}_I(x, y) \leq 0, & \\ & x_{V(J)} = y^J, & J \in \mathcal{P}, \\ & x \in [\underline{x}, \overline{x}], & \\ & y^J \in [\underline{x}_{V(J)}, \overline{x}_{V(J)}], & J \in \mathcal{P} \\ & x_B \text{ binary}, & \end{array}$$

where

$$\tilde{f}_k(x, y) = \sum_{J\in\mathcal{P}} \langle x_J, A^k_{JJ} x_J\rangle + 2 \sum_{J\in\mathcal{P}} \langle x_J, A^k_{J,V(J)} y^J\rangle + 2\langle a^k, x\rangle + d_k,$$

$k = 0, \ldots, m$. Problem (S) is block-separable with respect to the blocks $(x_J, y^J)_{J\in\mathcal{P}}$, $J \in \mathcal{P}$. From (4) it follows that

$$\tilde{f}_k(x, (x_{V(J)})_{J\in\mathcal{P}}) = f_k(x),$$

implying that the optimal values of (Q) and (S) are equivalent. The next Proposition shows that the optimal values of the related dual problems can be different.

Proposition 3 *Let (D1) be the classical dual of (Q) and let (DS) be the dual of (S), as defined in Section 2.1. Then* $\mathrm{val}(DS) \leq \mathrm{val}(D1)$.

Proof. See [LR01].

Depending on the cardinalities of the sets $V(J)$, $J \in \mathcal{P}$, the splitting scheme (S) will be efficient. We consider now special cases where decomposition could be efficient.

4.3 Block-angular structure

Problem (Q) has a block-angular structure if the matrices are of the form

$$A^i = \begin{pmatrix} A_i^1 & & & B_i^1 \\ & \ddots & & \vdots \\ & & A_i^{l-1} & B_i^{l-1} \\ (B_i^1)^T & \dots & (B_i^{l-1})^T & A_i^l \end{pmatrix}.$$

Problems with such a structure arise, for example, in process system engineering, telecommunication problems, network problems and stochastic programming. In [FH98] it is demonstrated that many sparse optimization problems can be efficiently transformed into problems with block-angular structure.

Let $\mathcal{P} = \{J_1, \dots, J_l\}$ be a partition of V according to the above block-structure. Then $V(J) = J_l$ for $J \neq J_l$. The related splitting scheme is block separable with respect to l blocks with block sizes $|J_1| + |J_l|, \dots, |J_{l-1}| + |J_l|, |J_l|$. It follows that the number of additional variables in the splitting scheme (S) is $(l-1)|J_l|$.

4.4 Band structure

Problem (Q) has a band-structure if the matrices have the form

$$A^i = \begin{pmatrix} A_i^1 & B_i^1 & & \\ (B_i^1)^T & \ddots & \ddots & \\ & \ddots & A_i^{l-1} & B_i^{l-1} \\ & & (B_i^{l-1})^T & A_i^l \end{pmatrix}.$$

There exist many methods for transforming sparse matrices into matrices with band-structure. A main application of these algorithms is to reduce the fill-in of a Cholesky factorization.

Let $\mathcal{P} = \{J_1, \dots, J_l\}$ be a partition of V according to the above block-structure. Then $V(J_k) = J_{k+1}$ for $k = 1, \dots, l-1$ and $V(J_l) = \emptyset$. The related splitting scheme is block separable with respect to l blocks with block sizes $|J_1|+|J_2|, \dots, |J_{l-1}|+|J_l|, |J_l|$. It follows that the number of additional variables in the splitting scheme (S) is not greater than $\sum_{k=2}^{l} |J_k| = n - |J_1|$.

5 Preliminary numerical results

We implemented the evaluation of the dual function D_3 with the modifications of Remarks 3, 4 and 5 in C++. Supergradients of D_3 were computed according to Lemma 2. For the computation of a minimum eigenvalue and minimum eigenvector we used two algorithms. The first algorithm is an implicit QL-method from the EISPACK-library [Eis73], used if the dimension of the matrix is less than or equal to 50. If the dimension is greater than 50, we used the Lanczos method ARPACK++ [GS97]. The proximal bundle code NOA 3.0 [Kiw94] of Kiwiel described in [Kiw90] was used for maximizing the dual function D_3. In order to compare the computation with and without decomposition, we stoped the dual method after 100 iterations using the following parameters of NOA: optimality tolerance = 0.0001, bundle size = 50, linesearch decrease = 0.1, QP weight = 10.0 and feasisibility tolerance = 0.1.

5.1 Block structure

For testing the decomposition scheme (D3), we produced block-separable random MIQQPs using the following procedure with parameters n, the number of variables, m, the number of constraints, and p, the block size.

Procedure `rand_miqqp`(n, m, p)

1. set $l = n/p$ (number of blocks)
2. set $B = \{1, \dots, n/2\}$, $C = \{n/2+1, \dots, n\}$, $\underline{x} = -e$ and $\overline{x} = e$
3. set $I = \{1, \dots, m/2\}$ and $E = \{m/2+1, \dots, m\}$
4. compute symmetric dense matrices $A_i^k \in I\!\!R^{(p,p)}$ with uniformly distributed random components in $[-10, 10]$ for $i = 0, \dots, m$, $k = 1, \dots, l$
5. compute vectors $a_i \in I\!\!R^n$ with uniformly distributed random components in $[-10, 10]$ for $i = 0, \dots, m$
6. set $d_i = 0$ for $i = 0, \dots, m$.

The above procedure produces a MIQQP, which is block separable with respect to the blocks $J_k = \{(k-1)p+1, \dots, kp\}$, $k = 1, \dots, l$. Since $d_i = 0$ for $i = 0, \dots, m$,

$x = 0$ is a feasible point. Therefore, the dual function D_3 is bounded. In order to study the influence of decomposition, we generated two dual problems of the form (D3) related to the partitions $\mathcal{P}_1 = \{V\}$ and $\mathcal{P}_2 = \{J_1, \dots, J_l\}$. The first dual problem is called (D-1) and the second (D-2).

For a given set of input parameters (n, m, p) we produced 10 random MIQQPs with the procedure `rand_miqqp` and solved (D-1) and (D-2) using the bundle method NOA with the parameters previously described. The averaged computing time for solving (D-1) and (D-2) is denoted by t_1 and t_2 respectively. The averaged value of the dual funcion of (D-1) and (D-2) at iteration 100 is denoted by d_1 and d_2 respectively. Tables 1-4 show the fractions t_1/t_2 and $100d_2/d_1$ for different input parameters of `rand_miqqp`.

It can be seen from the tables that the decomposition scheme accelerates the running time considerably. The acceleration is particularly large if the number of constraints is high. This is due to the increased cost for the matrix-vector multiplication used in the Lanczos algorithm.

Decomposition also makes the dual solution method more stable. We realized convergence problems of the Lanczos method if the optimality tolerance of the dual solver is very small. In contrast, the QL-method was very stable.

n	200	400	600	800	1000
t_1/t_2	37.8335	103.277	96.888	100.046	107.177
t_2	0.787	0.7985	0.8113	0.823	0.8416
$100 \cdot d_2/d_1$	76.4902	67.6819	61.4832	58.2171	54.7593

Table 1: number of constr. $m = 0$ and block-size $p = 10$

n	200	400	600	800	1000
t_1/t_2	74.8061	87.1678	82.393	87.0602	85.9397
t_2	1.328	1.347	1.371	1.399	1.414
$100 \cdot d_2/d_1$	91.0068	86.4787	82.0591	78.8589	76.0609

Table 2: number of constr. $m = 0$ and block-size $p = 20$

m	0	4	8	12	16	20
t_1/t_2	36.398	42.4635	32.4808	28.8326	25.4406	24.4572
t_2	0.784	0.919	1.069	1.153	1.364	1.507
$100 \cdot d_2/d_1$	84.6635	83.1523	80.5309	75.9177	68.6236	60.9222

Table 3: dimension $n = 200$ and block-size $p = 10$

m	0	4	8	12	16	20
t_1/t_2	37.7381	70.5956	43.8506	36.3231	20.4973	18.9905
t_2	1.321	1.491	1.66	1.755	2.035	2.209
$100 \cdot d_2/d_1$	94.7573	95.2154	92.8278	91.9332	85.3104	82.5667

Table 4: dimension $n = 200$ and block-size $p = 20$

5.2 Network structure

In order to study splitting schemes, we experimented with random Max-Cut problems of the form

$$\min\{\langle x, Ax\rangle \mid x \in \{-1,1\}^n\},$$

where $A \in I\!R^{(n,n)}$ is the sparse matrix

$$A = \begin{pmatrix} A^1 & B^1 & 0 & B^l \\ (B^1)^T & \ddots & \ddots & 0 \\ 0 & \ddots & A^{l-1} & B^{l-1} \\ (B^l)^T & 0 & (B^{l-1})^T & A^l \end{pmatrix}.$$

The submatrices $A^k \in I\!R^{(p,p)}$, $k = 1, \ldots, l$, are dense with a block-size $p = n/l$. The submatrices $B^k \in I\!R^{(p,p)}$ are sparse with nonzero entries at $(p-i, i)$, $i = 1, \ldots, s$, where $s \in \{0, \ldots, p\}$ is a given flow size. The resulting sparsity graph has a ring topology with l components which are each connected by s arcs. All nonzero components of A are uniformly distributed random numbers in $[-10, 10]$. For a given Max-Cut problem we generated a splitting scheme (S), as described in the previuos section using the partition $\mathcal{P} = \{J_1, \ldots, J_l\}$ with $J_k = \{(k-1)p + 1, \ldots, k \cdot p\}$, $k = 1, \ldots l$. For the original Max-Cut problem as well for the splitting scheme we constructed a dual problem of the form (D3), which we call (D-1) and (D-2) respectively.

As in the previous experiment, we produced 10 random Max-Cut problems for a given set of input parameters (n, p, s) and solved (D-1) and (D-2) using the bundle method NOA with the parameters previously described. The averaged computing time in seconds for solving (D-1) and (D-2) is denoted by t_1 and t_2 respectively. The averaged value of the dual funcion of (D-1) and (D-2) at iteration 100 is denoted by d_1 and d_2 respectively. Tables 5-8 show the fraction t_1/t_2, the running time t_2 and the percentage ratio of the dual function values $100d_2/d_1$ for different input parameters.

The results show that the splitting scheme accelates the evaluation of the dual by magnitudes. In contrast to the previous experiment with block-separable MIQQPs, the dual bounds of the decomposed programs are slightly worse than the ones of the original programs.

n	200	400	600	800	1000
t_1/t_2	82.582	44.9816	24.0894	19.5989	14.3374
t_2	1.516	2.117	2.696	3.262	3.806
$100 \cdot d_2/d_1$	108.924	107.011	102.419	102.63	100.431

Table 5: flow size $s = 2$ and block-size $p = 10$

n	200	400	600	800	1000
t_1/t_2	66.717	98.1649	71.2151	63.5916	46.537
t_2	1.734	1.96	2.139	2.356	2.544
$100 \cdot d_2/d_1$	107.812	107.452	106.515	105.458	101.546

Table 6: flow-size $s = 2$ and block-size $p = 20$

n	200	400	600	800	1000
t_1/t_2	45.6438	29.7396	16.3761	11.1594	8.89106
t_2	2.473	3.723	4.94	6.183	7.42
$100 \cdot d_2/d_1$	113.764	112.878	110.676	108.665	106.788

Table 7: flow-size $s = 4$ and block-size $p = 10$

n	200	400	600	800	1000
t_1/t_2	39.3363	74.5524	56.4051	44.4981	32.8466
t_2	2.218	2.67	3.089	3.529	3.969
$100 \cdot d_2/d_1$	112.244	112.121	111.659	110.65	109.205

Table 8: flow-size $s = 4$ and block-size $p = 20$

6 Concluding Remarks

We presented a Lagrangian decomposition scheme for block-separable MIQQPs inducing partial Lagrangian problems of the form of eigenvalue problems or linear programs. Preliminary numerical experiments on random problems demonstrate that this scheme is able to accelerate the running time of a dual solution algorithm by magnitudes. Moreover, it makes it possible to use the QL-algorithm for eigenvalue computation, which seems to be more stable than the Lanczos algorithm.

For non-separable MIQQPs, we discussed splitting schemes for generating block-separable reformulations of the problem. These schemes are efficient if it is possible to find good partitions of the sparsity graph related to the given problem.

Lagrangian decomposition provides a lower bound on the optimal value. This can be used as a quality measure for the best solution or as a lower bounding procedure in a branch-and-bound (B&B) algorithm. Under mild assumptions, it can be shown that B&B-algorithms using duality bounds are convergent (see [Dür01] for continuous optimization problems with compact feasible sets and [Now00] for all-quadratic problems).

Furthermore, Lagrangian decomposition can be used to compute primal solutions via Lagrangian heuristics. Proposition 2 gives a recipe for constructing convex underestimators of the objective function over the feasible set. Using smoothing (homotopy) methods, this can be used to compute local minimizers of the problem. In [AN02] a MaxCut heuristic based on this idea is presented.

Acknowledgment. We would like to thank Prof. Kiwiel for very fruitful discussions and for making NOA 3.0 available. We thank Stefan Vigerske for helping us with the C++ programming.

References

[Ada99] H.D. Sherali W.P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems.* Kluwer Academic Publishers, 1999.

[AKLvV95] F. A. Al-Khayyal, C. Larsen, and T. van Voorhis. A relaxation method for nonconvex quadratically constrained quadratic programs. *J. Glob. Opt*, 6:215–230, 1995.

[AKvV96] F. A. Al-Khayyal and T. van Voorhis. Accelerating convergence of branch-and-bound algorithms for quadratically constrained optimization problems. In *State of the Art in Global Optimization: Computational Methods and Applications, C. A. Floudas (ed.).* Kluwer Academic Publisher, 1996.

[AN02] H. Alperin and I. Nowak. Lagrangian Smoothing Heuristics for MaxCut. Technical report, HU–Berlin NR–2002–6, 2002.

[BCC93] E. Balas, S. Ceria, and G. Cornuejols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Progr.*, 58:295–324, 1993.

[BFG+00] R.E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. Mip: theory and practice - closing the gap. In M.J.D. Powell and S. Scholtes, editors, *System Modelling and Optimization: Methods, Theory and Applications*, pages 19–49. Kluwer Dordrecht, 2000.

[CS93] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups.* 2nd edn, Springer, New York, 1993.

[CS95] P. Chardaire and A. Sutter. A decomposition method for quadratic zero-one programming. *Management Science*, 41:704–712, 1995.

[DR56] J. Douglas and H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956.

[DT92] E. V. Demands and C. S. Tang. Linear control of a markov production system. *Operation Research*, 40:259–278, 1992.

[Dür01] M. Dür. Dual bounding procedures lead to convergent Branch-and-Bound algorithms. *Math. Progr.*, 91:117–125, 2001.

[DW60] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operation Research*, 8:101–111, 1960.

[Eis73] *EISPACK.* http://www.netlib.org/eispack/, 1972-1973.

[FH98] M. C. Ferris and J. D. Horn. Partitioning mathematical programs for parallel solution. *Math. Progr.*, 80:35–61, 1998.

[FKN97] K. Fujisawa, M. Kojima, and K. Nakata. Exploiting Sparsity in Primal-Dual Interior-Point Methods for Semidefinite Programming. *Math. Progr.*, 79:235–254, 1997.

[FS99] J. A. Filar and T. A. Schultz. Bilinear programming and structured stochastic games. *J. Opt. Theor. Appl.*, 53:85–104, 1999.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, New York, 1979.

[GK87] M. Guignard and S. Kim. Lagrangian decomposition: a model yielding stronger Lagrangean bounds. *Math. Progr.*, 39(2):215–228, 1987.

[GS97] F. Gomes and D. Sorensen. `ARPACK++`*: a* `C++` *Implementation of* `ARPACK` *eigenvalue package*, 1997. `http://www.crpc.rice.edu/software/ARPACK/`.

[Hel00] C. Helmberg. Semidefinite Programming for Combinatorial Optimization. Technical report, ZIB–Report 00–34, 2000.

[hH82] E. Phan huy Hao. Quadratically constrained quadratic programming: Some applications and a method for solution. *Zeitschrift für Operations Research*, 26:105–119, 1982.

[HPT95] R. Horst, P. Pardalos, and N. Thoai. *Introduction to Global Optimization.* Kluwer Academic Publishers, 1995.

[HR00] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10(3):673–695, 2000.

[HUL93] J. B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I and II.* Springer, Berlin, 1993.

[Kc00] M. Kojima and L. Tunçel. Cones of matrices and successive convex relaxations of nonconvex sets. *SIAM*, 10:750–778, 2000.

[Kiw90] K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Progr.*, 46:105–122, 1990.

[Kiw94] K. C. Kiwiel. *User's Guide for* `NOA 2.0/3.0`*: A* `FORTRAN` *Package for Convex Nondifferentiable Optimization.* Polish Academy of Science, System Research Institute, Warsaw, 1993/1994.

[KK00] S. Kim and M. Kojima. Second Order Cone Programming Relaxation of Nonconvex Quadratic Optimization Problems. Technical report, Research Reports on Mathematical and Comuting Sciences, Series B: Operations Research, Tokyo Institute of Technology, 2000.

[LR01] C. Lemaréchal and A. Renaud. A geometric study of duality gaps, with applications. *Mathematical Programming*, 90:399–427, 2001.

[Neu01] A. Neumaier. Constrained Global Optimization. In *COCONUT Deliverable D1, Algorithms for Solving Nonlinear Constrained and Optimization Problems: The State of The Art*, pages 55–111. http://www.mat.univie.ac.at/∼neum/glopt/coconut/StArt.html, 2001.

[Now00] I. Nowak. Dual bounds and optimality cuts for all-quadratic programs with convex constraints. *J. Glob. Opt.*, 18:337–356, 2000.

[PS01] P. Parrilo and B. Sturmfels. Minimizing polynomial functions. *to appear in DIMACS volume of the Workshop on Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science*, 2001.

[PTA94] Thai Quynh Phing, Pham Dinh Tao, and Le Thi Hoai An. A method for solving D.C. programming problems, Application to fuel mixture nonconvex optimization problems. *J. Global Opt.*, 6:87–105, 1994.

[RS71] D. P. Rutenberg and T. L. Shaftel. Product design: Sub-assemblies for multiple markets. *Management Science*, 18:B220–B231, 1971.

[Rus97] A. Ruszczyński. Decomposition methods in stochastic programming. *Math. Progr.*, 79:333–353, 1997.

[RW97] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Math. Progr.*, 77 (2):273–299, 1997.

[SW95] R. Stern and H. Wolkowicz. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optimization*, 5(2):286–313, 1995.

[Tho00] N. Thoai. Duality bound method for the general quadratic programming problem with quadratic constraints. *Journal of Optimization Theory and Applications*, 107 (2), 2000.

[VF90] V. Visweswaran and C. A. Floudas. A global optimization algorithm (GOP) for certain classes of nonconvex NLPs : II. Application of theory and test problems. *Comp. Chem. Eng.*, 1990.

[WSV00] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of Semidefinite Programming.* Kluwer Academic Publishers, 2000.

[WV91] A. Weintraub and J. Vera. A cutting plane approach for chance-constrained linear programs. *Operations Research*, 39:776–785, 1991.