Mathematical Programming manuscript No.
(will be inserted by the editor)

Ivo Nowak

Lagrangian Decomposition of Block-Separable Mixed-Integer All-Quadratic Programs ⋆

the date of receipt and acceptance should be inserted later

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 propose splitting schemes for reformulating non-separable problems as block-separable problems. Second 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. 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

Address(es) of author(s) should be given

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

1. Introduction

In this paper, we propose an efficiently computable dual bound for block-separable nonconvex mixed-integer all-quadratic programs (MIQQP). Since sparse problems can be reformulated to be block-separable, the proposed method can also be applied to sparse MIQQPs.

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 [44], modularization of product sub-assemblies [41] and special cases of structured stochastic games [15]. Other applications are packing problems [9], minmax location problems [24], chance-constrained problems in portfolio optimization [11,24,45], fuel mixture problems [37], placement and layout problems in integrated circuit design [2,3]. Several mixed-integer nonlinear programs (MINLP) can be reformulated as an MIQQP, for example polynomial programs [36]. Under mild assumptions it can be shown that every MINLP can be approximated by a MIQQP with arbitrary precision [33].

Since MIQQP is a difficult NP-hard [17] 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) [33], which can be solved in quite high dimensions [6]. 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 linear programming (LP) [5,1,43,28] are fast, but often too weak. Semidefinite programming (SDP) [31] provides stronger relaxations. However, in large dimensions the cost for solving SDPs may be too large to be practicable. Attempts to reduce this cost are based on exploiting sparsity [16, 21] and second-order cone programming [25].

A conceptual different approach is Lagrangian decomposition (LD). Originally utilized by Dantzig and Wolfe for exploiting block-angular structure of LPs [10] 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 [8]. 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 general block-separable MIQQPs. In Section 2, we discuss splitting schemes for reformulating sparse MIQQPs to have a block-structure, and introduce a homogenized block-separable reformulation, whose dual is equivalent to an eigenvalue optimization problem. Based on strong duality of the trust-region problem, we show in Section 3 that

the optimal value of the eigenvalue optimization problem is equivalent to the optimal value of the Lagrangian dual of the original problem. The proof is a straight-forward generalization of a dual-equivalent result in [38] on quadratic binary programs. In [38] the problem is dualized with respect to a full-dimensional sphere, whereas here the problem is dualized with respect to a Cartesian product of low-dimensional balls. In Section 4 we describe a method for solving the eigenvalue optimization problem. We report preliminary numerical results on random MIQQPs in Section 5 showing that decomposition is able to accelerate the computation of the dual bound considerably. We finish with conclusions in Section 6.

1.1. Notation

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}. 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. Block-separable reformulations

2.1. Problem statement

A general nonconvex MIQQP can be formulated as

$$
\text{(Q)} \qquad \begin{aligned} \min\ & q_0(x) \\ \text{s.t. }\ & q_E(x) = 0, \\ & q_I(x) \leq 0, \\ & r_B(x) = 0, \\ & r_C(x) \leq 0, \end{aligned}
$$

where

$$r(x) = \text{Diag}(x - \underline{x})(x - \overline{x}), \tag{1}$$

$\underline{x}, \overline{x} \in I\!R^n$, $B \cup C = \{1, \ldots, n\}$ with $B \cap C = \emptyset$, $I \cup E = \{1, \ldots, m\}$ with $I \cap E = \emptyset$, $q_i(x) = \langle x, A^i x\rangle + 2\langle b^i, x\rangle + c_i$, $A^i \in I\!R^{(n,n)}$ is symmetric, $b^i \in I\!R^n$, $c_i \in I\!R$, $i = 0, \ldots, m$. Note that the last two constraints in (Q) are equivalent to 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.

We assume that problem (Q) is *block-separable*, i.e. there exists a partition $\mathcal{P} = \{J_1, \ldots, J_p\}$ of $\{1, \ldots, n\}$ with $\bigcup_{k=1}^p J_k = \{1, \ldots, n\}$ and $J_i \cap J_k = \emptyset$ if $i \neq k$, such that

$$q_i(x) = c_i + \sum_{k=1}^{p} q_i^k(x_{J_k}), \tag{2}$$

where $q_i^k(x_{J_k}) = \langle x_{J_k}, A^i_{J_k J_k} x_{J_k}\rangle + 2\langle b^i_{J_k}, x_{J_k}\rangle$ for $i = 0, \ldots, m$. In other words, the matrices A^i are block-diagonal with $A^i_{J_k J_l} = 0$ for $k \neq l$. We denote by $n_k = |J_k|$ the size of a block J_k.

2.2. Splitting schemes

We discuss now splitting schemes for transforming non-separable sparse MIQQPs into block-separable MIQQPs. This technique goes back to 1956 [12] where it was

used for partial differential equations. It is much used in stochastic programming [40] and in combinatorial optimization [19].

2.2.1. Sparsity graph Problem (Q) has a sparse structure if most of the entries a^i_{kl} of the matrices A^i, $i = 0, \ldots, m$, are zero. Let $V = \{1, \ldots, n\}$ be a set of nodes, where a node i represents a variable x_i. We define the sparsity pattern of problem (Q) by

$$E_Q = \{kl \in V^2 \mid a^i_{kl} \neq 0 \text{ for some } i \in \{0, \ldots, m\}\}.$$

The graph (V, E_Q) is called sparsity-graph of (Q). We define the set of nodes of $\bigcup_{l=k+1}^{p} J_l$ connected to J_k by $R_k = \{i \in \bigcup_{l=k+1}^{p} J_l \mid ij \in E_Q, j \in J_k\}$, for $k = 1, \ldots, p$. The set R_k can be interpreted as the set of flows of a nonlinear network problem connecting a component J_k with components J_l, where $k < l \leq p$. If (Q) is block-separable with respect to the blocks $J_k, k = 1, \ldots, p$, then $R_k = \emptyset$.

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

$$\begin{aligned}
&\min\ \tilde{q}_0(x, y)\\
&\text{s.t. } \tilde{q}_E(x, y) = 0,\\
&\quad \tilde{q}_I(x, y) \leq 0,\\
&\quad r_C(x) \leq 0,\\
&\quad r_B(x) = 0,\\
&\quad x_{R_k} = y^k, \qquad k = 1, \ldots, p,\\
&\quad r_{C \cap J_k}(y^k) \leq 0, \quad k = 1, \ldots, p,\\
&\quad r_{B \cap J_k}(y^k) = 0, \quad k = 1, \ldots, p,
\end{aligned} \tag{S}$$

where

$$\tilde{q}_i(x,y) = \sum_{k=1}^{p} \langle x_{J_k}, A^i_{J_k J_k} x_{J_k} \rangle + 2 \sum_{k=1}^{p} \langle x_{J_k}, A^i_{J_k R_k} y^k \rangle + 2\langle b^i, x \rangle + c_i,$$

$i = 0, \ldots, m$. Problem (S) is block-separable with respect to the blocks (x_{J_k}, y^k), $k = 1, \ldots, p$. Since

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

it follows

$$\langle x, A^i x \rangle = \sum_{k=1}^{p} \langle x_{J_k}, A^i_{J_k J_k} x_{J_k} \rangle + 2 \sum_{k=1}^{p} \langle x_{J_k}, A^i_{J_k R_k} x_{R_k} \rangle \tag{3}$$

for $i = 0, \ldots, m$. From (3) it follows that $\tilde{q}_i(x, (x_{R_k})_{k=1,\ldots,p}) = q_i(x)$, implying that the optimal values of (Q) and (S) are equivalent.

Depending on the cardinalities of the sets R_k, $k = 1, \ldots, p$, the splitting scheme (S) will be efficient. We consider now special cases where decomposition could be efficient.

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

$$A^i = \begin{pmatrix} A^1_i & & & B^1_i \\ & \ddots & & \vdots \\ & & A^{p-1}_i & B^{p-1}_i \\ (B^1_i)^T & \ldots & (B^{p-1}_i)^T & A^p_i \end{pmatrix}.$$

Problems with such a structure arise, for example, in process system engineering, telecommunication problems, network problems and stochastic programming. In [14] it is demonstrated that many sparse optimization problems can be efficiently

transformed into problems with block-angular structure. Automatic detection of block-structure of sparse MILPs is dicussed in [30].

Let $\mathcal{P} = \{J_1, \ldots, J_p\}$ be a partition of V according to the above block-structure. Then $R_k = J_p$ for $k \neq p$. The related splitting scheme is block-separable with respect to p blocks with block sizes $n_1 + n_p, \ldots, n_{p-1} + n_p, n_p$. It follows that the number of additional variables in the splitting scheme (S) is $(p-1)n_p$.

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

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

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 $\mathcal{P} = \{J_1, \ldots, J_p\}$ be a partition of V according to the above block-structure. Then $R_k = J_{k+1}$ for $k = 1, \ldots, p-1$ and $R_p = \emptyset$. The related splitting scheme is block-separable with respect to p blocks with block sizes $n_1 + n_2, \ldots, n_{p-1} + n_p, n_p$. It follows that the number of additional variables in the splitting scheme (S) is not greater than $\sum_{k=2}^{p} n_k = n - n_1$.

2.3. Standardization and homogenization

We show that problem (Q) can be formulated in such a way, that all variables are bounded by -1 and 1 and all linear terms $\langle b^i, x\rangle$ of the functions q_i in (Q) disappear. This formulation allows to formulate the Lagrangian dual problem as a block-separable eigenvalue optimization problem, which can be solved efficiently (see next section). The transformation is carried out in two steps.

Let $u = \frac{1}{2}(\overline{x}+\underline{x})$ be the center and $w = \frac{1}{2}(\overline{x}-\underline{x})$ 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 $[\underline{x}, \overline{x}]$. In the first step of the transformation, called *standardization*, we replace the variables x of (Q) by $\theta(x)$. The transformed quadratic forms take the form

$$\hat{q}_i(x) = q_i(\theta(x)) = \langle x, \hat{A}^i x\rangle + 2\langle \hat{b}^i, x\rangle + \hat{c}_i, \quad i = 0, \ldots, m, \tag{4}$$

where $\hat{A}^i = WA^iW$, $\hat{b}^i = Wb^i + WA^iu$, $\hat{c}_i = u^TAu + 2u^Tb^i + c_i$ and $W = \mathrm{Diag}(w)$. In the second step of the transformation, we *homogenize* the problem by replacing linear terms $\langle \hat{b}^i_{J_k}, x_{J_k}\rangle$ by quadratic terms $x_{n+k} \cdot \langle \hat{b}^i_{J_k}, x_{J_k}\rangle$ and adding constraints $x^2_{n+k} - 1 = 0$. This gives the problem

$$(\tilde{Q}) \qquad \begin{aligned} &\min\ \tilde{q}_0(x) \\ &\text{s.t. } \tilde{q}_E(x) = 0, \\ &\quad \tilde{q}_I(x) \le 0, \\ &\quad x_j^2 - 1 \le 0, \quad j \in C \\ &\quad x_j^2 - 1 = 0, \quad j \in B \cup \{n+1, \ldots, n+p\} \end{aligned}$$

where $\tilde{q}_i(x) = \hat{c}_i + \sum_{k=1}^{p} \tilde{q}_i^k(x_{\tilde{J}_k})$, $\tilde{q}_i^k(x_{\tilde{J}_k}) = \langle x_{J_k}, \hat{A}^i_{J_k J_k} x_{J_k} \rangle + 2x_{n+k}\langle \hat{b}^i_{J_k}, x_{J_k} \rangle$, and $\tilde{J}_k = J_k \cup \{n+k\}$. Obviously, $\tilde{q}_i^k(x) = q_i^k(\hat{x})$, if $x_{1:n_k} = \hat{x}$ and $x_{n_k+1} = 1$ or $x_{1:n_k} = -\hat{x}$ and $x_{n_k+1} = -1$. Therefore, the optimal values of $(\tilde{Q})$ and (Q) coincide. Since each additional variable can be 1 or -1, the number of solutions of $(\tilde{Q})$ is 2^p times larger than of (Q).

3. Duality results

In this section we formulate and analyze Lagrangian dual problems related to the all-quadratic problems (Q) and $(\tilde{Q})$.

3.1. Lagrangian dual problems

Let $q(x) = (q_1(x), \ldots, q_m(x))^T$ and $r(x)$ be defined as in (1). By introducing the Lagrangian function

$$L(x; \mu) = q_0(x) + \langle \mu^q, q(x) \rangle + \langle \mu^r, r(x) \rangle$$

and the Lagrangian multiplier set

$$\mathcal{M} = \{\mu = (\mu^q, \mu^r) \in I\!R^m \times I\!R^n \mid \mu^q_I \geq 0,\ \mu^r_C \geq 0\}$$

we formulate the Lagrangian dual of (Q) by

$$\text{(D)} \qquad \begin{array}{c} \max\ D(\mu) \\ \text{s.t. } \mu \in \mathcal{M} \end{array}$$

where $D(\mu) = \inf_{x \in I\!R^n} L(x; \mu)$ is the dual function. Since (Q) is block-separable, i.e. (2) holds, the dual function D decomposes into $D(\mu) = c(\mu) + \sum_{k=1}^{p} D_k(\mu)$,

with $c(\mu) = c_0 + \sum_{i=1}^{m} \mu_i^q c_i$ and $D_k(\mu) = \min_{x \in I\!R^{n_k}} L_k(x;\mu)$ where

$$L_k(x_{J_k};\mu) = q_0^k(x_{J_k}) + \langle \mu^q, q^k(x_{J_k})\rangle + \langle \mu^r_{J_k}, r_{J_k}(x)\rangle$$

with $q^k(x) = (q_1^k(x), \ldots, q_m^k(x))^T$. We define a partial Lagrangian related to $(\tilde{Q})$ by

$$\tilde{L}_k(x_{\tilde{J}_k};\mu) = \tilde{q}_0^k(x_{\tilde{J}_k}) + \langle \mu^q, \tilde{q}^k(x_{\tilde{J}_k})\rangle + \langle \mu^r_{\tilde{J}_k}, \mathrm{Diag}(x_{\tilde{J}_k})x_{\tilde{J}_k} - e\rangle.$$

with $\tilde{q}^k(x) = (\tilde{q}_1^k(x), \ldots, \tilde{q}_m^k(x))^T$. Then the related partial dual function is the following eigenvalue function

$$\tilde{D}_k(\mu) = \min_{x \in I\!B(n_k+1)} \tilde{L}_k(x;\mu) = (n_k+1)\cdot \min\{0, \lambda_1(\tilde{A}^k(\mu))\} - \langle e, \mu^r_{\tilde{J}_k}\rangle$$

where

$$\tilde{A}^k(\mu) = \begin{pmatrix} \hat{A}^k(\mu) & \hat{b}^k(\mu) \\ \hat{b}^k(\mu)^T & \mu^r_{n+k} \end{pmatrix}, \tag{5}$$

$\hat{A}^k(\mu) = \hat{A}^0_{J_kJ_k} + \sum_{i=1}^m \mu_i^q \hat{A}^i_{J_kJ_k} + \mathrm{Diag}(\mu^r_{J_k})$, $\hat{b}^k(\mu) = \hat{b}^0_{J_k} + \sum_{i=1}^m \mu_i^q \hat{b}^i_{J_k}$, and $\hat{A}^i, \hat{b}^i$ are defined as in (4). A Lagrangian dual function related to $(\tilde{Q})$ is

$$\tilde{D}(\mu) = \hat{c}(\mu) + \sum_{k=1}^{p} \tilde{D}_k(\mu),$$

where $\hat{c}(\mu) = \hat{c}_0 + \sum_{i=1}^m \mu_i^q \hat{c}_i$, defining the eigenvalue optimization problem

$$(\tilde{D}) \qquad \begin{array}{l} \max \tilde{D}(\mu) \\ \text{s.t. } \mu \in \tilde{\mathcal{M}} \end{array}$$

with

$$\tilde{\mathcal{M}} = \{\mu = (\mu^q, \mu^r) \in I\!R^m \times I\!R^{n+p} \mid \mu^q_I \geq 0,\ \mu^r_C \geq 0\}.$$

A similar eigenvalue problem was used in [39] for solving the trust region problem and in [20] for unconstrained quadratic 0-1 programming.

Remark 1. Since $D(\mu) > -\infty$ if and only if $\nabla^2 L(\cdot;\mu)$ is positive semidefinite, the dual (D) contains a hidden semidefinite constraint. This implies that for all $\hat{\mu} \in \text{dom } D$ the function $L(\cdot;\hat{\mu})$ is a convex underestimator of q_0 over the feasible set of (Q).

Remark 2. Let $\beta(S)$ be the optimal value of the dual problem (D) 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 [34] it is shown

$$\lim_{k \to \infty} \beta(S_k) = \begin{cases} q_0(\hat{x}) & \text{if } \hat{x} \in \Omega \\ \infty & \text{else,} \end{cases}$$

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 [23].

3.2. Dual equivalence

Proposition 1. *The dual problems (D) and ($\tilde{D}$) have the same optimal value.*

Proof. Define the standardized partial Lagrangian $\hat{L}_k(x;\mu) = \langle x, \hat{A}^k(\mu)x\rangle + 2\langle \hat{b}^k(\mu), x\rangle - \langle \mu^r_{J_k}, e\rangle$ according to (5), and let

$$\hat{D}_k(\mu) = \inf_{x \in I\!B(n_k)} \hat{L}_k(x;\mu). \tag{6}$$

We denote by $e_{J_k} \in I\!R^n$ the characteristic vector of a partition element J_k defined by $e_{J_k,j} = \begin{cases} 1 & \text{for } j \in J_k \\ 0 & \text{else} \end{cases}$. From strong duality of the trust-region problem (see

(9) in the Appendix) it follows

$$\begin{aligned}\hat{D}_k(\mu) &= \min_{x\in I\!B(n_k)} \hat{L}_k(x;\mu)\\ &= \max_{t\in I\!R_+}\inf_{x\in I\!R^{n_k}} \hat{L}_k(x;\mu) + t\cdot(\|x\|^2 - n_k)\\ &= \max_{t\in I\!R_+}\inf_{x\in I\!R^{n_k}} \hat{L}_k(x;\mu^q,\mu^r + t\cdot e_{J_k})\end{aligned}$$

From Lemma 3 of the Appendix we have

$$\begin{aligned}\hat{D}_k(\mu) &= \min_{x\in I\!B(n_k)} \langle x, \hat{A}^k(\mu)x\rangle + 2\langle \hat{b}^k(\mu), x\rangle - \langle \mu^r_{J_k}, e\rangle\\ &= \max_{t\in I\!R}\inf_{x\in I\!R^{n_k+1}} \langle x, \tilde{A}^k(\mu^q,\mu^r + te_{n+k})x\rangle - \langle \mu^r_{\tilde{J}_k}, e\rangle\\ &= \max_{t\in I\!R} \tilde{D}_k(\mu^q,\mu^r + te_{n+k})\end{aligned}$$

Hence,

$$\begin{aligned}\mathrm{val}(D) &= \max_{\mu\in\mathcal{M}} c(\mu) + \sum_{k=1}^{p}\inf_{x\in I\!R^{n_k}} L_k(x;\mu)\\ &= \max_{\mu\in\mathcal{M}} \hat{c}(\mu) + \sum_{k=1}^{p}\max_{t\in I\!R_+}\inf_{x\in I\!R^{n_k}} \hat{L}_k(x;\mu^q,\mu^r + t\cdot e_{J_k})\\ &= \max_{\mu\in\mathcal{M}} \hat{c}(\mu) + \sum_{k=1}^{p}\hat{D}_k(\mu)\\ &= \max_{\mu\in\mathcal{M}} \hat{c}(\mu) + \sum_{k=1}^{p}\max_{t\in I\!R}\tilde{D}_k(\mu^q,\mu^r + te_{n+k})\\ &= \mathrm{val}(\tilde{D}).\end{aligned}$$

□

3.2.1. Modifications Several simplifications of the dual problem $(\tilde{D})$ are possible.

Remark 3. If all variables of a block J_k are binary, i.e. $J_k \subseteq B$, we can dualize the related partial Lagrangian function with respect to the sphere $\partial\mathbb{B}(n_k)$. This simplifies the dual problem $(\tilde{D})$ since the number of dual constraints is reduced. We show that this modification does not change val$(\tilde{D})$. To see this, we consider the modified partial dual function of $\tilde{D}$ defined by

$$\bar{D}_k(\mu) = (n_k + 1) \cdot \lambda_1(\tilde{A}^k(\mu)) - \langle \mu^r_{\tilde{J}_k}, e \rangle.$$

Since $\lambda_1(\tilde{A}^k(\mu^q, \mu^r + t \cdot e_{\tilde{J}_k})) = \lambda_1(\tilde{A}^k(\mu)) + t(n_k + 1)$ and $\langle (\mu^r + t \cdot e_{\tilde{J}_k})_{\tilde{J}_k}, e \rangle = \langle \mu^r_{\tilde{J}_k}, e \rangle + t(n_k + 1)$ for all $t \in \mathbb{R}$, it holds

$$\bar{D}_k(\mu) = \tilde{D}_k(\mu^q, \mu^r + t \cdot e_{\tilde{J}_k}).$$

For $t = \min\{0, -\lambda_1(\tilde{A}^k(\mu))\}$ we have $\lambda_1(\tilde{A}^k(\mu^q, \mu^r + t \cdot e_{\tilde{J}_k})) \geq 0$ and therefore $\bar{D}_k(\mu^q, \mu^r + t \cdot e_{\tilde{J}_k}) = \tilde{D}_k(\mu^q, \mu^r + t \cdot e_{\tilde{J}_k})$, which implies that val$(\tilde{D})$ is not changed.

Remark 4. A further simplification can be made in the case $b^i_{J_k} = 0$ for $i = 0, \ldots, m$. In this case, the trust region problem (6) is an eigenvalue problem and it holds

$$\hat{D}^k(\mu) = n_k \cdot \min\{0, \lambda_1(\hat{A}^k(\mu))\} - \langle \mu^r_{J_k}, e \rangle.$$

From Lemma 3 it follows that $\tilde{D}^k$ can be replaced by $\hat{D}^k$ without changing val$(\tilde{D})$.

Remark 5. If $A^i_{J_k J_k}$ is zero for $i = 0, \ldots, m$, the related Lagrangian problem is linear and therefore separable with respect to all variables of this block. Hence,

we can assume $J_k = \{j_k\}$, i.e. $I\!B(n_k) = [-1, 1]$. Then

$$\min_{x\in[-1,1]} \hat{L}_k(x;\mu) = \min_{x\in[-1,1]} 2\langle \hat{b}^k(\mu), x\rangle - \langle \mu^r_{J_k}, e\rangle = 2\min\{\hat{b}_{j_k}(\mu)\underline{x}_{j_k}, \hat{b}_{j_k}(\mu)\overline{x}_{j_k}\} - \langle \mu^r_{J_k}, e\rangle.$$

If (Q) is a MILP, this yields the traditional linear relaxation.

3.2.2. Influence of decomposition on the dual function Denote by $\tilde{D}^0$ the dual function $\tilde{D}$ of $(\tilde{Q})$ defined with respect to the trivial partition $\mathcal{P}^0 = \{V\}$ with $V = \{1, \ldots, n\}$. From Proposition 1 it follows that the optimal values related to $\tilde{D}^0$ and $\tilde{D}$ are the same. However, the dual values $\tilde{D}^0(\mu)$ and $\tilde{D}(\mu)$ at a dual point $\mu \in \tilde{\mathcal{M}}$ can be different. Let $\tilde{L}(x;\mu) = \hat{c}(\mu) + \sum_{k=1}^p \tilde{L}_k(x_{\tilde{J}_k};\mu)$ be the Lagrangian related to $(\tilde{Q})$ and $X = \{x \in I\!R^{n+p} \mid x_{\tilde{J}_k} \in I\!B(1+n_k),\ k = 1, \ldots, p\}$. Since $X \subseteq I\!B(n)$, we have

$$\tilde{D}^0(\mu) = \min_{x\in I\!B(n+p)} \tilde{L}(x;\mu) \le \min_{x\in X} \tilde{L}(x;\mu) = \tilde{D}(\mu).$$

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 $A^k \in I\!R^{(n_k,n_k)}$, $k = 1, \ldots, p$. Assuming $\lambda_1(A^1) < \lambda_1(A^j)$ for $j > 1$, it follows

$$\tilde{D}^0(0) = n \cdot \lambda_1(A) < \sum_{k=1}^p n_k \lambda_1(A^k) = \tilde{D}(0).$$

This demonstrates that decomposition not only facilitates the evaluation of the dual function, but also improves the initial dual bound $\tilde{D}(0)$ (see Section 5).

On the other hand, if a splitting scheme is used, decomposition can worsen the dual bound $\tilde{D}(0)$. In [29] it is shown:

Proposition 2. *Let (D) and (DS) be the Lagrangian dual of the original problem (Q) and the splitting scheme (S), as defined in Section 2.2, respectively. Then* $\mathrm{val}(DS) \leq \mathrm{val}(D)$.

The results of Section 5 demonstrate that this inequality can be strict.

4. Solving the Lagrangian dual problem ($\tilde{D}$)

The dual problem ($\tilde{D}$) is a convex non-differentiable optimization problem and can be solved by many methods (see [22]). We use the proximal bundle code NOA 3.0 [27] of Kiwiel described in [26] for maximizing the dual function $\tilde{D}$. Starting with a dual point $\mu^0 = 0$ the method produces at each so-called *serious step* a dual point $\mu^k \in \tilde{M}$ with $\tilde{D}(\mu^k) > \tilde{D}(\mu^{k-1})$.

At each trial point generated by the method the dual function $\tilde{D}$ and a related supergradient is evaluated. A supergradient $g \in I\!R^m$ of a concave function $D : I\!R^m \mapsto I\!R$ at a point $\mu \in I\!R^m$ satisfies $D(\mu) + \langle g, \lambda - \mu \rangle \geq D(\lambda)$ for all $\lambda \in I\!R^m$. A supergradient of a dual function can be obtained by evaluating constraint functions at a Lagrangian solution point. More precisely, it holds [22]

Lemma 1. *Let* $L(x;\mu) = f(x) + \langle h, \mu \rangle$ *be a continuous Lagrangian function related to an objective function* $f : I\!R^n \mapsto I\!R$ *and a constraint function* $h : I\!R^n \mapsto I\!R^m$ *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* $\mu \in \mathrm{dom}\ D$.

We apply this result to problem ($\tilde{D}$).

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

Proof. From the definition of x it follows $x_{\tilde{J}_k} \in \underset{y \in I\!B(n_k+1)}{\operatorname{Argmin}} \tilde{L}_k(y;\mu)$. Hence, $x \in \underset{y \in X}{\operatorname{Argmin}} \tilde{L}(y;\mu)$, where $\tilde{L}$ and X are defined as in Section 3.2.2. This proves the statement according to Lemma 1. □

We implemented the evaluation of the dual function $\tilde{D}$ with the modifications of Remarks 3, 4 and 5, and the supergradient formula of Lemma 2 in `C++`. For the computation of a minimum eigenvalue and minimum eigenvector we used two algorithms. The first algorithm is an implicit symmetric QL-method from the EISPACK-library [32], 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++ [18].

We used the following parameters of proximal bundle method NOA: bundle size = 50, linesearch decrease = 0.1, QP weight = 10.0 and feasibility tolerance = 0.1. As a stopping criterion for the bundle method we set either the optimality tolerance equal to 10^{-3}, or we stop the method if a *measure of relative improvement* is smaller than a given tolerance. In particular, we define

$$\delta_s^j = \frac{\tilde{D}(\mu^{s(j+1)}) - \tilde{D}(\mu^{sj})}{|\tilde{D}(\mu^0)| + 1},$$

and stop the iteration, if

$$\delta_s^j < \rho \cdot \delta_s^{j_{max}} \tag{7}$$

where $\{\mu^j\}$ is the sequence of dual points generated by the bundle method at serious steps, $\delta_s^{j_{max}} = \max\{\delta_s^0, \ldots, \delta_s^j\}$, with $\rho = 0.4$ and $s = 10$.

5. Preliminary numerical results

In order to study the influence of decomposition, we made numerical experiments with random MIQQP instances. All results were obtained on a machine that has a 1.8 GHz-Pentium IV processor on a LINUX system.

5.1. Block structure

In the first experiment we compared decomposition-based bounds computed by the QL-method and non-decomposition-based bounds computed by the full-dimensional Lanczos method. We produced block-separable random MIQQPs using the following procedure with parameters n, the number of variables, m, the number of quadratic constraints, and l, the block size.

Procedure `rand_miqqp`(n, m, l)

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

6. set $c_i = 0$ for $i = 0, \ldots, m$.

The above procedure produces a MIQQP, which is block-separable with respect to the blocks $J_k = \{(k-1)l + 1, \ldots, kl\}$, $k = 1, \ldots, p$. Since $c_i = 0$ for $i = 0, \ldots, m$, $x = 0$ is a feasible point. Therefore, the dual function $\tilde{D}$ is bounded. For a given set of input parameters (n, m, l) we produced 5 random MIQQPs with the procedure `rand_miqqp`.

For each instance we generated two dual problems of the form $(\tilde{D})$ related to the partitions $\mathcal{P}_1 = \{J_1, \ldots, J_p\}$ and $\mathcal{P}_2 = \{V\}$. The first dual problem is called (D_1) and the second (D_2). The corresponding dual functions are denoted by D_1 and D_2 respectively. From Proposition 1 we know that $\mathrm{val}(D_1) = \mathrm{val}(D_2)$.

We first computed a dual value $D_1(\hat{\mu})$ using the previously described bundle method with the stopping criterion (7). Then we maximized D_2, and stopped the iteration if D_2 reached the value $D_1(\hat{\mu})$. Furthermore, we calculated the initial relative error

$$\kappa_i^0 = \frac{\mathrm{val}(D_2) - D_i(0)}{|\,\mathrm{val}(D_2)| + 1}, \quad i \in \{1, 2\}, \tag{8}$$

where the optimal value of (D_2) was computed using the previously described bundle method with an optimality tolerance 10^{-3}. Tables 1 and 2 show

- the fraction t_2/t_1 where t_1 and t_2 is the average time in seconds for solving (D_1) and (D_2) respectively,
- the time t_1
- the fraction κ_2^0/κ_1^0 where κ_i^0 is the average initial relative error (8)

for different input parameters of `rand_miqqp`.

	block-size $l = 10$			block-size $l = 20$		
n	t_2/t_1	t_1	κ_2^0/κ_1^0	t_2/t_1	t_1	κ_2^0/κ_1^0
200	312.526	0.392	7.22114	85.7879	0.594	5.81284
400	1544.22	0.768	10.5006	271.037	1.234	8.79377
600	3551.09	1.204	12.8053	563.391	1.818	11.3668
800	4243.39	1.656	15.5317	861.217	2.428	12.9469
1000	6546.61	2.068	17.3226	1279.55	3.226	14.7185

Table 1. number of quadratic constr. $m = 0$

	block-size $l = 10$			block-size $l = 20$		
m	t_2/t_1	t_1	κ_2^0/κ_1^0	t_2/t_1	t_1	κ_2^0/κ_1^0
0	53.7087	0.206	4.63817	21.9728	0.294	3.72246
4	159.35	0.24	4.84415	38.9673	0.428	3.6699
8	135.229	0.376	4.52294	37.0607	0.626	3.41876
12	132.924	0.472	4.40023	29.1492	0.764	3.51218
16	157.272	0.766	4.33168	47.5457	1.378	3.4816
20	166.995	0.85	4.19541	56.2844	1.568	3.44

Table 2. dimension $n = 200$

It can be seen from the tables that the decomposition scheme accelerates the running time by magnitudes. 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. Moreover, the results show that $\kappa_1^0 < \kappa_2^0$ (see also Table 1).

Decomposition also makes the dual solution method more stable. We observed convergence problems of the Lanczos method if the optimality tolerance of the dual solver is small. It is well-known that the performance of the Lanczos

method depends highly on the separability of the eigenvalues, and eigenvalues cluster in eigenvalue optimization (see also [21]). In contrast, the QL-method is very stable.

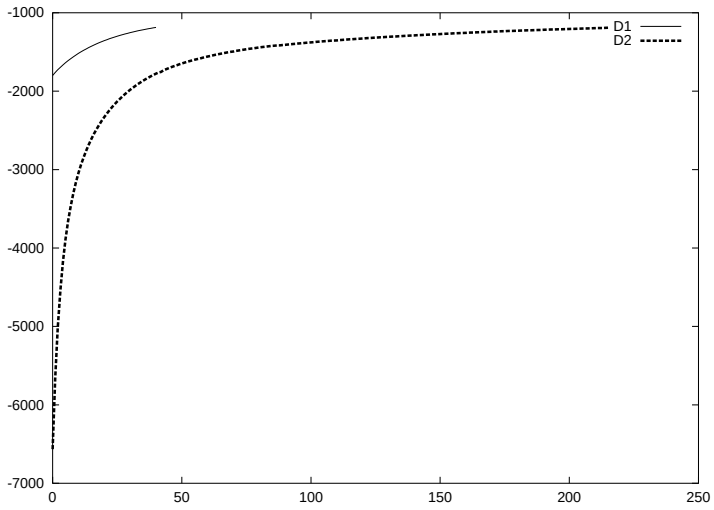

Fig. 1. Dual values of D_1 and D_2 at serious steps where $(n, m, l) = (200, 0, 10)$

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

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

The submatrices $A^k \in I\!R^{(l,l)}$, $k = 1, \ldots, p$, are dense with a block-size $l = n/p$. The submatrices $B^k \in I\!R^{(l,l)}$ are sparse with nonzero entries at $(l-i, i)$, $i = 1, \ldots, s$, where $s \in \{0, \ldots, l\}$ is a given flow size. The resulting sparsity graph has a ring topology with p 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 previous section using the partition $\mathcal{P} = \{J_1, \ldots, J_p\}$ with $J_k = \{(k-1)l + 1, \ldots, k \cdot l\}$, $k = 1, \ldots p$.

For the splitting scheme as well as for the original Max-Cut problem we constructed a dual problem of the form $(\tilde{D})$, which we call (D_1) and (D_2) respectively. As in the previous experiment, we produced 5 random Max-Cut problems for a given set of input parameters (n, p, s) and computed first a dual value of $D_1(\hat{\mu})$ and then a dual value $D_2(\tilde{\mu}) \simeq D_1(\hat{\mu})$ using the bundle method NOA with the parameters previously described. Tables 3 and 4 show

- the fraction t_2/t_1 where t_1 and t_2 is the average time in seconds for solving (D_1) and (D_2) respectively,
- the time t_1
- the fraction κ_2^0/κ_1^0 where κ_i^0 is the average initial relative error (8)

- the average percentage relative difference of the optimal dual values of (D_1) and (D_2)

$$\kappa_d = \frac{\text{val}(D_2) - \text{val}(D_1)}{|\,\text{val}(D_2)| + 1}.$$

The results demonstrate that the splitting scheme accelerates the evaluation of the dual in most cases considerably. However, in the last experiment the computation without decomposition was faster. It can also be seen that for these instances the relative difference of the optimal dual values κ_d is not zero (see Section 3.2.2). Moreover, the fraction κ_2^0/κ_1^0 was for $s = 2$ in most cases greater one, and for $s = 4$ smaller than one (see also Table 2).

	block-size $l = 10$				block-size $l = 20$			
n	t_2/t_1	t_1	κ_2^0/κ_1^0	$100 \cdot \kappa_d$	t_2/t_1	t_1	κ_2^0/κ_1^0	$100 \cdot \kappa_d$
200	8.21244	2.702	0.969418	0.468361	2.7542	2.026	1.30485	6.36203
400	7.18263	6.264	1.00836	0.953826	4.39145	5.288	1.48307	6.71984
600	6.98609	12.506	1.22847	0.827309	3.53643	8.426	1.64889	7.52866
800	6.96305	20.246	1.23818	0.627922	4.74037	12.826	1.69921	7.20949
1000	9.21404	29.322	1.19733	0.601579	5.22766	16.876	1.6948	7.33724

Table 3. flow size $s = 2$

6. Concluding Remarks

We presented a Lagrangian decomposition method for solving the dual of a block-separable MIQQP via eigenvalue computation. Preliminary numerical experiments on general block-separable MIQQPs and on non-separable network-MaxCut problems demonstrate that a decomposition-based evaluation of the

	block-size $l = 10$				block-size $l = 20$			
n	t_2/t_1	t_1	κ_2^0/κ_1^0	$100 \cdot \kappa_d$	t_2/t_1	t_1	κ_2^0/κ_1^0	$100 \cdot \kappa_d$
200	1.92812	3.756	0.634217	2.18598	0.256369	7.38	0.485861	0.801154
400	1.97711	11.532	0.711109	3.39876	0.463195	18.394	0.434094	2.24296
600	1.98623	22.364	0.755965	3.72389	0.441957	34.216	0.57898	1.94131
800	2.26108	36.732	0.892395	3.60844	0.51311	52.098	0.614649	3.39037
1000	2.10734	56.102	0.72452	3.69923	0.376503	73.864	0.539808	2.22469

Table 4. flow-size $s = 4$

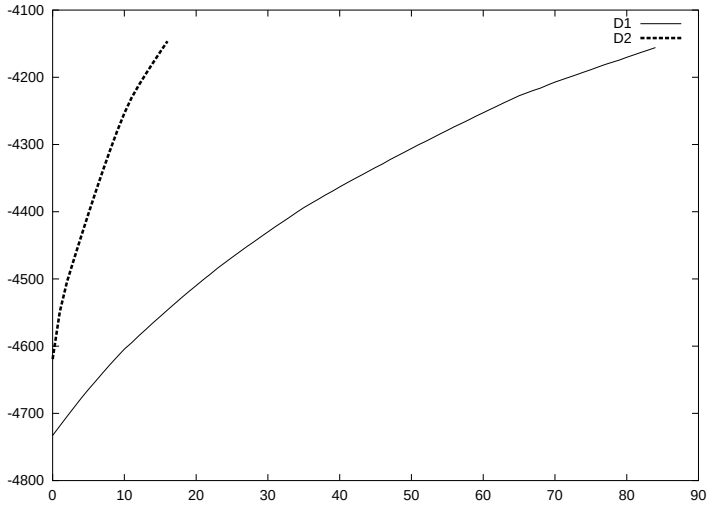

Fig. 2. Dual values of D_1 and D_2 at serious steps where $(n, s, l) = (100, 2, 10)$

dual function of a splitting-scheme using the symmetric QL-algorithm may be much faster and more stable than evaluating the dual function of the original problem using the full-dimensional Lanczos algorithm.

If the given problem is non-separable, a splitting-scheme usually increases the duality gap. However, it is possible that nevertheless the initial value of the dual function of the splitting scheme can be greater than the initial value of

the orginal dual function. This was demonstrated for network-MaxCut problems with a small flow size.

The artificial MIQQP instances in our experiments are designed to study the effect of decomposition-based bounds. It would be interesting to test the performance of the presented eigenvalue bound using real-world MIQQPs. For example, the MINLPLib [7] contains many block-separable MIQQPs.

It is well-known that the duality gap of eigenvalue bounds can be quite large. The proposed approach offers the possibility to improve a dual bound by shifting (strongly) violated binary constraints into a Lagrangian sub-problem. Provided that the dimension of the sub-problem is small, the resulting unconstrained binary problem can be solved quickly using a branch-and-bound method.

Lagrangian decomposition provides dual bounds on the optimal value, which can be used in a branch-and-bound (B&B) algorithm. Under mild assumptions, it can be shown that B&B-algorithms using dual bounds are convergent (see [13] for continuous optimization problems with compact feasible sets and [34] for all-quadratic problems). Furthermore, Lagrangian decomposition provides a convex underestimator of the objective function over the feasible set, which can be used to compute primal solutions via a relaxation-based heuristic (see[4],[35]).

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. Furthermore, we want to thank two anonymous referees for valuable comments which helped to improve the text.

Appendix

We describe some results on the trust-region problem needed for the proof of Proposition 1. A trust-region problem is defined by

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

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.

$$\text{val}(T1) = \text{val}(DT1) \tag{9}$$

where val$(T1)$ and val$(DT1)$ are the optimal values of (T1) and (DT1) respectively (see [42]). If $b = 0$, then (T1) is an eigenvalue problem and it holds 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

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

Clearly, we have val$(T1)$ = val$(T2)$. Dualization of (T2) with respect to the ball $I\!B(n+1)$ gives 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* $\mathrm{val}(T2) = \mathrm{val}(T1) = \mathrm{val}(DT2)$.

Proof. This was proved in [39]. We repeat the proof in oder to keep the paper self-contained.

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

□

References

1. H.D. Sherali W.P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Kluwer Academic Publishers, 1999.
2. 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.
3. 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.

4. H. Alperin and I. Nowak. Lagrangian Smoothing Heuristics for MaxCut. Technical report, HU–Berlin NR–2002–6, 2002.
5. 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.
6. 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.
7. M.R. Bussieck, A.S. Drud, and A. Meeraus. MINLPLib - A Collection of Test Models for Mixed-Iinteger Nonlinear Programming. *INFORMS J. Comput.*, 15(1), 2003.
8. P. Chardaire and A. Sutter. A decomposition method for quadratic zero-one programming. *Management Science*, 41:704–712, 1995.
9. J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. 2nd edn, Springer, New York, 1993.
10. G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.
11. E. V. Demands and C. S. Tang. Linear control of a Markov production system. *Operations Research*, 40:259–278, 1992.
12. 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.
13. M. Dür. Dual bounding procedures lead to convergent Branch-and-Bound algorithms. *Math. Progr.*, 91:117–125, 2001.
14. M. C. Ferris and J. D. Horn. Partitioning mathematical programs for parallel solution. *Math. Progr.*, 80:35–61, 1998.
15. J. A. Filar and T. A. Schultz. Bilinear programming and structured stochastic games. *J. Opt. Theor. Appl.*, 53:85–104, 1999.
16. K. Fujisawa, M. Kojima, and K. Nakata. Exploiting Sparsity in Primal-Dual Interior-Point Methods for Semidefinite Programming. *Math. Progr.*, 79:235–254, 1997.
17. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York, 1979.
18. F. Gomes and D. Sorensen. `ARPACK++`*: a* `C++` *Implementation of* `ARPACK` *eigenvalue package*, 1997. `http://www.crpc.rice.edu/software/ARPACK/`.

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

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

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

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

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

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

25. 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.

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

27. 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.

28. M. Kojima and L. Tunçel. Discretization and localization in successive convex relaxation methods for nonconvex quadratic optimization problems. *Math. Programming*, 89:97–111, 2000.

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

30. A. Martin. Integer programs with block structure. Technical report, ZIB–Report 99–03, Habilitationsschrift, 1999.

31. Yuri Nesterov, Henry Wolkowicz, and Yinyu Ye. Semidefinite programming relaxaxations of nonconvex quadratic optimization. In H. Wolkowicz, R. Saigal, and L. Vandenberghe,

editors, *Handbook of Semidefinite Programming*, pages 361–419. Kluwer Academic Publishers, 2000.

32. NETLIB. *EISPACK*. http://www.netlib.org/eispack/, 1972-1973.
33. 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.
34. I. Nowak. Dual bounds and optimality cuts for all-quadratic programs with convex constraints. *J. Glob. Opt.*, 18:337–356, 2000.
35. I. Nowak, H. Alperin, and S. Vigerske. LaGO - an object oriented library for solving MINLPs. *to appear in Lecture Notes on Computer Science*, 2003. available at http://www.mathematik.hu-berlin.de/∼eopt/papers/LaGO.pdf.
36. 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.
37. 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.
38. S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for (0,1)-quadratic programming. *J. Global Optim.*, 7(1):51–73, 1995.
39. 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.
40. A. Ruszczyński. Decomposition methods in stochastic programming. *Math. Progr.*, 79:333–353, 1997.
41. D. P. Rutenberg and T. L. Shaftel. Product design: Sub-assemblies for multiple markets. *Management Science*, 18:B220–B231, 1971.
42. R. Stern and H. Wolkowicz. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optimization*, 5(2):286–313, 1995.
43. N. Thoai. Duality bound method for the general quadratic programming problem with quadratic constraints. *Journal of Optimization Theory and Applications*, 107 (2), 2000.

44. 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.

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