Block coordinate descent methods for semidefinite programming

Zaiwen Wen[1], Donald Goldfarb[2], and Katya Scheinberg[3]

[1] Department of Mathematics, Shanghai Jiaotong University. zw2109@sjtu.edu.cn
[2] Department of Industrial Engineering and Operations Research, Columbia University. goldfarb@columbia.edu
[3] Department of Industrial and Systems Engineering, Lehigh University. kas410@lehigh.edu

We consider in this chapter block coordinate descent (BCD) methods for solving semidefinite programming (SDP) problems. These methods are based on sequentially minimizing the SDP problem's objective function over blocks of variables corresponding to the elements of a single row (and column) of the positive semidefinite matrix X; hence, we will also refer to these methods as row-by-row (RBR) methods. Using properties of the (generalized) Schur complement with respect to the remaining fixed $(n-1)$-dimensional principal submatrix of X, the positive semidefiniteness constraint on X reduces to a simple second-order cone constraint. It is well known that without certain safeguards, BCD methods cannot be guaranteed to converge in the presence of general constraints. Hence, to handle linear equality constraints, the methods that we describe here use an augmented Lagrangian approach. Since BCD methods are first-order methods, they are likely to work well only if each subproblem minimization can be performed very efficiently. Fortunately, this is the case for several important SDP problems, including the maxcut SDP relaxation and the minimum nuclear norm matrix completion problem, since closed-form solutions for the BCD subproblems that arise in these cases are available. We also describe how BCD can be applied to solve the sparse inverse covariance estimation problem by considering a dual formulation of this problem. The BCD approach is further generalized by using a rank-two update so that the coordinates can be changed in more than one row and column at each iteration. Finally, numerical results on the maxcut SDP relaxation and matrix completion problems are presented to demonstrate the robustness and efficiency of the BCD approach, especially if only moderately accurate solutions are desired.

1 Introduction

Semidefinite programming (SDP) problems are convex optimization problems that are solvable in polynomial time by interior point methods [57, 64, 70]. Unfortunately however, in practice large scale SDPs are quite difficult to solve because of the very large amount of work required by each iteration of an interior point method. Most of these methods form a positive definite $m \times m$ matrix M, where m is the number of constraints in the SDP, and then compute the search direction by finding the Cholesky factorization of M. Since m can be $O(n^2)$ when the unknown positive semidefinite matrix is $n \times n$, it can take $O(n^6)$ arithmetic operations to do this. Consequently, this becomes impractical both in terms of the time and the amount of memory $O(m^2)$ required when n is much larger than one hundred and m is much larger than a few thousand. Moreover forming M itself can be prohibitively expensive unless m is not too large or the constraints in the SDP are very sparse [27]. Although the computational complexities of the block coordinate descent (BCD) methods presented here are not polynomial, each of their iterations can, in certain cases, be executed much more cheaply than in an interior point algorithm. This enables BCD methods to solve very large instances of these SDPs efficiently. Preliminary numerical testing verifies this. For example, BCD methods produce highly accurate solutions to maxcut SDP relaxation problems involving matrices of size 4000×4000 in less than 5.25 minutes and nuclear norm matrix completion SDPs involving matrices of size 1000×1000 in less than 1 minute on a 3.4 GHZ workstation. If only moderately accurate solutions are required (i.e., a relative accuracy of the order of 10^{-3}) then less than 45 and 10 seconds, respectively, is needed. We note, however, that using a BCD method as a general purpose SDP solver is not a good idea.

1.1 Review of BCD methods

BCD methods are among the oldest methods in optimization. Since solving the original problem with respect to all variables simultaneously can be difficult or very time consuming, these approaches are able to reduce the overall computational cost by partitioning the variables into a few blocks and then minimizing the objective function with respect to each block by fixing all other blocks at each inner iteration. They have been studied in convex programming [45, 60], nonlinear programming [6, 34, 33], nonsmooth separable minimization with and without linearly constraints [58, 63, 59] and optimization by direct search [42]. Although these methods have never been the main focus of the mathematical optimization community, they remain popular with researchers in the scientific and engineering communities. Recently, interest in coordinate descent methods has been revived due to the wide range of large-scale problems in image reconstruction [9, 73, 24], machine learning including support vector machine training [17, 62, 8, 39], mesh optimization [20], compressive

sensing [44, 23, 75, 48], and sparse inverse covariance estimation [2] to which these methods have been successfully applied.

The basic BCD algorithmic strategy can be found under numerous names, including linear and nonlinear Gauss-Seidel methods [53, 33, 35, 69], subspace correction methods [56] and alternating minimization approaches. BCD methods are also closely related to alternating direction augmented Lagrangian (ADAL) methods which alternatingly minimize the augmented Lagrangian function with respect to different blocks of variables and then update the Lagrange multipliers at each iteration. ADAL methods have been applied to many problem classes, such as, variational inequality problems [37, 36, 72], linear programming [21], nonlinear convex optimization [7, 43, 18, 41, 61, 32, 31], maximal monotone operators [22], nonsmooth ℓ_1 minimization arising from compressive sensing [65, 71, 77] and SDP [74, 68].

There are several variants of coordinate and BCD methods. The simplest cyclic (or Gauss-Seidel) strategy is to minimize with respect to each block of variables one after another in a fixed order repeatedly. The essentially cyclic rule [45] selects each block at least once every T successive iterations, where T is an integer equal to or greater than the number of blocks. The Gauss-Southwell rule [45, 63] computes a positive value q_i for every block i according some criteria and then chooses the block with the largest value of q_i to work on next, or chooses the k-th block to work on, where $q_k \geq \beta \max_i q_i$ for $\beta \in (0, 1]$. An extreme case is to move along the direction corresponding to the component of the gradient with maximal absolute value [49, 19]. The approach in [49] chooses a block or a coordinate randomly according to prespecified probabilities for each block or coordinate.

The convergence properties of BCD methods have been intensively studied and we only summarize some results since the 1990s. Bertsekas [6] proved that every limit point generated by the coordinate descent method is a stationary point for the minimization of a general differentiable function $f(x_1, \ldots, x_N)$ over the Cartesian product of closed, nonempty and convex subsets $\{X_i\}_{i=1}^N$, such that $x_i \in X_i$, $i = 1, \ldots, N$, if the minimum of each subproblem is uniquely attained. Grippo and Sciandrone [33] obtained similar results when the objective function f is componentwise strictly quasiconvex with respect to $N - 2$ components and when f is pseudoconvex. Luo and Tseng [45] proved convergence with a linear convergence rate without requiring the objective function to have bounded level sets or to be strictly convex, by considering the problem $\min_{x \geq 0} g(Ex) + b^\top x$, where g is a strictly convex essentially smooth function and E is a matrix. In [58], Tseng studied nondifferentiable (nonconvex) functions f with certain separability and regularity properties, and established convergence results when f is pseudoconvex in every pair of coordinate blocks from among $N - 1$ coordinate blocks or f has at most one minimum in each of $N - 2$ coordinate blocks if f is continuous on a compact level set, and when f is quasiconvex and hemivariate in every coordinate block. Tseng and Yun [63] considered a nonsmooth separable problem whose objective function is the sum of a smooth function and a separable convex

function, which includes as special cases bound-constrained optimization and smooth optimization with ℓ_1-regularization. They proposed a (block) coordinate gradient descent method with an Armijo line search and established global and linear convergence under a local Lipschitz error bound assumption.

Recently, complexity results for BCD methods have also been explored. Saha and Tewari [54] proved $O(1/k)$ convergence rates, where k is the iteration counter, for two cyclic coordinate descent methods for solving $\min_x f(x) + \lambda\|x\|_1$ under an isotonicity assumption. In [49], Nesterov proposed unconstrained and constrained versions of a Random Coordinate Descent Method (RCDM), and showed that for the class of strongly convex functions, RCDM converges with a linear rate, and how to accelerate the unconstrained version of RCDM to have an $O(1/k^2)$ rate of convergence. A stochastic version of the coordinate descent method with runtime bounds was also considered in [55] for ℓ_1-regularized loss minimization.

1.2 BCD methods for SDP

All coordinate descent and block coordinate descent methods for SDP, that maintain positive semidefiniteness of the matrix of variables, are based upon the well known relationship between the positive semidefiniteness of a symmetric matrix and properties of the Schur complement of a sub-matrix of that matrix [76]. We note that Schur complements play an important role in SDP and related optimization problems. For example, they are often used to formulate problems as SDPs [10, 64]. In [1, 29, 30] they are used to reformulate certain SDP problems as second-order cone programs (SOCPs). More recently, they were used by Banerjee, El Ghaoui and d'Aspremont [2] to develop a BCD method for solving the sparse inverse covariance estimation problem whose objective function involves the log determinant of a positive semidefinite matrix. As far as we know, this was the first application of BCD to SDP.

As in the method proposed in Banerjee et. al [2], the basic approach described in this chapter uses Schur complements to develop an *overlapping* BCD method. The coordinates (i.e., variables) in each iteration of these methods correspond to the components of a single row (column) of the unknown semidefinite matrix. Since every row (column) of a symmetric matrix contains one component of each of the other rows (columns), the blocks in these methods overlap. As we shall see below, the convergence result in [6] can be extended to the case of overlapping blocks. However, they do not apply to the case where constraints couple the variables between different blocks. To handle general linear constraints, the BCD methods for SDP described here resort to incorporating these constraints into an augmented Lagrangian function, which is then minimized over each block of variables. Specifically, by fixing any $(n-1)$-dimensional principal submatrix of X and using its Schur complement, the positive semidefinite constraint is reduced to a simple second-order

cone constraint and then a sequence of SOCPs constructed from the primal augmented Lagrangian function are minimized.

Most existing first-order methods for SDP are also based on the augmented Lagrangian method (also referred to as the method of multipliers). Specific methods differ in how the positive semidefinite constraints are handled. In [13, 14], the positive definite variable X is replaced by $RR^\top$ in the primal augmented Lagrangian function, where R is a low rank matrix, and then nonlinear programming approaches are used. In [11, 15], a BCD (alternating minimization) method and an eigenvalue decomposition are used to minimize the primal augmented Lagrangian function. In [78], the positive semidefinite constraint is represented implicitly by using a projection operator and a semismooth Newton approach combined with the conjugate gradient method is applied to minimize the dual augmented Lagrangian function. The regularization methods [47, 50]) and the alternating direction augmented Lagrangian method [68] are also based on a dual augmented Lagrangian approach and the use of an eigenvalue decomposition to maintain complementarity.

We also generalize the BCD approach by using rank-two updates. This strategy also gives rise to SOCP subproblems and enables combinations of the coordinates of the variable matrix X in more than a single row and column to change at each iteration. Hence, it gives one more freedom in designing an efficient algorithm.

1.3 Notation and Organization

We adopt the following notation. The sets of $n \times n$ symmetric matrices and $n \times n$ symmetric positive semidefinite (positive definite) matrices are denoted by $\mathcal{S}^n$ and $\mathcal{S}^n_+$ ($\mathcal{S}^n_{++}$), respectively. The notation $X \succeq 0$ ($X \succ 0$) is also used to indicate that X is positive semidefinite (positive definite). Given a matrix $A \in \mathbb{R}^{n\times n}$, we denote the (i,j)-th entry of A by $A_{i,j}$. Let α and β be given index sets, i.e., subsets of $\{1, 2, \cdots, n\}$. We denote the cardinality of α by $|\alpha|$ and its complement by $\alpha^c := \{1, 2, \cdots, n\}\backslash\alpha$. Let $A_{\alpha,\beta}$ denote the submatrix of A with rows indexed by α and columns indexed by β, i.e.,

$$A_{\alpha,\beta} := \begin{pmatrix} A_{\alpha_1,\,\beta_1} & \cdots & A_{\alpha_1,\,\beta_{|\beta|}} \\ \vdots & & \vdots \\ A_{\alpha_{|\alpha|},\,\beta_1} & \cdots & A_{\alpha_{|\alpha|},\,\beta_{|\beta|}} \end{pmatrix}.$$

We write i for the index set $\{i\}$ and denote the complement of $\{i\}$ by $i^c := \{1, 2, \cdots, n\}\backslash\{i\}$. Hence, A_{i^c,i^c} is the submatrix of A that remains after removing its i-th row and column, and $A_{i^c,i}$ is the ith column of the matrix A without the element $A_{i,i}$. The inner product between two matrices C and X is defined as $\langle C, X\rangle := \sum_{jk} C_{j,k}X_{j,k}$ and the trace of X is defined as $\mathrm{Tr}(X) = \sum_{i=1}^n X_{ii}$. The vector $\begin{pmatrix} x \\ y \end{pmatrix}$ obtained by stacking the vector $x \in \mathbb{R}^p$ on the top of the vector $y \in \mathbb{R}^q$ is also denoted by $[x; y] \in \mathbb{R}^{p+q}$.

The rest of this chapter is organized as follows. In section 2, we briefly review the relationship between properties of the Schur complement and the positive semidefiniteness of a matrix, and present a prototype of the RBR method for solving a general SDP. In section 3.1, the RBR method is specialized for solving SDPs with only diagonal element constraints, and it is interpreted in terms of the logarithmic barrier function. Coordinate descent methods for sparse inverse covariance estimation are reviewed in section 3.3. Convergence of the RBR method for SDPs with only simple bound constraints is proved in section 3.4. To handle general linear constraints, we apply the RBR method in section 4 to a sequence of unconstrained problems using an augmented Lagrangian function approach. Specialized versions for the maxcut SDP relaxation and the minimum nuclear norm matrix completion problem are presented in sections 4.2 and 4.3, respectively. A generalization of the RBR scheme based on a rank-two update is presented in section 5. Finally, numerical results for the maxcut and matrix completion problems, are presented in section 6 to demonstrate the robustness and efficiency of our algorithms.

2 Preliminaries

In this section, we first present a theorem about the Schur complement of a positive (semi-) definite matrix, and then present a RBR prototype method for SDP based on it.

2.1 Schur complement

Theorem 1. *([76], Theorems 1.12 and 1.20) Let the matrix $X \in \mathcal{S}^n$ be partitioned as $X := \begin{pmatrix} \xi & y^\top \\ y & B \end{pmatrix}$, where $\xi \in \mathbb{R}$, $y \in \mathbb{R}^{n-1}$ and $B \in \mathcal{S}^{n-1}$. The Schur complement of B in X is defined as $(X/B) := \xi - y^\top B^\dagger y$, where $B^\dagger$ is the Moore-Penrose pseudo-inverse of B. Then the following holds.*
1) If B is nonsingular, then $X \succ 0$ if and only if $B \succ 0$ and $(X/B) > 0$.
2) If B is nonsingular, then $X \succeq 0$ if and only if $B \succ 0$ and $(X/B) \geq 0$.
3) $X \succeq 0$ if and only if $B \succeq 0$, $(X/B) \geq 0$ and $y \in \mathcal{R}(B)$, where $\mathcal{R}(B)$ is the range space of B.

Proof. We only prove here 1) and 2). Since B is nonsingular, X can be factorized as

$$X = \begin{pmatrix} 1 & y^\top B^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} \xi - y^\top B^{-1} y & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} 1 & 0 \\ B^{-1} y & I \end{pmatrix}. \tag{1}$$

Hence, $\det(X) = (\xi - y^\top B^{-1} y) \det(B)$ and

$$X \succ (\succeq) 0 \Longleftrightarrow B \succ 0 \text{ and } (X/B) := \xi - y^\top B^{-1} y > (\geq) 0. \tag{2}$$

2.2 A RBR method prototype for SDP

We consider here the standard form SDP problem

$$\begin{array}{ll} \min\limits_{X\in\mathcal{S}^n} & \langle C, X\rangle \\ \text{s.t.} & \mathcal{A}(X) = b, \quad X \succeq 0, \end{array} \tag{3}$$

where the linear map $\mathcal{A}(\cdot) : \mathcal{S}^n \to \mathbb{R}^m$ is defined by

$$\mathcal{A}(X) := \left(\left\langle A^{(1)}, X\right\rangle, \cdots, \left\langle A^{(m)}, X\right\rangle\right)^\top,$$

the matrices $C, A^{(i)} \in \mathcal{S}^n$, and the vector $b \equiv (b_1, \ldots, b_m)^\top \in \mathbb{R}^m$ are given. Henceforth, the following Slater condition for (3) is assumed to hold.

Assumption 2 *Problem* (3) *satisfies the Slater condition:*

$$\begin{cases} \mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m \text{ is onto }, \\ \exists X^1 \in \mathcal{S}^n_{++} \text{ such that } \mathcal{A}(X^1) = b. \end{cases} \tag{4}$$

Given a strictly feasible solution $X^k \succ 0$, we can construct a SOCP restriction for the SDP problem (3) as follows. Fix the $n(n-1)/2$ variables in the $(n-1)\times(n-1)$ submatrix $B := X^k_{1^c,1^c}$ of X^k and let ξ and y denote the remaining unknown variables $X_{1,1}$ and $X_{1^c,1}$ (i.e., row 1/column 1), respectively. Hence, the matrix $X := \begin{pmatrix} \xi & y^\top \\ y & B \end{pmatrix} := \begin{pmatrix} \xi & y^\top \\ y & X^k_{1^c,1^c} \end{pmatrix}$. It then follows from Theorem 1 that $X \succeq 0$ is equivalent to $\xi - y^\top B^{-1} y \geq 0$. Here we write this as $\xi - y^\top B^{-1} y \geq \nu$, with $\nu = 0$, so that strict positive definiteness of X can be maintained if we choose $\nu > 0$. Hence, the SDP problem (3) becomes

$$\begin{array}{ll} \min\limits_{[\xi;y]\in\mathbb{R}^n} & \widetilde{c}^\top [\xi; y] \\ \text{s.t.} & \widetilde{A}\,[\xi; y] = \widetilde{b}, \\ & \xi - y^\top B^{-1} y \geq \nu, \end{array} \tag{5}$$

where $\nu = 0$, and $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ are defined as follows using the subscript $i = 1$:

$$\widetilde{c} := \begin{pmatrix} C_{i,i} \\ 2C_{i^c,i} \end{pmatrix}, \quad \widetilde{A} := \begin{pmatrix} A^{(1)}_{i,i} & 2A^{(1)}_{i,i^c} \\ \cdots & \cdots \\ A^{(m)}_{i,i} & 2A^{(m)}_{i,i^c} \end{pmatrix} \text{ and } \widetilde{b} := \begin{pmatrix} b_1 - \left\langle A^{(1)}_{i^c,i^c}, B\right\rangle \\ \cdots \\ b_m - \left\langle A^{(m)}_{i^c,i^c}, B\right\rangle \end{pmatrix}. \tag{6}$$

If we let $LL^\top = B$ be the Cholesky factorization of B and introduce a new variable $z = L^{-1}y$, the Schur complement constraint $\xi - y^\top B^{-1} y \geq \nu$ is equivalent to the linear constraints $Lz = y$ and $\eta = \xi - \nu$ and the rotated second-order cone constraint $\|z\|_2^2 \leq \eta$. Clearly, similar problems can be constructed if for any i, $i = 1, \cdots, n$, all elements of X^k other than those in the

i-th row/column are fixed and only the elements in the i-th row/column are treated as unknowns.

We now present the RBR method for solving (3). Starting from a positive definite feasible solution X^1, we update one row/column of the solution X at each of n inner steps by solving subproblems of the form (5) with $\nu > 0$. As we shall show below, choosing $\nu > 0$ in (5) (i.e., keeping all iterates positive definite), is necessary for the RBR method to be well-defined. This procedure from the first row to the n-th row is called a *cycle*. At the first step of the k-th cycle, we fix $B := X^k_{1^c,1^c}$, and solve subproblem (5), whose solution is denoted by $[\xi; y]$. Then the first row/column of X^k is replaced by $X^k_{1,1} := \xi$ and $X^k_{1^c,1} := y$. Similarly, we set $B := X^k_{i^c,i^c}$ in the i-th inner iteration and assign the parameters $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ according to (6). Then the solution $[\xi; y]$ of (5) is used to set $X^k_{i,i} := \xi$ and $X^k_{i^c,i} := y$. The k-th cycle is finished after the n-th row/column is updated. Then we set $X^{k+1} := X^k$ and repeat this procedure until the relative decrease in the objective function on a cycle becomes smaller than some tolerance ϵ. This RBR method prototype is outlined in Algorithm 1. In the next section, we illustrate its usefulness for problems in which the linear constraints are simple bound constraints. Unfortunately, when they are not, the RBR prototype fails. Hence, for the general case we present an augmented Lagrangian version of the RBR method is section 4.

Algorithm 1: A RBR method prototype

Set $X^1 \succ 0$, $\nu \geq 0$, $k := 1$ and $\epsilon \geq 0$. Set $F^0 := +\infty$ and compute $F^1 := \langle C, X^1 \rangle$.

while $\frac{F^{k-1} - F^k}{\max\{|F^{k-1}|,1\}} \geq \epsilon$ **do**

  for $i = 1, \cdots, n$ **do**

    Set $B := X^k_{i^c,i^c}$ and the parameters $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ according to (6).

    Solve the subproblem (5) whose solution is denoted by ξ and y.

    Update $X^k_{i,i} := \xi$, $X^k_{i^c,i} := y$ and $X^k_{i,i^c} := y^\top$.

  Compute $F^k := \langle C, X^k \rangle$. Set $X^{k+1} := X^k$ and $k := k + 1$.

We note that the RBR method is similar to the block *Gauss-Seidel* method for solving a system of linear equations and the block coordinate descent method (sometimes referred to as the nonlinear Gauss-Seidel method) for nonlinear programming, except that because of the symmetry of X, the blocks in the RBR method overlap. Specifically, exactly one of the variables in any two inner iterations of the RBR method overlap.

3 The RBR methods for SDPs with bound constraints

We now apply the RBR method to SDPs with simple bound constraints, including the maxcut SDP relaxation, the SDP relaxation of the matrix completion problem, and the sparse inverse covariance estimation problem. Convergence of the RBR method for such problems is also analyzed.

3.1 Maxcut SDP relaxation

The well known SDP relaxation [28, 12, 38, 3] for the maxcut problem, which seeks to partition the vertices of a graph into two sets so that the sum of the weighted edges connecting vertices in one set with vertices in the other set is maximized, takes the following form:

$$\begin{array}{ll} \min\limits_{X\succeq 0} & \langle C, X\rangle \\ \text{s.t.} & X_{ii} = 1, \quad i = 1, \cdots, n. \end{array} \tag{7}$$

We now present the RBR subproblem for solving (7). Since the diagonal elements of X are known to be equal to 1, they are kept fixed at 1. At the ith step of the k-th cycle, we fix $B = X^k_{i^c,i^c}$, where X^k is the iterate at the $(i-1)$-st step of the k-th cycle. Although in all RBR algorithms positive definiteness of all iterates is maintained, we assume here that B is positive semidefinite and use the generalized Schur complement to construct the second-order cone constraint. Hence, the RBR subproblem (5) for problem (7) is

$$\begin{array}{ll} \min\limits_{y\in\mathbb{R}^{n-1}} & \widehat{c}^{\top} y \\ \text{s.t.} & 1 - y^{\top} B^{\dagger} y \geq \nu, \quad y \in \mathcal{R}(B), \end{array} \tag{8}$$

where $\widehat{c} := 2C_{i^c,i}$.

Lemma 1. *If $\gamma := \widehat{c}^{\top} B \widehat{c} > 0$, the solution of problem (8) with $\nu < 1$ is given by*

$$y = -\sqrt{\frac{1-\nu}{\gamma}} B\widehat{c}. \tag{9}$$

Otherwise, $y = 0$ is a solution.

Proof. Suppose that the matrix $B \in \mathcal{S}^n_+$ has rank r, where $0 < r \leq n$. Hence, B has the spectral decomposition

$$B = Q\Lambda Q^{\top} = \begin{pmatrix} Q_r & Q_l \end{pmatrix} \begin{pmatrix} \Lambda_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} Q_r^{\top} \\ Q_l^{\top} \end{pmatrix} = Q_r \Lambda_r Q_r^{\top}. \tag{10}$$

where Q is an orthogonal matrix, $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_r, 0, \cdots 0)$, and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$, and the Moore-Penrose pseudo-inverse of B is

$$B^\dagger = \begin{pmatrix} Q_r & Q_l \end{pmatrix} \begin{pmatrix} \Lambda_r^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} Q_r^\top \\ Q_l^\top \end{pmatrix} = Q_r \Lambda_r^{-1} Q_r^\top.$$

Let $z = Q^\top y =: [z_r; z_l]$. Since $y \in \mathcal{R}(B)$ and $\mathcal{R}(B) = \mathcal{R}(Q_r)$, $z_l = 0$; hence, problem (8) is equivalent to

$$\begin{array}{ll} \min\limits_{z_r \in \mathbb{R}^r} & (Q_r^\top \widehat{c})^\top z_r \\ \text{s.t.} & 1 - z_r^\top \Lambda_r^{-1} z_r \geq \nu, \end{array} \tag{11}$$

whose Lagrangian function is $\ell(z_r, \lambda) = (Q_r^\top \widehat{c})^\top z_r - \frac{\lambda}{2}(1 - \nu - z_r^\top \Lambda^{-1} z_r)$, where $\lambda \geq 0$. At an optimal solution z_r^* to (11),

$$\nabla_{z_r} \ell(z_r^*, \lambda^*) = Q_r^\top \widehat{c} + \lambda^* \Lambda_r^{-1} z_r^* = 0. \tag{12}$$

Suppose that $1 - (z_r^*)^\top \Lambda^{-1} z_r^* > \nu$. It follows from the complementary conditions that $\lambda^* = 0$, which implies that $Q_r^\top \widehat{c} = 0$ and $\gamma = 0$ by using (12). It is obvious that $y^* = 0$ is a solution. Otherwise, z_r^* satisfies the constraint (11) with equality, i.e., $1 - (z_r^*)^\top \Lambda^{-1} z_r^* = \nu$. Then, we have $z_r^* = -\Lambda_r Q_r^\top \bar{c}/\lambda^*$ and

$$1 - \frac{\widehat{c}^\top Q_r \Lambda_r \Lambda_r^{-1} \Lambda_r Q_r^\top \widehat{c}}{(\lambda^*)^2} = 1 - \frac{\gamma}{(\lambda^*)^2} = \nu.$$

Since $\nu < 1$, we must have $\gamma > 0$. Hence, we obtain $\lambda^* = \sqrt{\gamma/(1-\nu)}$ and

$$y^* = Q_r z_r^* = -\sqrt{\frac{1-\nu}{\gamma}} Q_r \Lambda_r Q_r^\top \widehat{c} = -\sqrt{\frac{1-\nu}{\gamma}} B\widehat{c}.$$

□

For simplicity, we let PURE-RBR-M denote the RBR method for the max-cut SDP described above. PURE-RBR-M is extremely simple since only a single matrix-vector product is involved at each inner step. Numerical experiments show PURE-RBR-M works fine if the initial solution X is taken as the identity matrix even if we take $\nu = 0$. However, there exist examples where starting from a rank-one point that is not optimal, the RBR method using $\nu = 0$ either does not move away from the initial solution or it moves to a non-optimal rank-one solution and stays there.

We next interpret PURE-RBR-M as a variant of the RBR method applied to a logarithmic barrier function approximation to (7). Consider the logarithmic barrier problem for (7), i.e.,

$$\begin{array}{ll} \min\limits_{X \in \mathcal{S}^n} & \phi_\sigma(X) := \langle C, X \rangle - \sigma \log\det X \\ \text{s.t.} & X_{ii} = 1, \forall i = 1, \cdots, n, \quad X \succ 0, \end{array} \tag{13}$$

where we define $\log\det(X)$ to be negative infinity for X not positive definite. Given a row i and fixing the block $B = X_{i^c, i^c}$, we have from (1) that

$$\det(X) = \det(B)(1 - X_{i^c,i}^\top B^{-1} X_{i^c,i}),$$

which implies that

$$\phi_\sigma(X) := \widehat{c}^\top X_{i^c,i} - \sigma \log(1 - X_{i^c,i}^\top B^{-1} X_{i^c,i}) + w(B),$$

where $\widehat{c} = 2C_{i^c,i}$ and $w(B)$ is a function of B (i.e., a constant). Hence, the RBR subproblem for (13) is the unconstrained minimization problem

$$\min_{y \in \mathbb{R}^{n-1}} \quad \widehat{c}^\top y - \sigma \log(1 - y^\top B^{-1} y). \tag{14}$$

Lemma 2 below shows that PURE-RBR-M is essentially the RBR method applied solving problem (13), if in the former algorithm ν is replaced by $2\sigma\frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma}$. If B is only positive semidefinite the (14) is replaced by

$$\min_{y \in \mathbb{R}^{n-1}} \quad \widehat{c}^\top y - \sigma \log(1 - y^\top B^{\dagger} y), \quad \text{s.t. } y \in \mathcal{R}(B). \tag{15}$$

Lemma 2. *If $\gamma := \widehat{c}^\top B \widehat{c} > 0$, the solution of problem* (15) *is*

$$y = -\frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma} B\widehat{c}. \tag{16}$$

Hence, the subproblem (8) *has the same solution as* (15) *if $\nu = 2\sigma\frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma}$.*

Proof. Similar to Lemma 1, we have the spectral decomposition (10) of B. Let $z = Q^\top y =: [z_r; z_l]$. Since $y \in \mathcal{R}(B)$ and $\mathcal{R}(B) = \mathcal{R}(Q_r)$, we obtain $z_l = 0$ and hence $y = Q_r z_r$. Therefore, problem (15) is equivalent to

$$\min_{z_r} \quad (Q_r^\top \widehat{c})^\top z_r - \sigma \log(1 - z_r^\top \Lambda_r^{-1} z_r), \tag{17}$$

whose first-order optimality conditions are

$$Q_r^\top \widehat{c} + \frac{2\sigma \Lambda_r^{-1} z_r^*}{1 - (z_r^*)^\top \Lambda_r^{-1} z_r^*} = 0, \text{ and } 1 - (z_r^*)^\top \Lambda_r^{-1} z_r^* > 0. \tag{18}$$

Let $\theta = 1 - (z_r^*)^\top \Lambda_r^{-1} z_r^*$. Then equation (18) implies that $z_r^* = -\frac{\theta \Lambda_r Q_r^\top \widehat{c}}{2\sigma}$. Substituting this expression for z_r^* into the definition of θ, we obtain $\theta^2 \frac{\gamma}{4\sigma^2} + \theta - 1 = 0$, which has a positive root $\theta = \frac{2\sigma\sqrt{\sigma^2+\gamma}-2\sigma^2}{\gamma}$. Hence, $y^* = -\frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma} B\widehat{c}$. Since $\nabla^2 \phi_\sigma(y) \succeq 0$, y^* is an optimal solution of (15). Furthermore, problems (8) and (15) are equivalent if $\frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma} = \sqrt{\frac{1-\nu}{\gamma}}$; that is $\nu = 2\sigma\frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma}$. □

Remark 1. Note from (16) that $\lim_{\sigma \to 0} y = -\frac{B\widehat{c}}{\sqrt{\gamma}}$.

3.2 Matrix Completion

Given a matrix $M \in \mathbb{R}^{p\times q}$ and an index set

$$\Omega \subseteq \{(i,j) \mid i \in \{1,\cdots,p\}, j \in \{1,\cdots,q\}\},$$

the nuclear norm matrix completion problem is

$$\begin{array}{ll} \min_{W\in\mathbb{R}^{p\times q}} & \|W\|_* \\ s.t. & W_{ij} = M_{ij},\ \forall\ (i,j) \in \Omega. \end{array} \tag{19}$$

An equivalent SDP formulation of (19) is

$$\begin{array}{ll} \min_{X\in\mathcal{S}^n} & \mathrm{Tr}(X) \\ s.t. & X := \begin{bmatrix} X^{(1)} & W \\ W^\top & X^{(2)} \end{bmatrix} \succeq 0 \\ & W_{ij} = M_{ij},\ \forall\ (i,j) \in \Omega, \end{array} \tag{20}$$

where $n = p + q$ and the number of linear constraints is $m = |\Omega|$. Let M_Ω be the vector whose elements are the components of $\{M_{i,j} \mid (i,j) \in \Omega\}$ obtained by stacking the columns of M from column 1 to column q and then keeping only those elements that are in Ω. Hence, M_Ω corresponds to the right hand side b of the constraints in the general SDP (3).

We now present the RBR subproblem (46) corresponding to problem (20). First, the vector y can be partitioned into two subvectors corresponding to elements whose indices are, respectively, in and not in the set Ω:

$$y :\approx \begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix}, \quad \widehat{y} := X_{\alpha,i},\ \text{and}\ \widetilde{y} := X_{\beta,i},$$

where, the index sets $\beta := i^c \backslash \alpha$ and

$$\alpha := \begin{cases} \{j+p,\ \mid j \in \bar{\alpha}\},\ \text{where}\ \bar{\alpha} := \{j \mid (i,j) \in \Omega, j = 1,\cdots,q\},\ \text{if}\ i \le p, \\ \{j \mid (j,i) \in \Omega, j = 1,\cdots,p\},\ \text{if}\ p < i \le n. \end{cases} \tag{21}$$

Letting

$$\widetilde{b} := \begin{cases} (M_{i,\bar{\alpha}})^\top, & \text{if } i \le p, \\ M_{\alpha,i-p}, & \text{if } p < i \le n, \end{cases} \tag{22}$$

the RBR subproblem (5) becomes

$$\begin{array}{ll} \min\limits_{(\xi;y)\in\mathbb{R}^n} & \xi \\ \text{s.t.} & \widehat{y} = \widetilde{b}, \quad \xi - y^\top B^{-1} y \ge \nu, \end{array} \tag{23}$$

where the matrix $B = \begin{pmatrix} X^k_{\alpha,\alpha} & X^k_{\alpha,\beta} \\ X^k_{\beta,\alpha} & X^k_{\beta,\beta} \end{pmatrix}$.

Lemma 3. *The optimal solution of the RBR subproblem* (23) *is given by*

$$\xi = \lambda^\top \widetilde{b} + \nu, \quad \widetilde{y} = X^k_{\beta,\alpha}\lambda, \quad \text{where,} \quad \lambda = (X^k_{\alpha,\alpha})^{-1}\widetilde{b}. \tag{24}$$

Proof. Note that the optimal solution $[\xi; y] = [\xi; \widehat{y}; \widetilde{y}]$ of (23) must satisfy $\xi = y^\top B^{-1} y + \nu$. Hence, (23) is equivalent to the linearly constrained quadratic minimization problem

$$\min_y \{y^\top B^{-1} y \mid \widehat{y} = \widetilde{b}\} \tag{25}$$

whose optimality conditions are

$$\begin{pmatrix} X^k_{\alpha,\alpha} & X^k_{\alpha,\beta} \\ X^k_{\beta,\alpha} & X^k_{\beta,\beta} \end{pmatrix}^{-1} \begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix} - \begin{pmatrix} \lambda \\ \mathbf{0} \end{pmatrix} = 0, \tag{26}$$

which implies that

$$\begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix} = \begin{pmatrix} \widetilde{b} \\ \widetilde{y} \end{pmatrix} = \begin{pmatrix} X^k_{\alpha,\alpha} \\ X^k_{\beta,\alpha} \end{pmatrix} \lambda.$$

□

Note from (24) that we only need to solve a single system of linear equations, whose size is the number of known elements in the row and hence expected to be small, to obtain the minimizer of the RBR subproblem (23).

3.3 Sparse inverse covariance estimation

In this subsection, we review the block coordinate descent methods proposed in [2] and [26] for solving the sparse inverse covariance estimation problem. Given an empirical covariance matrix $S \in \mathcal{S}^n$, the problem is to maximize the ℓ_1-penalized log-likelihood function, i.e.,

$$\hat{\Sigma}^{-1} = \arg\max_{X \succ 0} \ \log\det X - \mathrm{Tr}(SX) - \lambda \|X\|_1, \tag{27}$$

where $\lambda > 0$ and $\|X\|_1 = \sum_{i,j} |X_{i,j}|$. Instead of solving (27) directly, the approaches in [2] and [26] consider the dual of (27)

$$\hat{\Sigma} = \arg\max_{W \succ 0} \ \log\det W, \quad \text{s.t.} \quad \|W - S\|_\infty \leq \lambda, \tag{28}$$

which is a problem with only simple bound constraints. To derive this, note that (27) is equivalent to

$$\max_{X \succ 0} \min_{\|U\|_\infty \leq \lambda} \ \log\det X - \mathrm{Tr}(X(S+U)), \tag{29}$$

since the ℓ_1-norm $\|X\|_1$ can be expressed as $\max_{\|U\|_\infty \leq 1} \mathrm{Tr}(XU)$, where $\|U\|_\infty$ is the maximum of the absolute values of the elements of the symmetric matrix U. It is obvious that

$$-\log\det(S+U)-n=\max_{X\succ 0}\quad \log\det X-\mathrm{Tr}(X(S+U)).$$

Hence, the dual (28) is obtained by exchanging the max and the min in (29).

The subproblems solved at each iteration of the BCD methods in [2] and [26] are constructed as follows. Given a positive definite matrix $W \succ 0$, W and S are partitioned according to the same pattern as

$$W=\begin{pmatrix}\xi & y^\top\\ y & B\end{pmatrix} \text{ and } S=\begin{pmatrix}\xi_S & y_S^\top\\ y_S & B_S\end{pmatrix},$$

where $\xi,\xi_S\in\mathbb{R}$, $y,y_S\in\mathbb{R}^{n-1}$ and $B,B_S\in S^{n-1}$. Since $\log\det W=\log(\xi-y^\top B^{-1}y)\det B$, and B is fixed, the RBR subproblem for (28) becomes the quadratic program

$$\min_{[\xi;y]}\ y^\top B^{-1}y-\xi,\quad \text{s.t.}\quad \|[\xi;y]-[\xi_S;y_S]\|_\infty\le\lambda,\ \xi\ge 0. \tag{30}$$

Note that (30) is separable in y and ξ. The solution ξ is equal to $\xi_S+\lambda$. In fact, the first-order optimality conditions of (27) and $X\succ 0$ imply that $W_{ii}=S_{ii}+\lambda$ for $i=1,\dots,n$. Hence, problem (30) reduces to

$$\min_{y}\ y^\top B^{-1}y,\quad \text{s.t.}\quad \|y-y_S\|_\infty\le\lambda. \tag{31}$$

It can be verified that the dual of (31) is

$$\min_{x}\ x^\top Bx-y_S^\top x+\lambda\|x\|_1, \tag{32}$$

which is also equivalent to

$$\min_{x}\ \left\|B^{\frac{1}{2}}x-\frac{1}{2}B^{-\frac{1}{2}}y_S\right\|_2^2+\lambda\|x\|_1. \tag{33}$$

If x solves (33), then $y=Bx$ solves (31).

The BCD method in [2] solves a sequence of constrained problems (31). Specifically, the initial point is set to $W^1=S+\lambda I$ so that only off-diagonal elements have to be updated. The parameters $B:=W^k_{i^c,i^c}$, $y_S=S_{i^c,i}$ and $B_S=S_{i^c,i^c}$ are assigned in the i-th inner iteration at k-th cycle. Then the solution y of (30) is computed and one sets $W^k_{i^c,i}:=y$. A similar procedure is used in the approach in [26] except that the solution y is obtained by solving the so-called LASSO problem (33) using a coordinate descent algorithm, which does not require computation of either $B^{\frac{1}{2}}$ or $B^{-\frac{1}{2}}$.

3.4 Convergence results

The RBR method can be extended to solve

$$\begin{array}{ll}\min\limits_{X\in\mathcal{S}^n} & \psi_\sigma(X) := f(X) - \sigma\log\det X \\ \text{s.t.} & X\in\mathcal{X} := \{X\in\mathcal{S}^n \mid L\le X\le U, X\succ 0\}.\end{array} \tag{34}$$

where $f(X)$ is a differentiable convex function of X, the constant matrices $L, U \in \mathcal{S}^n$ satisfy $L \le U$ and $L \le X$ means that $L_{i,j} \le X_{i,j}$ for all $i, j = 1, \cdots, n$. Note that $L_{i,j} = -\infty$ ($U_{i,j} = \infty$) if $X_{i,j}$ is unbounded below (above). Clearly, problem (34) includes (13) and the logarithmic barrier function version of problems (20) and (28) as special cases. Starting from the point $X^k \succ 0$ at the k-th cycle, we fix the $n(n-1)/2$ variables in the $(n-1)\times(n-1)$ submatrix $B := X^k_{i^c,i^c}$ of X^k and let ξ and y denote the remaining unknown variables $X_{i,i}$ and $X_{i^c,i}$ (i.e., row i/column i), respectively; i.e., $X^k :\approx \begin{pmatrix}\xi & y^\top\\ y & B\end{pmatrix}$. Hence, the RBR subproblem for problem (34) becomes

$$\begin{array}{ll}\min\limits_{X\in\mathcal{S}^n} & \widetilde{f}(\xi,y) - \sigma\log(\xi - y^\top B^{-1}y) \\ \text{s.t.} & \begin{pmatrix}L_{i,i}\\ L_{i^c,i}\end{pmatrix} \le \begin{pmatrix}\xi\\ y\end{pmatrix} \le \begin{pmatrix}U_{i,i}\\ U_{i^c,i}\end{pmatrix},\end{array} \tag{35}$$

where $\widetilde{f}(\xi, y) := f(X^k)$. Inspired by Proposition 2.7.1 in [6], we now prove the following convergence result for the RBR method applied to problem (34).

Theorem 3. *Let $\{X^k\}$ be a sequence generated by the RBR method for solving (34). Assume that the level set $\{X\in\mathcal{X}|\psi_\sigma(X)\le\psi_\sigma(X^1)\}$ is compact. Then every limit point of $\{X^k\}$ is a global minimizer of (34).*

Proof. Clearly, the RBR method produces a sequence of nondecreasing objective function values

$$\psi_\sigma(X^k)\ge\psi_\sigma(X^{k,1})\ge\psi_\sigma(X^{k,2})\ge\cdots\ge\psi_\sigma(X^{k,n-1})\ge\psi_\sigma(X^{k+1}). \tag{36}$$

Let $\widetilde{X}$ be a limit point of the sequence $\{X^k\}$. It follows from equation (36) that the sequences $\{\psi_\sigma(X^k)\}$, $\{\psi_\sigma(X^{k,1})\}$, $\cdots$, $\{\psi_\sigma(X^{k,n-1})\}$ all converge to a bounded number $\psi_\sigma(\widetilde{X})$. Hence, $\widetilde{X}$ must be positive definite. We now show that $\widetilde{X}$ minimizes $\psi_\sigma(X)$.

Let $\{X^{k_j}\}$ be a subsequence of $\{X^k\}$ that converges to $\widetilde{X}$. We first show that $\{X^{k_j,1} - X^{k_j}\}$ converges to zero as $j\to\infty$. Assume on the contrary, that $\{X^{k_j,1} - X^{k_j}\}$ does not converges to zero. Then there exists a subsequence $\{\bar{k}_j\}$ of $\{k_j\}$ and some $\bar{\gamma} > 0$ such that $\gamma^{\bar{k}_j} := \|X^{\bar{k}_j,1} - X^{\bar{k}_j}\|_F \ge \bar{\gamma}$ for all j. Let $D^{\bar{k}_j,1} := (X^{\bar{k}_j,1} - X^{\bar{k}_j})/\gamma^{\bar{k}_j}$. Thus $X^{\bar{k}_j,1} = X^{\bar{k}_j} + \gamma^{\bar{k}_j} D^{\bar{k}_j,1}$, $\|D^{\bar{k}_j,1}\|_F = 1$ and $D^{\bar{k}_j,1}$ differs from zero only along the first row/column. Since $D^{\bar{k}_j,1}$ belongs to a compact set, it has a limit point $\bar{D}^1$. Hence, there exists a subsequence of $\{\hat{k}_j\}$ of $\{\bar{k}_j\}$ such that $D^{\hat{k}_j,1}$ converges to $\bar{D}^1$. Consider an arbitrary $t\in[0,1]$. Since $0\le t\bar{\gamma}\le\gamma^{\hat{k}_j}$, $X^{\hat{k}_j} + tD^{\hat{k}_j,1}$ lies on the segment joining $X^{\hat{k}_j}$ and $X^{\hat{k}_j} + \gamma^{\hat{k}_j} D^{\hat{k}_j,1} = X^{\hat{k}_j,1}$, and belongs to $\mathcal{X}$ since $\mathcal{X}$ is a convex set. Moreover,

since $X^{\hat{k}_j,1}$ uniquely minimizes $\psi_\sigma(X)$ over all X that differ from $X^{\hat{k}_j}$ along the first row/column, it follows from the convexity of $\psi_\sigma(X)$ that

$$\psi_\sigma(X^{\hat{k}_j,1}) = \psi_\sigma(X^{\hat{k}_j} + \gamma^{\hat{k}_j} D^{\hat{k}_j,1}) \le \psi_\sigma(X^{\hat{k}_j} + t\gamma^{\hat{k}_j} D^{\hat{k}_j,1}) \le \psi_\sigma(X^{\hat{k}_j}). \quad (37)$$

Since $\psi_\sigma(X^{\hat{k}_j,1})$ converges to $\psi_\sigma(\widetilde{X})$, it follows (37) that $\psi_\sigma(\widetilde{X}) \le \psi_\sigma(\widetilde{X} + t\bar{\gamma}\bar{D}^1) \le \psi_\sigma(\widetilde{X})$, which implies that $\psi_\sigma(\widetilde{X}) = \psi_\sigma(\widetilde{X} + t\bar{\gamma}\bar{D}^1)$ for all $t \in [0,1]$. Since $\bar{\gamma}\bar{D}^1 \neq 0$, this contradicts the fact that $\psi_\sigma(X)$ is strictly convex; hence $X^{k_j,1} - X^{k_j}$ converges to zero and $X^{k_j,1}$ converges to $\widetilde{X}$.

From the definition (34), we have $\psi_\sigma(X^{k_j,1}) \le \psi_\sigma(X)$ for all

$$X \in V^{k_j,1} := \left\{ \begin{pmatrix} \xi & y^\top \\ y & X^{k_j}_{1^c,1^c} \end{pmatrix} \middle| \begin{pmatrix} \xi \\ y \end{pmatrix} \in \mathbb{R}^n, \begin{pmatrix} L_{1,1} \\ L_{1^c,1} \end{pmatrix} \le \begin{pmatrix} \xi \\ y \end{pmatrix} \le \begin{pmatrix} U_{1,1} \\ U_{1^c,1} \end{pmatrix} \right\}.$$

Taking the limit as j tends to infinity, we obtain that $\psi_\sigma(\widetilde{X}) \le \psi_\sigma(X)$ for all

$$X \in V^1 := \left\{ \begin{pmatrix} \xi & y^\top \\ y & \widetilde{X}_{1^c,1^c} \end{pmatrix} \middle| \begin{pmatrix} \xi \\ y \end{pmatrix} \in \mathbb{R}^n, \begin{pmatrix} L_{1,1} \\ L_{1^c,1} \end{pmatrix} \le \begin{pmatrix} \xi \\ y \end{pmatrix} \le \begin{pmatrix} U_{1,1} \\ U_{1^c,1} \end{pmatrix} \right\},$$

which implies that, for any $p \in \{1, \cdots, n\}$,

$$\psi_\sigma(\widetilde{X}) \le \psi_\sigma(X), \quad \forall X \in V^1 \text{ and } X_{p^c,1} = \widetilde{X}_{p^c,1},$$

i.e., all components of the first row and column $[\xi; y]$ other than the p-th are fixed. Since $\widetilde{X}$ lies in the open convex set $\mathcal{S}^n_{++}$, we obtain from the optimality conditions that, for any $p \in \{1, \cdots, n\}$,

$$\left\langle \nabla\psi_\sigma(\widetilde{X}), X - \widetilde{X} \right\rangle \ge 0, \quad \forall X \in V^1 \text{ and } X_{p^c,1} = \widetilde{X}_{p^c,1},$$

which further gives that, for any $p \in \{1, \cdots, n\}$,

$$\left(\nabla\psi_\sigma(\widetilde{X})\right)_{p,1} \left(X_{p,1} - \widetilde{X}_{p,1}\right) \ge 0, \forall X_{p,1} \text{ such that } L_{p,1} \le X_{p,1} \le U_{p,1}. \quad (38)$$

Repeating the above argument shows that for $i = 2, \cdots, n$, the points $X^{k_j,i}$ also converges to $\widetilde{X}$ and

$$\left(\nabla\psi_\sigma(\widetilde{X})\right)_{p,i} \left(X_{p,i} - \widetilde{X}_{p,i}\right) \ge 0, \quad \forall L_{p,i} \le X_{p,i} \le U_{p,i}, \quad (39)$$

for any $p \in \{1, \cdots, n\}$. Therefore, for any $X \in \mathcal{X}$, it follows from (38) and (39) that

$$\left\langle \nabla\psi_\sigma(\widetilde{X}), X - \widetilde{X} \right\rangle = \sum_{i,j=1,\cdots,n} \left(\nabla\psi_\sigma(\widetilde{X})\right)_{i,j} \left(X_{i,j} - \widetilde{X}_{i,j}\right) \ge 0,$$

which implies that $\widetilde{X}$ is a global minimizer.

4 A RBR method for SDP with general linear constraints

We now consider SDP problem (3) with general linear constraints. Unfortunately, in this case, the RBR method may not converge to an optimal solution. This is similar to the fact that the BCD method may not converge to an optimal solution for a linearly constrained convex problem [33]. It has long been known in [51] that the coordinate descent method for general nonlinear programming may not converge. Here is a 2-dimensional example that shows that for general linear constraints the RBR method may not converge to a global minimizer. Consider the SDP

$$\begin{array}{ll} \min & X_{11} + X_{22} - \log\det(X) \\ \text{s.t.} & X_{11} + X_{22} \geq 4, \quad X \succeq 0. \end{array} \tag{40}$$

Starting from a point X, where $X_{11} = 1$, $X_{12} = 0$ and $X_{22} = 3$, the RBR subproblems are

$$\min \quad X_{11} - \log(3X_{11} - X_{12}^2), \text{ s.t. } X_{11} \geq 1,$$

and

$$\min \quad X_{22} - \log(X_{22} - X_{12}^2), \text{ s.t. } X_{22} \geq 3,$$

since $\det(X) = X_{11}X_{22} - X_{12}^2$. It is readily verified that optimal solutions to these subproblems are, respectively, $X_{11} = 1$, $X_{12} = 0$ and $X_{12} = 0$, $X_{22} = 3$; hence, the RBR method remains at the initial point, while the true optimal solution is $X = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$.

To overcome this type of failure, the coordinate descent method is usually applied to a sequence of unconstrained problems obtained by penalizing the constraints in the objective function. We adopt a similar approach here by embedding the pure RBR method in an augmented Lagrangian function framework. We then introduce specialized versions of this algorithm for the SDP relaxation of the maxcut problem (7) and the minimum nuclear norm matrix completion problem.

4.1 A RBR augmented Lagrangian method

In this subsection, we first introduce an augmented Lagrangian method and then combine it with the RBR method for solving the standard form SDP (3).

The augmented Lagrangian function for problem (3) taking into consideration only the general linear constraints $\mathcal{A}(X) = b$ is defined as:

$$\mathcal{L}(X, \pi, \mu) := \langle C, X\rangle - \pi^\top (\mathcal{A}(X) - b) + \frac{1}{2\mu}\|\mathcal{A}(X) - b\|_2^2, \tag{41}$$

where $\pi \in \mathbb{R}^m$ and $\mu > 0$. Starting from $\pi^1 = \mathbf{0}$, $\mu^1 \in (0, +\infty)$ and $0 < \eta < 1$, our augmented Lagrangian method iteratively solves

$$X^k := \arg\min_X \mathcal{L}(X, \pi^k, \mu^k), \quad \text{s.t. } X \succeq 0, \tag{42}$$

chooses $\mu^{k+1} \in [\eta\mu^k, \mu^k]$ and then updates the vector of Lagrange multipliers by

$$\pi^{k+1} := \pi^k - \frac{\mathcal{A}(X^k) - b}{\mu^k}, \tag{43}$$

for the next iteration $k+1$. It is important to note that our algorithm does not incorporate the positive semidefinite constraint into the augmented Lagrangian function, and therefore, it is different from the methods in [47, 78].

As is well known (see chapter 12.2 in [25]), (42) is equivalent to minimizing a quadratic penalty function:

$$X^k := \arg\min_X \ \mathcal{F}(X, b^k, \mu^k) := \langle C, X\rangle + \frac{1}{2\mu^k}\|\mathcal{A}(X) - b^k\|_2^2, \ \text{s.t. } X \succeq 0, \tag{44}$$

where $b^k = b + \mu^k\pi^k$ and the difference between $\mathcal{L}(X, \pi^k, \mu^k)$ and $\mathcal{F}(X, b^k, \mu^k)$ is the constant $-\frac{\mu^k}{2}\|\pi^k\|_2^2$. Hence, we consider an alternative version of the augmented Lagrangian method which solves (44) and updates b^k by

$$b^{k+1} := b + \frac{\mu^{k+1}}{\mu^k}\left(b^k - \mathcal{A}(X^k)\right), \tag{45}$$

where $b^1 := b$. We now apply the RBR method to minimize (44). Starting from the point $X^k \succ 0$ at the k-th iteration, the RBR subproblem corresponding to the quadratic SDP (44) that is obtained by fixing all elements of X^k other than those in the i-th row and column results in a minimization problem with two conic constraints. Specifically, we fix the $n(n-1)/2$ variables in the $(n-1)\times(n-1)$ submatrix $B := X^k_{i^c,i^c}$ of X^k and let ξ and y denote the remaining unknown variables $X_{i,i}$ and $X_{i^c,i}$ (i.e., row i/column i), respectively. Hence, the quadratic SDP problem (44) becomes, after, replacing the zero on the right hand side of the Schur complement constraint by $\nu > 0$ to ensure positive definiteness of X^k,

$$\begin{array}{ll} \min\limits_{(\xi;y)\in\mathbb{R}^n} & \widetilde{c}^\top \begin{pmatrix}\xi\\ y\end{pmatrix} + \dfrac{1}{2\mu^k}\left\|\widetilde{A}\begin{pmatrix}\xi\\ y\end{pmatrix} - \widetilde{b}\right\|_2^2 \\ \text{s.t.} & \xi - y^\top B^{-1} y \geq \nu, \end{array} \tag{46}$$

where $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ are given by (6) with b_i for $i = 1, \cdots, m$ replaced by b_i^k. If we let $LL^\top = B$ be the Cholesky factorization of B and introduce a new variable $z = L^{-1}y$, problem (46) can be written as:

$$\begin{array}{ll} \min\limits_{(\xi;z;\tau)} & \widetilde{c}^\top \begin{pmatrix}\xi\\ Lz\end{pmatrix} + \dfrac{1}{2\mu}\tau \\ \text{s.t.} & \left\|\widetilde{A}\begin{pmatrix}\xi\\ Lz\end{pmatrix} - \widetilde{b}\right\|_2^2 \leq \tau \\ & \|z\|_2^2 \leq \xi - \nu. \end{array} \tag{47}$$

Therefore, each step of our RBR augmented Lagrangian method involves solving a SOCP with two rotated second-order cone constraints. We plan to show how advantage can be taken of the particular form of these SOCPs in a future paper. If B is only positive semidefinite, we can derive a similar SOCP by using the spectral decomposition of B. For references on solving SOCPs, see [1] for example. Our combined RBR augmented Lagrangian method for minimizing (3) is presented in Algorithm 2.

Algorithm 2: Row-by-row augmented Lagrangian method

Set $X^1 \succ 0$, $b^1 = b$, $\eta \in (0,1)$, $\nu > 0$, $\mu^1 > 0$, $\epsilon, \epsilon_r, \epsilon_f \geq 0$ and $k := 1$.
Set $F^0 := +\infty$ and compute $F^1 := \langle C, X^1 \rangle$.
while $\frac{F^{k-1}-F^k}{\max\{|F^{k-1}|,1\}} \geq \epsilon$ *or* $\|\mathcal{A}(X^k) - b\|_2 \geq \epsilon_r$ **do**
  Compute $f^1 := \langle C, X^k \rangle + \frac{1}{2\mu^k}\|\mathcal{A}(X^k) - b^k\|_2^2$ and set $f^0 := +\infty$.
  while $\frac{f^{k-1}-f^k}{\max\{|f^{k-1}|,1\}} \geq \epsilon_f$ **do**
    for $i = 1, \cdots, n$ **do**
S1   Set $B := X^k_{i^c,i^c}$ and compute $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ from (6) with $b = b^k$.
S2   Solve the SOCP (46) and denote its solution by ξ and y.
S3   Set $X^k_{i,i} := \xi$, $X^k_{i^c,i} := y$ and $X^k_{i,i^c} := y^\top$.
    Compute $F^k := \langle C, X^k \rangle$ and $f^k := F^k + \frac{1}{2\mu^k}\|\mathcal{A}(X^k) - b^k\|_2^2$.
S4 Update $b^{k+1} := b + \frac{\mu^{k+1}}{\mu^k}\left(b^k - \mathcal{A}(X^k)\right)$.
  Choose $\mu^{k+1} \in [\eta\mu^k, \mu^k]$ and set $X^{k+1} := X^k$ and $k := k+1$.

The RBR method applied to problem (44) converges by Theorem 3 since solving the RBR subproblem (46) essentially corresponds to minimizing the unconstrained function obtained by subtracting $\sigma \log(\xi - y^\top B^{-1} y)$ from the objective function in (46) using an argument analogous to the one made in section 3.4. It is well known that an augmented Lagrangian method applied to minimizing a strictly convex function subject to linear equality constraints, where the minimization of the augmented Lagrangian for each value of the multiplier λ^k (b^k in Algorithm 2) is either done exactly or is asymptotically exact, converges to an optimal solution [4, 5, 52]. Hence, it is clear that a slightly modified version of Algorithm 2 converges to such a solution. For more details for the exact minimization case, we refer the reader to [66].

4.2 Application to Maxcut SDP

Since the constraints in problem (7) are $X_{i,i} = 1$ for $i = 1, \cdots, n$, the quadratic term in the objective function of the RBR subproblem simplifies to

$$\left\| \widetilde{A} \begin{pmatrix} \xi \\ y \end{pmatrix} - \widetilde{b} \right\|^2 = (\xi - b_i^k)^2,$$

and problem (46) reduces to

$$\begin{aligned} \min_{(\xi;y)\in\mathbb{R}^n} \quad & c\xi + \widehat{c}^\top y + \frac{1}{2\mu^k}(\xi - b_i^k)^2 \\ \text{s.t.} \quad & \xi - y^\top B^{-1} y \geq \nu, \end{aligned} \tag{48}$$

where $c := C_{i,i}$, $\widehat{c} := 2C_{i^c,i}$ and $b_i^1 = 1$. The first-order optimality conditions for (48) are

$$\begin{aligned} & \xi = b_i^k + \mu^k(\lambda - c), \quad y = -\frac{1}{2\lambda} B\widehat{c} \\ & \xi \geq y^\top B^{-1} y + \nu, \quad \lambda \geq 0 \quad \text{and} \quad (\xi - y^\top B^{-1} y - \nu)\lambda = 0. \end{aligned}$$

If $\widehat{c} = 0$, then $y = 0$ and $\xi = \max\{\nu, b_i^k - \mu^k c\}$. Otherwise, λ is the unique real root of the cubic equation:

$$\varphi(\lambda) := 4\mu^k\lambda^3 + 4(b_i^k - \mu^k c - \nu)\lambda^2 - \gamma = 0, \tag{49}$$

which is positive. This follows from the continuity of $\varphi(\lambda)$ and the facts that $\varphi(0) = -\widehat{c}^\top B\widehat{c} < 0$, $\lim_{\lambda\to+\infty}\varphi(\lambda) = +\infty$ and

$$\varphi'(\lambda) = 12\mu^k\lambda^2 + 8(b_i^k - \mu^k c - \nu)\lambda \geq 4\mu^k\lambda^2$$

since $\xi = b_i^k - \mu^k c + \mu^k\lambda \geq \nu$, which implies that $\varphi'(0) = 0$ and $\varphi'(\lambda) > 0$ for $\lambda \neq 0$. The RBR augmented Lagrangian method for problem (7) is denoted by ALAG-RBR-M.

4.3 Application to Matrix Completion SDP

Using the notation from section 3.2, the norm of the constraint residual of each RBR subproblem (46) is $\left\|\widetilde{A}\begin{pmatrix}\xi\\ y\end{pmatrix} - \widetilde{b}\right\| = \left\|X_{\alpha,i} - \widetilde{b}\right\| =: \|\widehat{y} - \widetilde{b}\|$, where

$$\widetilde{b} := \begin{cases} (M_{i,\bar{\alpha}}^k)^\top, & \text{if } i \leq p, \\ M_{\alpha,i-p}^k, & \text{if } p < i \leq n, \end{cases} \tag{50}$$

and $M^1 = M$. Therefore, the SOCP (46) becomes

$$\begin{aligned} \min_{(\xi;y)\in\mathbb{R}^n} \quad & \xi + \frac{1}{2\mu^k}\left\|\widehat{y} - \widetilde{b}\right\|_2^2 \\ \text{s.t.} \quad & \xi - y^\top B^{-1} y \geq \nu, \end{aligned} \tag{51}$$

where the matrix $B = \begin{pmatrix} X_{\alpha,\alpha}^k & X_{\alpha,\beta}^k \\ X_{\beta,\alpha}^k & X_{\beta,\beta}^k \end{pmatrix}$.

The optimal solution $[\xi; y] = [\xi; \widehat{y}; \widetilde{y}]$ of (51) must satisfy $\xi = y^\top B^{-1} y + \nu$. Hence, (51) is equivalent to an unconstrained quadratic minimization problem

$$\min_y y^\top B^{-1} y + \frac{1}{2\mu^k} \left\| \widehat{y} - \widetilde{b} \right\|_2^2, \tag{52}$$

whose optimality conditions are

$$\begin{pmatrix} X^k_{\alpha,\alpha} & X^k_{\alpha,\beta} \\ X^k_{\beta,\alpha} & X^k_{\beta,\beta} \end{pmatrix}^{-1} \begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix} + \frac{1}{2\mu^k} \begin{pmatrix} \widehat{y} - \widetilde{b} \\ \mathbf{0} \end{pmatrix} = 0. \tag{53}$$

which implies that

$$\begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix} + \frac{1}{2\mu^k} \begin{pmatrix} X^k_{\alpha,\alpha} \\ X^k_{\beta,\alpha} \end{pmatrix} \widehat{y} = \frac{1}{2\mu^k} \begin{pmatrix} X^k_{\alpha,\alpha} \\ X^k_{\beta,\alpha} \end{pmatrix} \widetilde{b}.$$

Solving for $\widehat{y}$ and $\widetilde{y}$ we obtain $\widetilde{y} = \frac{1}{2\mu^k} X^k_{\beta,\alpha}(\widetilde{b} - \widehat{y})$, where $\widehat{y}$ can be computed from the system of linear equations $\left(2\mu^k I + X^k_{\alpha,\alpha}\right) \widehat{y} = X^k_{\alpha,\alpha} \widetilde{b}$. Then, it follows from $\xi = y^\top B^{-1} y + \nu$ and (53) that $\xi = \frac{1}{2\mu^k} \widehat{y}^\top (\widetilde{b} - \widehat{y}) + \nu$.

The above specialized augmented Lagrangian RBR method for minimizing (20) is denoted by RBR-MC. As in the pure RBR method for the matrix completion problem, we only need to solve a single system of linear equations, whose size is expected to be small for each RBR subproblem (51).

We note that the subproblems that arise when the augmented Lagrangian version of the RBR method is applied to other SDP problems is also solvable in closed form as in the computation of the Lovasz theta function. We did not include a discussion of this or of other SDPs, such as the theta plus problem, that result in subproblems that are rather special quadratic programs, and hence efficiently solvable, to keep the length of this chapter reasonable.

5 An extension of RBR using rank-two updates

In this section, we propose a generalization of the RBR scheme that uses rank-two updates besides those that correspond to modifying a single row and column. This results in a method that also requires solving a sequence of SOCPs. Specifically, given a positive definite matrix $X \succ 0$ and vectors $u, v \in \mathbb{R}^n$, we consider the rank-two update:

$$X_+ = X + \frac{1}{2}(uv^\top + vu^\top). \tag{54}$$

The RBR scheme is a special case of (54), corresponding to $u = e_i$, where e_i is the i-th column of the identity matrix.

By allowing the algorithm to consider different vectors u we significantly increase the flexibility of the BCD scheme to exploit the problem structure. For instance, if $u = (e_i + e_j)/2$ then the BCD will modify the (i, j)-th pair of rows and columns simultaneously. This can be useful, for instance, when the linear constraints are of the form $X_{ii} + X_{jj} - 2X_{ij} = d_{ij}$ as occurs in

SDP relaxations of sensor network localization problems. More generally, one might want to update a whole block of columns and rows at a time because the variables defined in the blocks are closely related to each other via some constraints. For instance, in sparse inverse covariance selection some of the random variables may be known to be directly correlated, hence it makes sense to update the corresponding rows and columns of the covariance matrix in a related manner. In the case of the sensor network localization problem the network may consist of several loosely connected small clusters, for each of which the distance structure is highly constrained. In this case it also makes sense to update the rows and columns related to the whole cluster rather than to individual sensors, while preserving the constraints for the cluster by choosing an appropriate u which keeps the step $uv^\top + vu^\top$ in the nullspace of the chosen subset of constraints.

Alternatively, one may choose u to be the leading eigenvector of the objective function gradient, hence including the steepest descent rank-two direction into the range of possible BCD steps. While a numerically efficient choice of u is likely to be tailored to the specific SDP problem being solved, here we consider the general case.

The positive definiteness of X_+ in (54) can be expressed as a second-order cone constraint for any fixed vector u, given that X is positive definite. To see this, let $X = LL^\top$ be the Cholesky factorization of X, where L is a lower triangular matrix. If we define $y = L^{-1}u$ and $x = L^{-1}v$, then the matrix X_+ can be factorized as $X_+ = LVL^\top$, where $V := I + \frac{1}{2}(yx^\top + xy^\top)$. It can be easily verified that $z_1 := \|y\|_2 x - \|x\|_2 y$ and $z_2 := \|y\|_2 x + \|x\|_2 y$ are eigenvectors of V corresponding to the eigenvalues

$$\lambda_1 := 1 + \frac{1}{2}y^\top x - \frac{1}{2}\|y\|_2\|x\|_2 \quad \text{and} \quad \lambda_2 := 1 + \frac{1}{2}y^\top x + \frac{1}{2}\|y\|_2\|x\|_2, \tag{55}$$

respectively. The eigenvalues other than λ_1 and λ_2 are equal to 1 since V is a rank-two update of the identity. Hence, the matrix V is positive definite if

$$\lambda_1 = 1 + \frac{1}{2}y^\top x - \frac{1}{2}\|y\|_2\|x\|_2 \ > 0, \tag{56}$$

which is equivalent to $2 + u^\top X^{-1}v - \sqrt{(u^\top X^{-1}u)(v^\top X^{-1}v)} \ > 0$. Since X_+ in (54) can be written as $X_+ = X + \bar{u}\bar{u}^\top - \bar{v}\bar{v}^\top$, where $\bar{u} = \frac{1}{2}(u+v)$ and $\bar{v} = \frac{1}{2}(u-v)$, the Cholesky factorization of X_+ can be obtained in $O(n^2)$ operations from two rank-1 updates to the Cholesky factorization of X.

As in the augmented Lagrangian RBR approach for solving (3) described in subsection 4.1, we can incorporate the rank-two update in an augmented Lagrangian framework. Our goal is to solve (44) by iteratively solving subproblems generated by our rank-two updating technique. Given a matrix $X \succ 0$ and a vector u, substituting X_+ for X in (44) and using (54) and (56), we obtain the subproblem

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & \varphi(x) := c^\top x + \dfrac{1}{2\mu^k}\|Bx-d\|_2^2 \\ \text{s.t.} & 2 + y^\top x - \|y\|_2\|x\|_2 \ \geq \sigma, \end{array} \tag{57}$$

where

$$y = L^{-1}u, \quad c := L^\top C u, \quad B := \begin{pmatrix} u^\top A^{(1)} L \\ \cdots \\ u^\top A^{(m)} L \end{pmatrix}, \quad d = b^k - \mathcal{A}(X), \tag{58}$$

for finding $v = Lx$. Note that the subproblem (57) can be formulated as an SOCP with two second-order cone constraints.

In general the matrix B has m rows. The l-th row of B is $u^\top A^{(l)} L$ and hence is equal to 0 if $u^\top A^{(l)} = 0$. As discussed above, u can be chosen to contain only a few nonzeros. For instance, when $u = e_i + e_j$, the only rows of B that are nonzero are those corresponding to $A^{(l)}$ that have nonzero elements in row i or j. In particular, in sensor network localization problems the number of rows in B will equal the number of links that involve nodes i or j; hence, the size of the SOCP cone in the objective function in the subproblem (57) will often be much smaller than the total number of constraints.

We can extend the convergence result stated in Theorem 3 for optimizing the log det analog of Problem (57) to the case of rank-two updates. The theory easily extends if the set of rank-two updates is defined by a finite set of directions u_i, which span R^n and through which the algorithm cycles (as in the case of RBR, where $u_i = e_i$). More generally we can allow an infinite set of directions, but only under some additional restrictions. For instance one such restriction is that the set of limit points of the set of directions is finite and spans R^n. A suitable choice for the set of possible directions is likely to depend on the particular application and is subject to further study.

6 Numerical Results

Although the numerical results that we present in this section are limited to two special classes of SDP problems, they illustrate the effectiveness of our RBR algorithmic framework when it gives rise to easily solved subproblems. Specifically, they show that in these cases, large scale SDPs can be solved in a moderate amount of time using only moderate amount of memory. Moreover, our tests show that the number of cycles taken by our algorithm grows very slowly with the size of the problem.

6.1 The maxcut SDP relaxation

In this subsection, we demonstrate the effectiveness of the RBR methods PURE-RBR-M and ALAG-RBR-M on a set of maxcut SDP relaxation problems and compare them with the general solver DSDP (version 5.8) [3] and a

routine in SDPLR (version 0.130301) [13] developed especially for the maxcut SDP. The DSDP code implements a dual interior point method that is designed to take advantage of the structure of such problems. The SDPLR code implements a low-rank factorization approach. The main parts of our code were written in C Language MEX-files in MATLAB (Release 7.3.0), and all experiments were performed on a Dell Precision 670 workstation with an Intel Xeon 3.4GHZ CPU and 6GB of RAM.

The test problems are based on graphs generated by "rudy", a machine independent graph generator written by G.Rinaldi. Details of the generation including the arguments of "rudy" are provided in [67]. The parameters of DSDP were set to their default values. The tolerance in the code SDPLR was set to 2e-5 and the parameter file "p.maxcut5" was used. The parameter ν in the RBR methods was set to 10^{-6}. We ran PURE-RBR-M with two different tolerances, i.e., ϵ was set to 10^{-3} (moderately accurate) and 10^{-6} (highly accurate), respectively. Similarly, we ran ALAG-RBR-M with two different tolerance settings, that is, $\epsilon, \epsilon_r, \epsilon_f$ were all set to 10^{-1} and 10^{-4}, respectively. For practical considerations, we terminated minimizing each augmented Lagrangian function if the number of cycles was greater than 5. The initial penalty parameter μ^1 in ALAG-RBR-M was set to 5 and was updated by $\mu^{k+1} = \max(0.5\mu^k, 10^{-1})$.

A summary of the computational results obtained by DSDP, SDPLR and PURE-RBR-M is presented in Table 1. In the table, "obj" denotes the objective function of the dual problem computed by DSDP, "rel-obj" denotes the relative error between "obj" and the objective function value computed by either the RBR methods or SDPLR, "CPU" denotes CPU time measured in seconds, and "cycle" denotes the total number of RBR cycles. From Table 1, we can see that our RBR code is able to solve the maxcut SDP relaxation very efficiently. The number of cycles required was almost the same for all of the problems, no matter what their size was. The RBR method was also quite competitive with SDPLR in achieving a relative accuracy of roughly 5×10^{-5} in the objective function value.

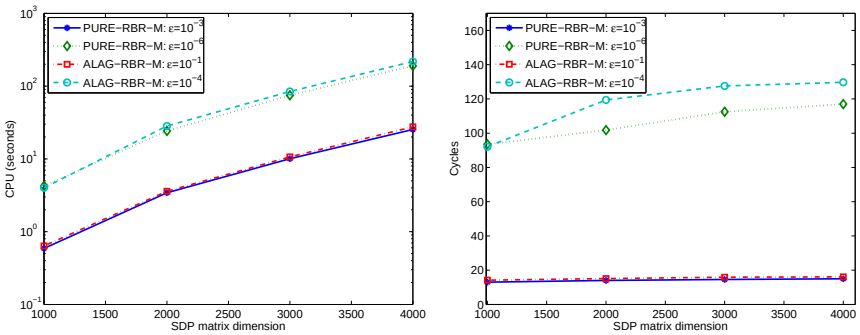
To illustrate the relationship between the computational cost of the RBR methods and the dimension of the SDP matrices, we plot the average of the CPU time versus the dimension in Figure 1 (a) and the average of the number of cycles versus the dimension in Figure 1 (b). Somewhat surprisingly, these plots show that the augmented Lagrangian RBR algorithm solved the maxcut SDP problems almost as efficiently as the pure RBR algorithm for a given relative error. Consequently we did not include test results for ALAG-RBR-M in Table 1.

6.2 Matrix Completion

In this subsection, we evaluate the augmented Lagrangian version of the RBR method (RBR-MC) for the matrix completion problem (20). While the pure RBR method can be directly applied to this problem, preliminary numerical

Table 1. Computational results for the maxcut SDP relaxation.

	DSDP		SDPLR		PURE-RBR-M					
					$\epsilon = 10^{-3}$			$\epsilon = 10^{-6}$		
Name	obj	CPU	rel-obj	CPU	rel-obj	CPU	cycle	rel-obj	CPU	cycle
random graphs										
R1000-1	-1.4e+3	52.6	1.6e-05	4.0	4.9e-3	0.6	13	3.0e-5	3.9	90
R1000-2	-1.4e+3	57.0	6.5e-06	6.0	5.0e-3	0.6	13	3.6e-5	4.1	96
R2000-1	-4.1e+3	607.6	5.1e-05	18.8	5.0e-3	3.9	14	3.7e-5	26.5	97
R2000-2	-4.1e+3	602.3	5.8e-05	19.4	5.2e-3	3.9	14	3.6e-5	27.5	101
R3000-1	-7.7e+3	2576	4.2e-05	38.2	5.0e-3	12.8	15	4.1e-5	90.0	103
R3000-2	-7.7e+3	2606	4.3e-05	42.2	5.2e-3	13.2	15	3.7e-5	89.4	105
R4000-1	-1.2e+4	6274	6.9e-05	64.3	5.9e-3	36.5	15	4.0e-5	261.1	108
R4000-2	-1.2e+4	6310	6.6e-05	63.4	5.7e-3	36.3	15	3.9e-5	265.7	108
random planar graphs										
P1000-1	-1.4e+3	45.1	1.5e-05	6.3	5.0e-3	0.6	13	4.0e-5	4.9	102
P1000-2	-1.4e+3	45.5	8.9e-06	4.6	4.4e-3	0.6	13	2.9e-5	4.2	89
P2000-1	-2.9e+3	386.1	4.4e-06	43.9	5.5e-3	3.0	14	3.7e-5	21.6	102
P2000-2	-2.8e+3	362.8	5.7e-05	19.2	5.8e-3	2.9	14	3.9e-5	22.1	109
P3000-1	-4.3e+3	1400	1.1e-05	49.9	6.0e-3	7.3	15	4.0e-5	56.3	117
P3000-2	-4.3e+3	1394	1.4e-05	62.3	6.5e-3	7.0	14	4.7e-5	57.2	119
P4000-1	-5.7e+3	3688	1.5e-05	122.7	6.5e-3	14.3	15	4.3e-5	114.2	124
P4000-2	-5.9e+3	3253	9.9e-06	123.9	6.5e-3	14.4	15	4.9e-5	116.7	126



(a) CPU Time (b) Cycles

Fig. 1. Relationship between the computational cost and SDP matrix dimension for the maxcut SDP relaxation

testing showed that this approach is much slower (i.e., converges much more slowly) than using RBR-MC, which requires only a small amount of additional work to solve each subproblem than the pure method. It seems that the pure RBR method gets trapped close to the boundary of the semidefinite cone. To overcome this we also tried starting with a very large value of ν (say $\nu = 100$), reducing ν every 20 cycles by a factor of 4 until it reached a value of 10^{-6}. While this improved the performance of the method, the augmented Lagrangian version was still two to four times faster. Hence, we only present results for the latter method. Although we compare RBR-MC with the specialized algorithms, such as SVT [40] and FPCA [46], for the

matrix completion problem (19), our main purpose here is to demonstrate that the RBR method can efficiently solve the SDP problem (20) rather than to compete with those latter algorithms. In fact, the solver LMaFit [69] was consistently much faster than all the methods mentioned above. DSDP is not included in this comparison because it takes too long to solve all problems.

Random matrices $M \in \mathbb{R}^{p\times q}$ with rank r were created by the procedure in [46]. The ratio $m/(pq)$ between the number of measurements and the number of entries in M is denoted by "SR" (sampling ratio). The ratio $r(p+q-r)/m$ of the dimension of a rank r matrix to the number of samples is denoted by "FR". In our tests, the rank r and the number of sampling entries m were taken consistently so that according to the theory in [16] the matrix M is the optimal solution of problem (20). Specifically, FR was set to 0.2 and 0.3 and r was set to 10. We tested five square matrices M with dimensions $p = q \in \{200, \cdots, 500\}$ and set the number m to $r(p+q-r)$/FR. All parameters p, q, r, m and the random seeds "seed" used by the random number generators "rand" and "randn" in MATLAB are reported in Tables 2 and 3.

We ran RBR-MC with two different tolerance settings, i.e., $\epsilon, \epsilon_r, \epsilon_f$ were all set to 10^{-1} and 10^{-3}, respectively. All other parameters of RBR-MC were set to the same values as those used in ALAG-RBR-M. The tolerance parameter "xtol" of FPCA was set to 10^{-6} and all other parameters were set to their default values. We tried many different parameter settings but could not get SVT to work well on all problems. Hence, we only report the results of SVT for the "best" parameter setting that we found, i.e., the parameters "tau" and "delta" and "tol" were set to $5n$, $\min(\max(1.2n^2/p, 1), 3)$ and 10^{-5}, respectively. Summaries of the computational results for FR=0.2 and FR = 0.3 are presented in Tables 2 and 3, respectively. In these tables, rel-X := $\frac{\|X-M\|_F}{\|M\|_F}$ gives the relative error between the true and the recovered matrices. From these tables, we can see that the RBR method can be faster than FPCA when the SDP matrix dimension is small, although usually FPCA is somewhat to as much as twice as fast. However, there is an exception to this in that FPCA took from 20 to 35 times as much CPU time to solve the examples with $p = q = 500$ when FR=0.3 as did the RBR method. To illustrate the relationship between the computational cost of the RBR method and the dimension of the matrices, we plot the average of the CPU time versus the dimension of the SDP matrix (i.e., $p + q$) in Figure 2 (a) and the average of the number of cycles versus this dimension in Figure 2 (b).

7 Summary

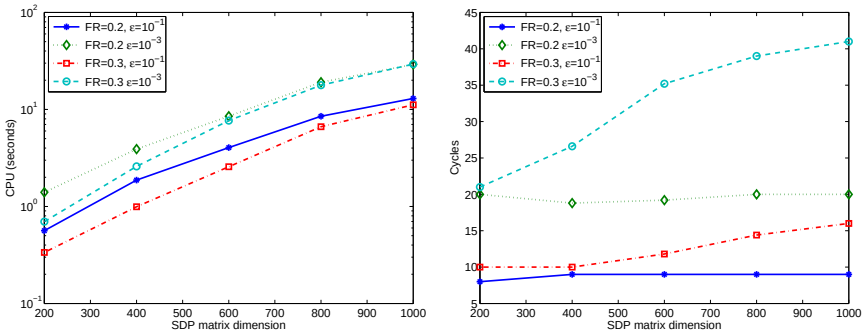
In this chapter, we have shown that RBR block coordinate descent methods can be very effective for solving certain SDPs. In particular, they work extremely well when the subproblem that needs to be solved for each block of variables can be given in closed form, as in SDPs that arise as relaxations of maxcut, matrix completion and Lovasz theta function problems. The RBR

Table 2. Computational results for the matrix completion problem with FR=0.2

	RBR-MC ($\epsilon = 10^{-1}$)			RBR-MC ($\epsilon = 10^{-3}$)			FPCA		SVT	
seed	rel-X	CPU	cycle	rel-X	CPU	cycle	rel-X	CPU	rel-X	CPU
p=q=200; r=10; m=19500; SR=0.49										
68521	7.5e-05	1.8	9	9.4e-07	3.6	17	1.6e-06	2.7	1.6e-05	20.4
56479	6.0e-05	1.9	9	7.4e-07	3.4	17	1.5e-06	2.7	1.6e-05	13.6
p=q=300; r=10; m=29500; SR=0.33										
68521	1.0e-04	3.9	9	1.5e-06	7.2	17	2.2e-06	4.6	1.7e-05	28.2
56479	1.0e-04	4.0	9	1.7e-06	7.5	17	2.1e-06	4.6	1.7e-05	39.0
p=q=400; r=10; m=39500; SR=0.25										
68521	1.0e-04	7.6	9	2.1e-06	14.3	17	2.8e-06	6.0	1.8e-05	28.8
56479	9.9e-05	5.7	9	1.9e-06	10.9	17	2.9e-06	6.0	1.7e-05	28.1
p=q=500; r=10; m=49500; SR=0.20										
68521	2.4e-04	8.8	9	1.8e-06	18.6	19	3.6e-06	10.0	1.9e-05	49.0
56479	1.1e-04	9.0	9	1.5e-06	19.0	19	3.8e-06	10.0	1.8e-05	45.9

Table 3. Computational results for the matrix completion problem with FR=0.3

	RBR-MC ($\epsilon = 10^{-1}$)			RBR-MC ($\epsilon = 10^{-3}$)			FPCA		SVT	
seed	rel-X	CPU	cycle	rel-X	CPU	cycle	rel-X	CPU	rel-X	CPU
p=q=200; r=10; m=13000; SR=0.33										
68521	1.0e-03	1.0	10	4.4e-06	2.3	24	3.3e-06	8.4	5.9e-04	96.7
56479	1.5e-03	1.0	10	6.8e-06	2.3	24	3.0e-06	8.5	1.3e-03	88.7
p=q=300; r=10; m=19666; SR=0.22										
68521	1.0e-03	2.4	11	3.7e-06	5.5	26	3.4e-06	4.9	2.7e-03	180.6
56479	3.3e-04	2.6	12	2.2e-06	5.9	27	3.6e-06	4.5	2.5e-03	230.8
p=q=400; r=10; m=26333; SR=0.16										
68521	9.8e-03	6.1	15	9.9e-04	16.2	40	4.8e-06	10.4	2.8e-02	418.8
56479	3.0e-03	5.6	14	4.6e-06	11.2	28	4.0e-06	16.1	1.5e-02	374.7
p=q=500; r=10; m=33000; SR=0.13										
68521	5.3e-03	11.4	16	6.9e-06	20.1	29	6.1e-06	223.4	2.7e-02	675.0
56479	5.4e-03	11.0	16	5.9e-06	21.3	32	6.2e-06	212.8	2.8e-02	667.4



(a) CPU Time (b) Cycles

Fig. 2. Relationship between the computational cost and SDP matrix dimension for SDP matrix completion

method is also effective when the RBR subproblems can be formulated as simple quadratic programming problems, as in sparse inverse covariance estimation, the computation of the Lovasz theta plus function, and relaxations of the maximum k-cut and bisection problems.

Like all first-order methods, they are best suited to situations where highly accurate solutions are not required. As is the case for BCD and coordinate descent methods in general, only constraints that do not couple different variable blocks can be handled directly. For more general linear constraints, the RBR approach has to be incorporated into an augmented Lagrangian framework. Our numerical testing has shown that even problems in which the constraints do not couple variables from different blocks, it still may be advantageous to employ an augmented Lagrangian approach, since this gives the method more freedom of movement. In addition, starting very close to the boundary of the semidefinite cone, especially when combined with linear constraints that very tightly limit the size of steps that the RBR method can take, can result in very slow rates of convergence.

Finally, we have shown that the RBR approach can be generalized to accommodate rank-two updates other than those that correspond to modifying a single row and column.

Acknowledgement

The work was supported in part by NSF Grants DMS-0439872, DMS 06-06712, DMS 10-16571, ONR Grant N00014-08-1-1118 and DOE Grant DE-FG02-08ER58562. The authors would like to thank Shiqian Ma for his help in writing and testing the codes, especially for the matrix completion problem, and the editors and an anonymous referee for their valuable comments and suggestions.

References

1. F. Alizadeh and D. Goldfarb. Second-order cone programming. *Math. Programming*, 95(1, Ser. B):3–51, 2003. ISMP 2000, Part 3 (Atlanta, GA).
2. Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.*, 9:485–516, 2008.
3. Steven J. Benson, Yinyu Ye, and Xiong Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10(2):443–461, 2000.
4. Dimitri P. Bertsekas. Necessary and sufficient condition for a penalty method to be exact. *Math. Programming*, 9(1):87–99, 1975.
5. Dimitri P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press Inc., New York, 1982.

6. Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, September 1999.
7. Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
8. L. Bo and C. Sminchisescu. Greedy Block Coordinate Descent for Large Scale Gaussian Process Regression. In *Uncertainty in Artificial Intelligence*, 2008.
9. Charles A. Bouman and Ken Sauer. A unified approach to statistical tomography using coordinate descent optimization. *IEEE Transactions on Image Processing*, 5:480–492, 1996.
10. Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004.
11. Samuel Burer. Optimizing a polyhedral-semidefinite relaxation of completely positive programs. Technical report, Department of Management Sciences, University of Iowa, 2008.
12. Samuel Burer and Renato D. C. Monteiro. A projected gradient algorithm for solving the maxcut SDP relaxation. *Optim. Methods Softw.*, 15(3-4):175–200, 2001.
13. Samuel Burer and Renato D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95(2, Ser. B):329–357, 2003.
14. Samuel Burer and Renato D. C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Math. Program.*, 103(3, Ser. A):427–444, 2005.
15. Samuel Burer and Dieter Vandenbussche. Solving lift-and-project relaxations of binary integer programs. *SIAM J. Optim.*, 16(3):726–750, 2006.
16. Emmanuel J. Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, May 2009.
17. Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. Coordinate descent method for large-scale L2-loss linear support vector machines. *J. Mach. Learn. Res.*, 9:1369–1398, 2008.
18. Gong Chen and Marc Teboulle. A proximal-based decomposition method for convex minimization problems. *Math. Programming*, 64(1, Ser. A):81–101, 1994.
19. Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 922–931, 2008.
20. L. F. Diachin, P. Knupp, T. Munson, and S. Shontz. A comparison of inexact newton and coordinate descent mesh optimization techniques. In *13th International Meshing Roundtable, Williamsburg, VA, Sandia National Laboratories*, pages 243–254, 2004.
21. Jonathan Eckstein and Dimitri P. Bertsekas. An alternating direction method for linear programming. LIDS-P, 1967. Laboratory for Information and Decision Systems, MIT.
22. Jonathan Eckstein and Dimitri P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming*, 55(3, Ser. A):293–318, 1992.
23. Michael Elad, Boaz Matalon, and Michael Zibulevsky. Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization. *Appl. Comput. Harmon. Anal.*, 23(3):346–367, 2007.

24. Jeffrey A. Fessler. Grouped coordinate descent algorithms for robust edge-preserving image restoration. In *in Proc. SPIE 3071, Im. Recon. and Restor. II*, pages 184–94, 1997.
25. R. Fletcher. *Practical methods of optimization*. A Wiley-Interscience Publication. John Wiley & Sons Ltd., Chichester, second edition, 1987.
26. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9:432–441, 2008.
27. Katsuki Fujisawa, Masakazu Kojima, and Kazuhide Nakata. Exploiting sparsity in primal–dual interior-point methods for semidefinite programming. *Math. Programming*, 79(1-3, Ser. B):235–253, 1997.
28. Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995.
29. D. Goldfarb and G. Iyengar. Robust convex quadratically constrained programs. *Math. Program.*, 97(3, Ser. B):495–515, 2003. New trends in optimization and computational algorithms (NTOC 2001) (Kyoto).
30. D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Math. Oper. Res.*, 28(1):1–38, 2003.
31. Donald Goldfarb and Shiqian Ma. Fast alternating linearization methods for minimizing the sum of two convex functions. Technical report, IEOR, Columbia University, 2009.
32. Donald Goldfarb and Shiqian Ma. Fast multiple splitting algorithms for convex optimization. Technical report, IEOR, Columbia University, 2009.
33. L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Oper. Res. Lett.*, 26(3):127–136, 2000.
34. Luigi Grippo and Marco Sciandrone. Globally convergent block-coordinate techniques for unconstrained optimization. *Optim. Methods Softw.*, 10(4):587–637, 1999.
35. Wolfgang Hackbusch. *Multigrid methods and applications*, volume 4 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1985.
36. B. S. He, H. Yang, and S. L. Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *J. Optim. Theory Appl.*, 106(2):337–356, 2000.
37. Bingsheng He, Li-Zhi Liao, Deren Han, and Hai Yang. A new inexact alternating directions method for monotone variational inequalities. *Math. Program.*, 92(1, Ser. A):103–118, 2002.
38. C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10(3):673–696, 2000.
39. Fang-Lan Huang, Cho-Jui Hsieh, Kai-Wei Chang, and Chih-Jen Lin. Iterative scaling and coordinate descent methods for maximum entropy. In *ACL-IJCNLP '09: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 285–288, Morristown, NJ, USA, 2009. Association for Computational Linguistics.
40. Cai Jian-Feng, Emmanuel J. Candes, and Shen Zuowei. A singular value thresholding algorithm for matrix completion export. *SIAM J. Optim.*, 20:1956–1982, 2010.
41. Krzysztof C. Kiwiel, Charles H. Rosa, and Andrzej Ruszczyński. Proximal decomposition via alternating linearization. *SIAM J. Optim.*, 9(3):668–689, 1999.
42. Tamara G. Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Rev.*, 45(3):385–482, 2003.

43. Spyridon Kontogiorgis and Robert R. Meyer. A variable-penalty alternating directions method for convex optimization. *Math. Programming*, 83(1, Ser. A):29–53, 1998.
44. Yingying Li and Stanley Osher. Coordinate descent optimization for ℓ^1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Probl. Imaging*, 3(3):487–503, 2009.
45. Z. Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *J. Optim. Theory Appl.*, 72(1):7–35, 1992.
46. S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*.
47. Jerome Malick, Janez Povh, Franz Rendl, and Angelika Wiegele. Regularization methods for semidefinite programming. *SIAM Journal on Optimization*, 20(1):336–356, 2009.
48. Rahul Mazumder, Jerome Friedman, and Trevor Hastie. Sparsenet: Coordinate descent with non-convex penalties. Technical report, Stanford University, 2009.
49. Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. Technical report, CORE Discussion Paper, 2010.
50. J. Povh, F. Rendl, and A. Wiegele. A boundary point method to solve semidefinite programs. *Computing*, 78(3):277–286, 2006.
51. M. J. D. Powell. On search directions for minimization algorithms. *Math. Programming*, 4:193–201, 1973.
52. R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.*, 1(2):97–116, 1976.
53. Yousef Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.
54. Ankan Saha and Ambuj Tewari. On the finite time convergence of cyclic coordinate descent methods. Technical report, University of Chicago, 2010.
55. Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for l1 regularized loss minimization. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 929–936. ACM, 2009.
56. Xue-Cheng Tai and Jinchao Xu. Global and uniform convergence of subspace correction methods for some convex optimization problems. *Math. Comp.*, 71(237):105–124, 2002.
57. M. J. Todd. Semidefinite optimization. *Acta Numer.*, 10:515–560, 2001.
58. P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494, 2001.
59. P. Tseng and S. Yun. Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *J. Optim. Theory Appl.*, 140(3):513–535, 2009.
60. Paul Tseng. Dual coordinate ascent methods for non-strictly convex minimization. *Math. Programming*, 59(2, Ser. A):231–247, 1993.
61. Paul Tseng. Alternating projection-proximal methods for convex programming and variational inequalities. *SIAM J. Optim.*, 7(4):951–965, 1997.
62. Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training. *Comput. Optim. Appl.*, 2007.
63. Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program.*, 117(1-2, Ser. B):387–423, 2009.

64. Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.
65. Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imaging Sci.*, 1(3):248–272, 2008.
66. Zaiwen Wen. *First-order methods for semidefinite programming*. PhD thesis, Columbia University, 2009.
67. Zaiwen Wen, Donald Goldfarb, Shiqian Ma, and Katya Scheinberg. Row by row methods for semidefinite programming. Technical report, Dept of IEOR, Columbia University, 2009.
68. Zaiwen Wen, Donald Goldfarb, and Wotao Yin. Alternating direction augmented lagrangian methods for semidefinite programming. Technical report, Dept of IEOR, Columbia University, 2009.
69. Zaiwen Wen, Wotao Yin, and Yin Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Technical report, Rice University, 2010. CAAM Technical Report TR10-07.
70. Henry Wolkowicz, Romesh Saigal, and Lieven Vandenberghe, editors. *Handbook of semidefinite programming*. Kluwer Academic Publishers, Boston, MA, 2000.
71. Junfeng Yang and Yin Zhang. Alternating direction algorithms for l1-problems in compressive sensing. Technical report, Rice University, 2009.
72. Caihong Ye and Xiaoming Yuan. A descent method for structured monotone variational inequalities. *Optimization Methods and Software*, 22:329–338, 2007.
73. Jong Chul Ye, Kevin J. Webb, Charles A. Bouman, and R. P. Millane. Optical diffusion tomography by iterative-coordinate-descent optimization in a bayesian framework. *J. Opt. Soc. Am. A*, 16(10):2400–2412, 1999.
74. Zhensheng Yu. Solving semidefinite programming problems via alternating direction methods. *J. Comput. Appl. Math.*, 193(2):437–445, 2006.
75. Sangwoon Yun and Kim-Chuan Toh. A coordinate gradient descent method for l1-regularized convex minimization. *Comput. Optim. Appl.*, 2009.
76. Fuzhen Zhang, editor. *The Schur complement and its applications*, volume 4 of *Numerical Methods and Algorithms*. Springer-Verlag, New York, 2005.
77. Yin Zhang. User's guide for yall1: Your algorithms for l1 optimization. Technical report, Rice University, 2009.
78. Xinyuan Zhao, Defeng Sun, and Kimchuan Toh. A Newton-CG augmented lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20:1737–1765, 2010.