Rank-Sparsity Decomposition of a Positive Semidefinite Matrix

Marcia Fampa
Instituto de Matemática and PESC/COPPE, Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
fampa@cos.ufrj.br

Jon Lee
IOE Department, University of Michigan
Ann Arbor, Michigan, USA
jonxlee@umich.edu

Sandra A. Santos
IMECC-UNICAMP, University of Campinas
Campinas SP, Brazil
sandra@ime.unicamp.br

ABSTRACT

We investigate the problem of decomposing a given (symmetric) positive semidefinite matrix R as $R = S + L$, where S is sparse, L has low rank, and both are (symmetric) positive semidefinite. Additionally, we enforce that S and L belong to convex and compact sets of matrices, specified via linear matrix inequalities. Our goal is to minimize a weighted sum of the sparsity of S and the rank of L. Aiming at the global optimization of this nonconvex problem, we investigate convex relaxations of its "simplified" version, where the rank of L is fixed and we seek rather to minimize the sparsity of S. Following ideas from the literature, we discuss how strong lower bounds for this simplified problem can be used to increase the ability of a branch-and-bound method to find global optimal solutions for the original nonconvex problem. A heuristic procedure is also proposed to construct a feasible solution for the original problem, taking the solution of our relaxation as a starting point. Preliminary numerical results are shown for randomly generated problems.

KEYWORDS. rank-sparsity decomposition, convex relaxation, semidefinite programming.

PM Mathematical programming

1. Introduction

Given an $n \times n$ symmetric positive semidefinite matrix R, we consider

$$\min\{\gamma\|S\|_0 + \mathrm{r}(L) \,:\, S + L = R,\ S \succeq 0,\ L \succeq 0,\ S \in \mathcal{S},\ L \in \mathcal{L}\}\,. \tag{1}$$

where $B \succeq 0$ indicates that the matrix B is symmetric and positive semidefinite, the *sparsity* $\|S\|_0$ is the number of nonzeros of S, and $\mathrm{r}(L)$ is the rank of L. The constant $\gamma > 0$ is simply used to trade off between the two objectives, sparsity and rank. $\mathcal{S}$ and $\mathcal{L}$ are convex and compact sets of matrices, specified via linear matrix inequalities.

Our interest in the problem stems from a difference of convex functions ("DC") approach to the constrained global minimization of an indefinite quadratic form $\frac{1}{2}x^tQx$, where Q is symmetric but indefinite; see [Fampa et al., 2017]. In such an approach, Q is decomposed as $Q = P - R$, where P and R are (symmetric) positive semidefinite. In [Fampa et al., 2017], we concentrated on the having R "small" in some sense, as that part captures the nonconvexity which is the hard part for global minimization. Now, the nonconvex part $-\frac{1}{2}x^tRx$ can be attacked by a variety of lifting methods. In particular, there are:

- methods that treat the nonconvexity termwise. That is, Reformulation Linearization Technique (RLT) approaches (see [Sherali and Tuncbilek, 1995; Sherali and Adams, 1999]) which concentrate on viewing $-\frac{1}{2}x^tRx$ termwise, that is as $-\sum_{i<j} r_{ij}x_ix_j$ - $\frac{1}{2}\sum_i r_{ii}x_i^2$, and then attacking the nonconvex terms (ie., the ones with $r_{ij} > 0$ all of the univariate terms) via RLT convexification methods.

- methods that treat the convexity holistically. Specifically, the approach of [Saxena et al., 2010, 2011], which dynamically applies disjunctive programming on directions of nonconvexity. In some sense, [Fampa et al., 2017] follows some of the spirit of [Saxena et al., 2010], but in a computationally less intensive manner, where the nonconvexity is isolated to the $\mathrm{r}(R)$ directions of the eigenvectors of R.

In recent software implementations (e.g., Cplex), one or the other of these two approaches is pursued (i.e., the RLT termwise approach or the [Fampa et al., 2017] holistic approach). There are also possibilities to combine these ideas, by further splitting R as $R = S + L$, where S is sparse, L has low rank, and both are (symmetric) positive semidefinite. By doing so:

- $-\frac{1}{2}x^tSx$ is a *sparse* nonconvex quadratric form, and is particularly well suited to the RLT approach.

- $-\frac{1}{2}x^tLx$ is a *low-rank* nonconvex quadratric form, and is particularly well suited to the treatment of [Fampa et al., 2017] which here would do spatial branching on the only $\mathrm{r}(L)$ directions of nonconvexity.

We have thus established our motivation for considering (1).

2. Simplified problem

In this section, we consider a "simplified" version of problem (1), where we minimize the sparsity of S, with the rank of L fixed to r. The motivation for considering this version of the problem comes from the fact that we can globally solve problem (1), by solving it for all ranks r. In this way, we do not need to branch on the rank, in a branch-and-bound algorithm for (1), but only on sparsity, which could be done more easily, for example, as described in [Lee and Zou, 2014].

The simplified problem can be formulated as:

$$\min\{\|S\|_0 : S + L = R,\ \mathrm{r}(L) = r,\ S \succeq 0,\ L \succeq 0,\ S \in \mathcal{S},\ L \in \mathcal{L}\}\,. \tag{2}$$

Theorem 1 *Problems* (1) *and* (2) *are NP-Hard.*

Proof We begin with positive integers $b, a_1, a_2, \ldots, a_n$ defining the NP-Complete *exact-knapsack decision problem*: Is there a solution in 0/1 variables to $\sum_{j=1}^{n} a_j x_j = b$? We will proceed to define instance of (1) and (2) that solve this decision problem.

Let $R := I_{2n}$. We use the sets $\mathcal{S}$ and $\mathcal{L}$ to enforce that S and L are diagonal matrices of particular forms. Specifically, we enforce that S has the form

$$S = \text{Diag}(s_{11}, s_{22}, \ldots, s_{nn}, 1 - s_{11}, 1 - s_{22}, \ldots, 1 - s_{nn}) .$$

Then, because $S + L = R$, we have that

$$L = \text{Diag}(1 - s_{11}, 1 - s_{22}, \ldots, 1 - s_{nn}, s_{11}, s_{22}, \ldots, s_{nn}) .$$

With $\mathcal{S}$, we also include $\sum_{j=1}^{n} a_j s_{jj} = b$.

Now, it is easy to see that all solutions of the resulting (2) have $||S||_0 \geq n$, and the only ones having $||S||_0 = n$ are the solutions with $s_{jj} \in \{0, 1\}$, for $j = 1, 2, \ldots, n$. We interpret such a solution as a solution of our knapsack problem via $x_j := s_{jj}$, for $j = 1, 2, \ldots, n$. So we solve the exact-knapsack decision problem by instead solving an instance of (2). It is now easy to see that choosing $\gamma \geq 2n$, we can as well solve the exact-knapsack decision problem by instead solving an instance of (1). The result follows. □

As (2) is a nonconvex NP-Hard problem, our ability to solve instances of it to optimality is highly dependent on the knowledge of good lower bounds for its optimal solution value. Our initial goal in this work is, therefore, to construct strong convex relaxations for (2).

With this purpose in mind, we first underestimate $\|S\|_0$ with $\|S\|_1/\alpha$, where $\|S\|_1 := \sum_i \sum_j |S_{ij}|$ and $\alpha > 0$. Concerning the value of α, as in [Lee and Zou, 2014], we assume that $\|S\|_{\max} := \max\{|S_{ij}| : i, j = 1, \ldots, n\}$ is bounded on $\mathcal{S}$, and therefore, it is possible to calculate a scalar $\alpha > 0$, such that $\|S\|_{\max} \leq \alpha$, for all $S \in \mathcal{S}$. For this scalar α, we may use $\|S\|_1/\alpha$ to underestimate $\|S\|_0$.

Next, we represent L as a sum of r rank-one matrices, replacing L with $\sum_{k=1}^{r} X^k$, where $X^k := x^k {x^k}^t$ and $x^k \in \mathbb{R}^n$, for $k = 1, \ldots, r$. An initial (nonconvex) relaxation of (2) is then formulated as:

$$\begin{array}{ll} \min_{S,L,Z,X^1,\ldots,X^r,x^1,\ldots,x^r} & \frac{1}{\alpha} e^t Z e \ , \\ \text{subject to} & \\ -Z \leq S \leq Z, \ S + L = R, \ L = \sum_{k=1}^{r} X^k, \ S \succeq 0, & \\ X^k = x^k {x^k}^t, \ x^k \in \mathbb{R}^n, \ k = 1, \ldots, r, & \\ S \in \mathcal{S}, \ L \in \mathcal{L}, & \end{array} \quad (3)$$

where e is the n-vector of all ones. Note that $e^t Z e = \sum_i \sum_j Z_{ij}$, which is equal to $\|S\|_1$ due to the minimization.

Convex relaxations for the nonconvex equations $X^k = x^k {x^k}^t$ in (3) have been extensively studied in the literature. These relaxations mostly use Semidefinite Programming (SDP) [Nesterov, 1998] and the Reformulation-Linearization Technique (RLT) [Sherali and Adams, 1999]. In [Anstreicher, 2009], the author also shows on selected test problems, that the combined use of SDP and RLT can produce bounds that are substantially better than either technique used alone. We discuss in the following, the application of these approaches to obtain lower bounds for (3). The use of secant underestimation and disjunctive programming to generate valid inequalities for the relaxations, as introduced in [Saxena et al., 2010] and [Lee and Rendl, 2008], is also investigated.

2.1. Semidefinite Programming (SDP) relaxation

When applying SDP to generate lower bounds to (3), the idea is to relax the nonconvex constraints $X^k = x^k {x^k}^t$ to $X^k - x^k {x^k}^t \succeq 0$. Using the well known result

$$X^k - x^k {x^k}^t \succeq 0 \Leftrightarrow \begin{pmatrix} 1 & {x^k}^t \\ x^k & X^k \end{pmatrix} \succeq 0$$

on the Schur complement of X^k [Zhang, 2005, Theorem 1.12], the following *linear* SDP relaxation is produced for (3):

$$\begin{array}{l} \min_{S,Z,X^1,\dots,X^r,x^1,\dots,x^r} \ \frac{1}{\alpha} e^t Z e \ , \\ \text{subject to} \\ -Z \le S \le Z, \ S + \sum_{k=1}^r X^k = R, \ S \succeq 0, \\ \begin{pmatrix} 1 & {x^k}^t \\ x^k & X^k \end{pmatrix} \succeq 0, \ k = 1,\dots,r, \\ x^k \in \mathbb{R}^n, \ k = 1,\dots,r, \\ S \in \mathcal{S}, \ L \in \mathcal{L}. \end{array} \tag{4}$$

2.2. Valid inequalities produced by the Reformulation Linearization Technique

The Reformulation Linearization Technique (RLT) was introduced in [Sherali and Adams, 1999] and can be used to generate valid inequalities to (3). The idea is to replace the nonconvex constraints $X^k - x^k {x^k}^t = 0$ in (3) by linear inequalities derived from lower and upper bounds on the variables. Considering that

$$l^k \le x^k \le u^k, \tag{5}$$

for given vectors $l^k, u^k \in \mathbb{R}^n$, we multiply each pair of the four inequalities $x_i^k - l_i^k \ge 0$, $u_i^k - x_i^k \ge 0$, $x_j^k - l_j^k \ge 0$, $u_j^k - x_j^k \ge 0$ for all $i, j = 1,\dots,n$. Defining then, $X_{ij}^k := x_i^k x_j^k$, we obtain the following RLT inequalities

$$\begin{array}{l} X_{ij}^k - l_i^k x_j^k - u_j^k x_i^k + l_i^k u_j^k \le 0, \\ X_{ij}^k - l_j^k x_i^k - u_i^k x_j^k + l_j^k u_i^k \le 0, \\ X_{ij}^k - l_j^k x_i^k - l_i^k x_j^k + l_j^k l_i^k \ge 0, \\ X_{ij}^k - u_j^k x_i^k - u_i^k x_j^k + u_j^k u_i^k \ge 0, \end{array} \tag{6}$$

for all $k = 1,\dots,r$, and $i, j = 1,\dots,n$.

Considering that S and X^k, for $k = 1,\dots,r$, are positive semidefinite, the diagonal elements of these matrices are nonnegative. Consequently, for all $i = 1,\dots,n$, we have $X_{ii}^k \le R_{ii}$ and $-\sqrt{R_{ii}} \le x_i^k \le \sqrt{R_{ii}}$. It is then possible to set

$$\begin{array}{l} l_i^k := -\sqrt{R_{ii}}, \\ u_i^k := \sqrt{R_{ii}}, \end{array} \tag{7}$$

for all $k = 1,\dots,r$ and $i = 1,\dots,n$.

We note that the inequalities (6) were introduced by McCormick in [McCormick, 1976]. It has been proved in [Al-Khayyal and Falk, 1983] that they represent the convex envelope of

$$\{(x_i^k, x_j^k, X_{ij}^k) \in \mathbb{R}^3 : \begin{array}{l} l_i^k \le x_i^k \le u_i^k, \\ l_j^k \le x_j^k \le u_j^k, \\ X_{ij}^k = x_i^k x_j^k \}. \end{array}$$

Considering (7), we can add the valid inequalities (6) and the box constraints (5) to (4) to obtain a stronger relaxation of (3). We will refer to this relaxation as (SDP+RLT).

2.3. Secant valid inequalities

Applying the ideas in [Saxena et al., 2010], we now intend to develop a convex relaxation of the nonconvex constraint

$$X^k - x^k {x^k}^t \preceq 0, \tag{8}$$

to better approximate the identity $X^k = x^k {x^k}^t$, for each $k = 1,\dots,r$.

The constraint (8) could equivalently be modeled by the infinite number of inequalities ${c^k}^t (X^k - x^k {x^k}^t) c^k \le 0$, or

$$({c^k}^t x^k)^2 \ge (c^k {c^k}^t) \bullet X^k, \tag{9}$$

for all $c^k \in \mathbb{R}^n$, where " $\bullet$ " is the usual inner product of matrices: $A \bullet B := \sum_{i,j} A_{ij}B_{ij} = \mathrm{Tr}(A^t B)$. The nonconvex inequality (9) is therefore a valid inequality for the SDP relaxation (4), for any choice of the vector c^k. It should not be added to (4), though, otherwise we would lose the convexity of our relaxation. [Saxena et al., 2010] point out that it is possible to convexify (9) by replacing the concave quadratic function $-({c^k}^t x^k)^2$ with its secant in an interval $[\eta_L(c^k), \eta_U(c^k)]$. The convex relaxation of (9) is then given by the linear secant inequality

$$({c^k}^t x^k)(\eta_L(c^k) + \eta_U(c^k)) - \eta_L(c^k)\eta_U(c^k) \geq (c^k {c^k}^t) \bullet X^k. \tag{10}$$

The interval $[\eta_L(c^k), \eta_U(c^k)]$ represents the range of the linear function ${c^k}^t x^k$ in the feasible set of (4), and can be computed by solving the two optimization problem, where we minimize and maximize ${c^k}^t x^k$, subject to the constraints in (4).

Finally, [Saxena et al., 2010] also note that if $\bar{X}^k$ and $\bar{x}^k$ are obtained from the solution of the relaxation (4), and $\bar{X}^k \neq (\bar{x}^k)(\bar{x}^k)^t$, then $\bar{X}^k - (\bar{x}^k)(\bar{x}^k)^t$ has at least one positive eigenvalue. Furthermore, if the vector c^k is chosen as the unit-length eigenvector corresponding to any positive eigenvalue of this matrix, then the constraint (9) would be violated by the solution of the relaxation. This observation guides the choice of the vector c^k in the inequalities.

2.4. Disjunctive programming over the SDP relaxation

Other valid inequalities to (4) can be obtained with SDP disjunctive cuts, which were proposed in [Lee and Rendl, 2008] and are based on the polyhedral disjunctive cuts introduced in [Saxena et al., 2010]. The idea is to divide the interval $[\eta_L(c^k), \eta_U(c^k)]$ into ν intervals $[\eta_{\delta,k}, \eta_{\delta+1,k}]$, for $\delta = 1, \dots, \nu$, such that $\eta_L(c^k) := \eta_{1,k} < \eta_{2,k} < \dots < \eta_{\nu,k} < \eta_{\nu+1,k} := \eta_U(c^k)$.

For each interval, we have the secant inequality

$$({c^k}^t x^k)(\eta_{\delta,k} + \eta_{\delta+1,k}) - \eta_{\delta,k}\eta_{\delta+1,k} \geq (c^k {c^k}^t) \bullet X^k, \tag{11}$$

which is valid for ${c^k}^t x^k \in [\eta_{\delta,k}, \eta_{\delta+1,k}]$.

Let

$$E^{ii} := \begin{array}{cc} & \begin{array}{ccc} & i & \end{array} \\ \begin{array}{c} \\ i \\ \\ \end{array} & \begin{pmatrix} & & \\ & 1 & \\ & & \end{pmatrix} \end{array}, \quad E^{ij(i \neq j)} := \begin{array}{cc} & \begin{array}{cccc} & i & j & \end{array} \\ \begin{array}{c} i \\ \\ j \end{array} & \begin{pmatrix} & & \frac{1}{2} & \\ & & & \\ & \frac{1}{2} & & \end{pmatrix} \end{array} \in \mathbb{R}^{n \times n},$$

$$E^{ii+} := \begin{array}{cc} & \begin{array}{ccc} & i+1 & \end{array} \\ \begin{array}{c} i+1 \\ \\ \\ \end{array} & \begin{pmatrix} & 1 & \\ & & \\ & & \end{pmatrix} \end{array}, \quad E^{ij+(i \neq j)} := \begin{array}{cc} & \begin{array}{cccc} & i+1 & j+1 & \end{array} \\ \begin{array}{c} i+1 \\ \\ j+1 \end{array} & \begin{pmatrix} & & \frac{1}{2} & \\ & & & \\ & \frac{1}{2} & & \end{pmatrix} \end{array} \in \mathbb{R}^{(n+1) \times (n+1)},$$

$$\bar{E}^{+} := \begin{array}{cc} & 1 \\ 1 & \begin{pmatrix} 1 & \\ & \end{pmatrix} \end{array} \in \mathbb{R}^{(n+1) \times (n+1)}, \quad \bar{E}^{i+} := \begin{pmatrix} 0 & \frac{1}{2}e_i^t \\ \frac{1}{2}e_i & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}.$$

Problem (4) can then be recast as:

$$\begin{array}{ll}
\min_{S,Z,B_1,\dots,B_r,X_1,\dots,X_r} \; ((ee^t) \bullet Z) & \\
\text{subject to} & \\
E^{ij} \bullet S + \sum_{k=1}^{r} E^{ij} \bullet X^k = E^{ij} \bullet R, & i,j = 1,\dots,n, \\
E^{ij} \bullet S + E^{ij} \bullet Z \geq 0, & i,j = 1,\dots,n, \\
-E^{ij} \bullet S + E^{ij} \bullet Z \geq 0, & i,j = 1,\dots,n, \\
E^{ij+} \bullet B^k - E^{ij} \bullet X^k = 0, & i,j = 1,\dots,n, k = 1,\dots,r, \\
\bar{E}^{+} \bullet B^k = 1, & k = 1,\dots,r, \\
\bar{E}_1^{+} \bullet B^k \geq 0, & k = 1,\dots,r, \\
S, B^k \succeq 0, & k = 1,\dots,r,
\end{array} \tag{12}$$

where

$$B^k := \begin{pmatrix} 1 & {x^k}^t \\ x^k & X^k \end{pmatrix}.$$

Note that for brevity, the constraints $S \in \mathcal{S}$ and $L \in \mathcal{L}$, and also the valid inequalities developed in Subsections 2.2 and 2.3, are omitted in this subsection. However they can be included in (12), to lead to stronger SDP disjunctive cuts.

For each $\delta = 1, \dots, \nu$, the secant inequalities

$$\begin{array}{c} \eta_{\delta,k} \leq {c^k}^t x^k \leq \eta_{\delta+1,k} \\ ({c^k}^t x^k)(\eta_{\delta,k} + \eta_{\delta+1,k}) - \eta_{\delta,k}\eta_{\delta+1,k} \geq (c^k {c^k}^t) \bullet X^k \end{array} \tag{13}$$

can also be recast as

$$\begin{array}{r} -\tilde{C}^k \bullet B^k \geq -\eta_{\delta+1,k}, \\ \tilde{C}^k \bullet B^k \geq \eta_{\delta,k}, \\ -\tilde{D}_{\delta,k} \bullet B^k \geq \eta_{\delta,k}\eta_{\delta+1,k}, \end{array} \tag{14}$$

where

$$\tilde{C}^k := \begin{pmatrix} 0 & \frac{1}{2}{c^k}^t \\ \frac{1}{2}c^k & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)},$$

and

$$\tilde{D}_{\delta,k} := \begin{pmatrix} 0 & \mathbf{0}^t \\ \mathbf{0} & c^k {c^k}^t \end{pmatrix} - (\eta_{\delta,k} + \eta_{\delta+1,k})\tilde{C}^k \in \mathbb{R}^{(n+1)\times(n+1)}.$$

We define

$$R_{\delta,k} := \left\{ (S, Z, B_1, \dots, B_r, X_1, \dots, X_r) : \begin{array}{rcl} -\tilde{C}^k \bullet B^k & \geq & -\eta_{\delta+1,k} \\ \tilde{C}^k \bullet B^k & \geq & \eta_{\delta,k} \\ -\tilde{D}_{\delta,k} \bullet B^k & \geq & \eta_{\delta,k}\eta_{\delta+1,k} \end{array} \right\}, \tag{15}$$

for all $\delta = 1, \dots, \nu$ and $k = 1, \dots, r$.

Our goal is to construct, for each $k = 1, \dots, r$, a linear inequality of the form

$$\Gamma_1^k \bullet S + \Gamma_3^k \bullet Z + \sum_{l=1}^{r} (\Gamma_{4,l}^k \bullet B_l + \Gamma_{5,l}^k \bullet X_l) \geq \beta^k \tag{16}$$

that is valid for

$$R^k := \text{convcl}\left(\cup_{\delta=1}^{\nu} (\mathcal{F} \cap R_{\delta,k})\right),$$

where $\mathcal{F}$ is the feasible set of problem (12) and convcl(t) denotes the convex closure of t.

In order to construct the valid inequality, we consider

$$\begin{array}{rl} z_{\delta,k} := & \min \Gamma_1^k \bullet S + \Gamma_3^k \bullet Z + \sum_{l=1}^{r} (\Gamma_{4,l}^k \bullet B_l + \Gamma_{5,l}^k \bullet X_l) \\ & \text{subject to } (S, Z, B_1, \dots, B_r, X_1, \dots, X_r) \in \mathcal{F} \cap R_{\delta,k} \end{array} \tag{17}$$

and choose β^k such that $z_{\delta,k} \geq \beta^k$, for all $\delta = 1, \dots, \nu$.

It is straightforward to verify that Slater's condition holds for problem (17). So, by strong duality, we have

$$z_{\delta,k} := \max\ R \bullet Y + \sum_{l=1}^{r} g_l - \eta_{\delta+1,k} r_{\delta k} + \eta_{\delta,k} s_{\delta k} + \eta_{\delta k}\eta_{\delta+1,k} t_{\delta k} \tag{18}$$

$$
\begin{array}{ll}
\text{subject to} & \\
Y + W - V \preceq \Gamma_1^k, & \\
W + V = \Gamma_3^k, & \\
\begin{pmatrix} g_l & \frac{1}{2}h_l e_1^t \\ \frac{1}{2}h_l e_1 & U^l \end{pmatrix} \preceq \Gamma_{4,l}^k, & l = 1,\dots,r, l \neq k \\
\begin{pmatrix} g_k & \frac{1}{2}h_k e_1^t \\ \frac{1}{2}h_k e_1 & U^k \end{pmatrix} - \tilde{C} r_{\delta k} + \tilde{C} s_{\delta k} - \tilde{D}_{\delta,k} t_{\delta k} \preceq \Gamma_{4,k}^k, & \\
Y - U^l \preceq \Gamma_{5,l}^k, & l = 1,\dots,r, \\
W, V \geq 0, & \\
h_l \geq 0, & l = 1,\dots,r, \\
r_{\delta k}, s_{\delta k}, t_{\delta k} \geq 0, &
\end{array} \tag{19}
$$

where $Y, W, V, U^l \in \mathbb{R}^{n\times n}$ and $h_l, r_{\delta k}, s_{\delta k}, t_{\delta k} \in \mathbb{R}_+$, for all $l = 1,\dots,r$, are the dual variables for problem (17).

Finally, searching for a valid inequality that is violated by a given solution

$$(\hat{S}, \hat{Z}, \hat{B}_1, \dots, \hat{B}_r, \hat{X}_1, \dots, \hat{X}_r),$$

we set $\Gamma_1^k, \Gamma_3^k, \Gamma_{4,1}^k, \dots, \Gamma_{4,r}^k, \Gamma_{5,1}^k, \dots, \Gamma_{5,r}^k$, and β^k by solving the following problem:

$$
\begin{array}{ll}
\Delta^k := \min \Gamma_1^k \bullet \hat{S} + \Gamma_3^k \bullet \hat{Z} + \sum_{l=1}^r (\Gamma_{4,l}^k \bullet \hat{B}_l + \Gamma_{5,l}^k \bullet \hat{X}_l) - \beta^k & \\
\text{subject to} & \\
R \bullet Y + \sum_{l=1}^r g_l - \eta_{\delta+1,k} r_{\delta k} + \eta_{\delta,k} s_{\delta k} + \eta_{\delta,k}\eta_{\delta+1,k} t_{\delta k} \geq \beta^k, & \delta = 1,\dots,\nu, \\
(19), & \\
\sum_i \sum_j |\Gamma_{1ij}^k| \leq 1, \ \sum_i \sum_j |\Gamma_{2ij}^k| \leq 1, & \\
\sum_i \sum_j |\Gamma_{4,l_{ij}}^k| \leq 1, \ \sum_i \sum_j |\Gamma_{5,l_{ij}}^k| \leq 1, & l = 1,\dots,r.
\end{array} \tag{20}
$$

If $\Delta^k < 0$, the valid inequality (16) obtained from the solution of problem (20) is violated by $(\hat{S}, \hat{Z}, \hat{B}_1, \dots, \hat{B}_r, \hat{X}_1, \dots, \hat{X}_r)$. The last constraints in the formulation, included below (19), are normalization constraints, which are added to bound the objective function of the problem from below.

2.5. Our framework to generate the lower bounds

We apply the secant valid inequalities described in Subsection 2.3 and the valid inequalities based on disjunctive programming, described in Subsection 2.4, to iteratively strengthen the relaxation (SDP+RLT). We applied the ideas described in Subsections 2.3 and 2.4, always considering the relaxation (SDP+RLT) as the baseline, and not the relaxation (4). As already mentioned, this can make the cuts generated stronger. Our framework is described in Algorithm 1.

3. A heuristic procedure

In this section we propose a heuristic procedure to construct a feasible solution to the general problem (1), based on the matrices $\hat{X}^1, \dots, \hat{X}^r$, obtained by Algorithm 1. Considering the matrix $\hat{L} := \sum_{k=1}^r \hat{X}^k$, let $\hat{\lambda}_1, \dots, \hat{\lambda}_n$ be the eigenvalues of $\hat{L}$ such that $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_n$, and $\hat{v}_1, \dots, \hat{v}_n$ be the corresponding unit eigenvectors. The heuristic consists of solving the following SDP problem for a given parameter $r_{heur} \leq n$, where the eigenvectors of L are fixed as the r_{heur} eigenvectors of $\hat{L}$ associated to its r_{heur} largest eigenvalues, and the eigenvalues of L are optimized.

$$
\begin{array}{l}
\min_{S,Z,\gamma_1,\dots,\gamma_{r_{heur}}} \ \frac{1}{\alpha} e^t Z e \\
\text{subject to} \\
-Z \leq S \leq Z, \ S + \sum_{k=1}^{r_{heur}} \gamma^k (\hat{v}^k)(\hat{v}^k)^t = R, \ S \succeq 0. \\
\gamma^k \geq 0, \ k = 1,\dots,r_{heur}, \\
S \in \mathcal{S}, \ \sum_{k=1}^{r_{heur}} \gamma^k (\hat{v}^k)(\hat{v}^k)^t \in \mathcal{L}.
\end{array} \tag{21}
$$

Note that the solution of problem (21) corresponds to a feasible solution to problem (1), where the low-rank factor L has its rank fixed at r_{heur}.

Input: $NumIter, MaxCuts, \nu$
Output: lb
Let $\hat{S}, \hat{Z}, \hat{X}^1, \ldots, \hat{X}^r, \hat{x}^1, \ldots, \hat{x}^r$ be the solution of (SDP+RLT) with optimal solution value lb ;
Define (Relaxation) as problem (SDP+RLT) ;
for $iter = 1, \ldots, NumIter$ **do**
  for $k = 1, \ldots, r$ **do**
    Let $\lambda_{k,1}, \ldots, \lambda_{k,n}$ be the eigenvalues of $\hat{X}^k - (\hat{x}^k)(\hat{x}^k)^t$ such that $\lambda_{k,1} \geq \lambda_{k,2} \geq \ldots \geq \lambda_{k,n}$, and $c_{k,1}, \ldots, c_{k,n}$ be the corresponding eigenvectors ;
    Let $num.pos$ be the number of positive eigenvalues of $\hat{X}^k - (\hat{x}^k)(\hat{x}^k)^t$;
    Let $MaxCuts := \min\{num.pos, MaxCuts\}$;
    for $i = 1, \ldots, MaxCuts$ **do**
      $c^k := c_{k,i}$;
      Compute $\eta_L(c^k)$ and $\eta_U(c^k)$;
      Generate the ith linear secant inequality (10) corresponding to k ;
      Let $\eta_{1,k} := \eta_L(c^k)$;
      for $\delta = 1 : (\nu - 1)$ **do**
        $\eta_{\delta+1,k} = \eta_{\delta,k} + (\eta_U(c^k) - \eta_L(c^k))/\nu$;
      Solve problem (20) to generate the ith SDP disjunctive cut (16) corresponding to k ;
  Include the $r \times MaxCuts$ cuts in problem (Relaxation) ;
  Let $\hat{S}, \hat{Z}, \hat{X}^1, \ldots, \hat{X}^r, \hat{x}^1, \ldots, \hat{x}^r$ be the solution of problem (Relaxation) with optimal solution value lb ;

Algorithm 1: Computing a lower bound for problem (3)

4. Numerical experiments

Considering randomly generated matrices R, we apply the framework presented as Algorithm 1 to obtain lower bounds to problem (3). We then solve problem (21) to obtain upper bounds to the more general problem (1).

The input $n \times n$ symmetric positive definite matrices R for the test problems were generated in the following way. We first randomly generate a sparse high rank symmetric matrix $\tilde{S}$, with $\tilde{S}_{ii} = \beta_i$ on the diagonal elements, and $1'$s on $d\%$ of the nondiagonal elements randomly chosen. The remaining elements are equal to zero. We set $\beta_i = 1 + \sum_{j \neq i} \tilde{S}_{ij}$. Next, we randomly generate a dense matrix $\tilde{L}$ with rank equal to r_{input}, where $\tilde{L} := BB^t$, and $B \in \mathbb{R}^{n \times r_{input}}$. The elements of B are integers randomly selected in $[-5, 5]$. We then set $R := \tilde{S} + \tilde{L}$. The reason why we construct the matrix R as the sum of a sparse matrix $\tilde{S}$ and a low-rank matrix $\tilde{L}$ is to directly obtain good feasible solutions for problem (1). These are the best solutions known for our test problems, and they are used to evaluate the solutions produced by the heuristic.

We have set $n = 25$, $r_{input} = 1, 5, 12$, $d = 13.33$, and generated 3 instances for each value of r_{input}. To obtain $d = 13, 33$ on the 25×25 symmetric matrix $\tilde{S}$, we randomly chose 40 out of the 300 elements below the diagonal to be fixed at 1 (40/300=13.33%).

The experiments were conducted on a 3.40 GHz Intel Core i7-3770 CPU, with 8 GB RAM, running under Windows 7. The Matlab software for convex optimization CVX [Grant and Boyd, 2014] was used to solve all optimization problems pointed in Algorithm 1 as well as problem (21). The optimization problems to generate the disjunctive cuts took up to 180 seconds to be solved. The relaxations took up to 20 seconds to be solved.

In Algorithm 1, we have set $MaxIter = 10$, $MaxCuts = 3$, and $\nu = 4$. In our preliminary numerical experiments, we have only considered $r = 1$ in problem (4). We note that the optimal solution value of relaxation (SDP+RLT) is equal to zero, for all instances considered. In the optimal solution of this initial relaxation, we always have $S = 0$ and $L = R$.

Concerning the application of Algorithm 1, we verified from the numerical results that while the bounds obtained with the baseline relaxation (SDP+RLT) were equal to zero for all test problems, the average lower bounds obtained by the algorithm for $r_{input} = 1, 5, 12$ were respectively, 19.78, 391.6, and 313.4, showing the ability of the cuts added by the algorithm to effectively increase the lower bounds.

Concerning the application of the heuristic procedure, to evaluate the quality of the solutions obtained, we measured how close the eigenvalues and eigenvectors of the low-rank factor computed by the heuristic, which we denote by L^h, are to the eigenvalues and eigenvectors of the input matrix $\tilde{L}$. Two measures were considered:

$$dif.l_i := |(\tilde{\lambda}_i - \lambda_i^h)/\tilde{\lambda}_i| \times 100\% \ ,$$
$$dif.v_i := \|\tilde{v}_i - v_i^h\| \ ,$$

for $i = 1, \dots, n$, where $\tilde{\lambda}_i$ and $\tilde{v}_i$ are respectively, the i^{th} eigenvalue and corresponding eigenvector of $\tilde{L}$, and λ_i^h and v_i^h are the i^{th} eigenvalue and corresponding eigenvector of L^h. From our numerical results, we verify that, for $r_{input} = 1, 5, 12$, the average values of $dif.l_i$ are respectively, 0.22%, 3.18%, and 1.98%, and the average values of $dif.v_i$ are respectively, 0.01, 0.59, and 0.52. The results indicate that the heuristic can construct feasible solutions very similar to the input matrices, for our test problems.

5. Concluding remarks

We address a version of the rank-sparsity decomposition problem, where the objective is to decompose a given symmetric positive definite matrix as a sum of a sparse matrix and a low-rank matrix, both positive semidefinite. With the goal of globally solving this nonconvex problem by a branch-and-bound algorithm (as discussed in [Lee and Zou, 2014]), we propose a relaxation for its simplified version, where the rank of the low-rank factor in the decomposition is fixed. We finally present a heuristic procedure that constructs a feasible solution for the original problem, taking the solution of the relaxation as a starting point.

The test problems considered in our numerical experiments were randomly generated. The input matrix R is given by the sum of a sparse matrix $\tilde{S}$ and a low-rank matrix $\tilde{L}$, suggesting that they are a good solution for the general rank-sparsity decomposition problem. These are in fact the best solutions known for these random instances of the problem.

Lower bounds for the simplified version of the rank-sparsity decomposition problem where the low-rank matrix has rank equal to r, were obtained by an iterative algorithm (Algorithm 1), where an initial SDP relaxation of the problem, strengthened by RLT inequalities, is further strengthened by the iterative addition of linear secant inequalities and SDP disjunctive cuts, as it was first proposed by [Saxena et al., 2010] and further extended to the SDP disjunctive cuts by [Lee and Rendl, 2008].

Preliminary results for $r = 1$ show that the lower bounds obtained by Algorithm 1 were significant better than the zero bound always given by the initial relaxation (SDP+RLT), showing the effectiveness of the valid cuts given by both secant inequalities and SDP disjunctive cuts.

It was observed on our numerical experiments, that the solutions $\hat{S}$ and $\hat{L} := \sum_{k=1}^{r} \hat{X}^k$ obtained with Algorithm 1 were very similar to the randomly generated matrices $\tilde{S}$ and $\tilde{L}$, for all values of r_{input}. The r_{input} largest eigenvalues of $\hat{L}$ were always separated from the others, being much larger than the others and very close to the r_{input} positive eigenvalues of $\tilde{L}$. The similarity between $\hat{L}$ and $\tilde{L}$ was observed not only with respect to the eigenvalues, but also with respect to the eigenvectors. This fact motivated the development of a heuristic for the general rank-sparsity decomposition problem, where the solution matrices of the relaxation were used. The feasible solution L^h constructed by the heuristic, has r_{heur} eigenvectors given by the eigenvectors of $\hat{L}$ corresponding to its r_{heur} largest eigenvalues. The eigenvalues of L^h are then optimized, i.e., are chosen in order to minimize the sparsity of the complementary matrix $R - L^h$. We chose the parameter r_{heur} as the number of eigenvalues of $\hat{L}$ that were separated from the others, and with this choice we could construct solutions very similar to the input matrices $\tilde{S}$ and $\tilde{L}$.

Acknowledgments

This research was partially supported by a collaborative grant of the São Paulo Research Foundation (FAPESP) and the University of Michigan (U-M) (2014/50309-0). M. Fampa was partially supported by CNPq grant 303898/2016-0. J. Lee was partially supported by ONR grants N00014-14-1-0315 and N00014-17-1-2296. S. A. Santos was partially supported by FAPESP grants 2013/05475-7 and 2013/07375-0 and CNPq grant 302915/2016-8.

References

Al-Khayyal, A. and Falk, J. E. (1983). Jointly constrained biconvex programming. *Math. Oper. Res.*, 8:273–286.

Anstreicher, K. M. (2009). Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43:471–484.

Fampa, M., Lee, J., and Melo, W. (2017). On global optimization with indefinite quadratics. *To appear in: EURO Journal on Computational Optimization*.

Grant, M. and Boyd, S. (2014). Cvx: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`.

Lee, J. and Rendl, F. (2008). Improved bounds for max-cut via disjunctive programming. *Personal notes*.

Lee, J. and Zou, B. (2014). Optimal rank-sparsity decomposition. *Journal of Global Optimization*, 60:307–315.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: part i convex underestimating problems. *Mathematical Programming*, 10:147–175.

Nesterov, Y. E. (1998). Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9:141–160.

Saxena, A., Bonami, P., and Lee, J. (2010). Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Mathematical Programming, Series B*, 124:383–411.

Saxena, A., Bonami, P., and Lee, J. (2011). Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Mathematical Programming, Series A*, 130(2):359–413.

Sherali, H. D. and Adams, W. P. (1999). *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*. Kluwer Academic Publishers, Dordrecht.

Sherali, H. D. and Tuncbilek, C. H. (1995). A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, 7:1–31.

Zhang, F., editor (2005). *The Schur complement and its applications*, volume 4 of *Numerical Methods and Algorithms*. Springer-Verlag, New York.