A RANK PRESERVING FLOW ALGORITHM FOR QUADRATIC OPTIMIZATION PROBLEMS SUBJECT TO QUADRATIC EQUALITY CONSTRAINTS

John B. Moore
Department of Systems Engineering
Research School of Information Sciences and Engineering
Australian National University
ACT 0200, Australia

*Danchi Jiang**
Department of Mechanical and Automation Engineering
Chinese University of Hong Kong
Shatin, N. T. Hong Kong
Email: dcjiang@mae.cuhk.edu.hk

ABSTRACT

This paper concerns quadratic programming problems subject to quadratic equality constraints such as arise in broadband antenna array signal processing and elsewhere. At first, such a problem is converted into a semidefinite programming problem with a rank constraint. Then, a rank preserving flow is used to accommodate the rank constraint. The associated gradient formulas are carefully developed. The convergence of the resulted algorithm is also guaranteed. Our approach is demonstrated by a numerical experiment.

1 PROBLEM DESCRIPTION

Consider the following general quadratic programming problem:

$$\begin{aligned} \min \quad & J_0(X) := \operatorname{tr}(X^\top Q_0 X + B_0 X) & (1) \\ \text{subject to:} \quad & J_i(X) := \operatorname{tr}(X^\top Q_i X + B_i X) = c_i, & \\ & i = 1, 2, \ldots, m. & (2) \end{aligned}$$

where $X \in \Re^{p\times q}$, Q_0 is a positive definite matrix, and $Q_i, i = 1, 2, \ldots, m$ are positive semi-definite matrices. A linear constraint is covered as a special case where the matrix Q_i for the corresponding index i is a zero matrix.

For given generic $Q_i, i = 1, 2, \ldots, m$, it is a difficult task to solve the problem (1) (2). One of the main reasons is that the admissible set in the generic case is disconnected. Hence, any gradient based methods for the searching of the optimal solution is bound to lead to a local optimal. Another reason is that, eventhough one can use a gradient based method to solve it, the computation of the gradient of the cost function is complicated for problems of large size. Also because this problem is not convex, quadratic equality constraints can not be touched by the popular interior-point polynomial methods such as reported in [1], [3] and [4].

However, this quadratic optimization problem arises in applications. For instance, in the area of broad band antenna array signal processing, the global minimum of such a problem leads to an optimal design. See [2] for details. It is of interest to developed appropriate algorithm to solve this problem.

In this paper, motivated by a desire to achieve improved optimization techniques, we convert the quadratic programming problem into an optimization problem of a linear matrix function subject to linear equality, linear matrix inequality, and matrix rank constraints. Then, an algorithm is developed to solve the converted problem based on gradient flow with respect to certain Riemannian metric.

The paper is divided into four sections. Section 1 introduces the problem of interest. Section 2 converts the problem into an optimization problem for linear matrix function subject to linear equality, linear matrix inequality, and matrix rank constraints. The related convex problem is also discussed. Sections 3 is devoted to the development of an algorithm. The variable evolves in a set of positive semi-definite matrices of a fixed rank. Section 4 contains a numerical experiment using the algorithm developed.

2 CONVERTED PROBLEMS

First, we note the following lemma.

Lemma 1 *The matrix equality*

$$Y = XX^\top \qquad (3)$$

is equivalent to

$$Z := \begin{pmatrix} Y & X \\ X^\top & I_q \end{pmatrix} \geq 0, \quad \operatorname{rank}\begin{pmatrix} Y & X \\ X^\top & I_q \end{pmatrix} = q. \qquad (4)$$

*Partly supported by the funding of the activities of the Cooperative Research Centre for Robust and Adaptive Systems by the Australian Government under the Cooperative Research Centres Program.

We also note that the original problem is equivalent to the problem defined as follows:

$$\min \quad J_0 = \mathrm{tr}\,(Q_0 Y + B_0 X) \tag{5}$$

$$\text{subject to:} \quad \begin{cases} J_i = \mathrm{tr}(Q_i Y + B_i X), \\ i = 1, 2, \ldots, m. \\ Y = XX^\top \text{ i.e. (3) holds.} \end{cases} \tag{6}$$

The original problem is equivalent to the following:

$$\min \quad \mathrm{tr}(K_0 Z) \tag{7}$$

$$\text{subject to:} \quad \begin{cases} \mathrm{tr}(K_i Z) = c_i, \quad i = 1, 2, \ldots, m. \\ Z^\top = Z, \quad Z \geq 0, \quad \mathrm{rank}(Z) = q, \\ \mathrm{diag}(0, I_q) Z \mathrm{diag}(0, I_q) = \mathrm{diag}(0, I_q) \end{cases} \tag{8}$$

where

$$K_i = \begin{pmatrix} Q_i & \frac{1}{2} B_i^\top \\ \frac{1}{2} B_i & 0 \end{pmatrix}, \quad i = 0, 1, \ldots, m. \tag{9}$$

To simplify the formulation of (8), note that the last equality constraint can be converted into a group of equality constraints on the trace of some linear matrix function. More specifically, it is equivalent to the following $\frac{q(q+1)}{2}$ equality constraints:

$$\mathrm{tr}(e_j^\top e_j Z) = 1, \quad j = p+1, p+2, \ldots, p+q. \tag{10}$$

$$\mathrm{tr}([e_u + e_v]^\top [e_u + e_v] Z) = 2, \quad p + 1 \leq u, v \leq p + q, \quad u \neq v \tag{11}$$

where e_j is the the j-th elementary column vector whose j-th component is 1 and other components are zero. Let $c_i = 1$ for $m + 1 \leq i \leq m + q$, $c_i = 2$ for $m + q + 1 \leq i \leq L$, where $L := m + \frac{q(q+1)}{2}$ and let the K_i be the corresponding coefficient matrices in the equalities (10) and (11). The optimization problem defined by (7) (8) now is converted into the following form:

$$\min \quad \mathrm{tr}(K_0 Z) \tag{12}$$

$$\text{subject to:} \quad \begin{cases} \mathrm{tr}(K_i Z) = c_i, \quad i = 1, 2, \ldots, L \\ Z^\top = Z, \quad Z \geq 0. \end{cases} \tag{13}$$

$$\text{and:} \quad \mathrm{rank}(Z) = q. \tag{14}$$

This problem is easy to solve if the rank condition is removed. In fact, the optimal problem defined by (12) and (13) is a standard semi-definite programming problem. As an important class of convex problem, it has been extensively studied recently and is known to be solved by interior-point methods in polynomial time with respect to the size of the problem. For details, see [1]. It is also referred to as a linear matrix inequality (LMI).

The following results concern properties of the corresponding solution.

Theorem 1 *Suppose $X^\star$ is the optimal solution to the problem (12) (13).*

(1). *If $X^\star$ is of full rank, then, K_0 can be represented as a linear combination of K_i, $i = 1, 2, \ldots, L$. i.e., there exist real numbers k_i, $i = 1, 2, \ldots, L$ such that $K_0 = \sum_{i=1}^{L} k_i K_i$. In this case, the cost (12) is invariant for all admissible solutions.*

(2). *If $X^\star$ is not of full rank, let N_j, $j = 1, \ldots, r$ for integer $r = p + q - \mathrm{rank}(X^\star)$ be independent null vectors of it. Then, K_0 is a linear combination of K_i, $i = 1, 2, \ldots, L$ and $N_i N_i^\top$, $i = 1, \ldots, r$. In this case, all positive semi-definite matrices Z that satisfy the following conditions*

$$\begin{aligned} \mathrm{tr}(K_i Z) &= c_i, \quad i = 1, 2, \ldots, L \\ \mathrm{tr}(N_i N_i^\top Z) &= 0, \quad i = 1, \ldots, r. \end{aligned}$$

are the optimal solutions and have the same cost.

(3). *If the optimal solution for the problem (12) (13) is not unique, there are a group of common zero eigenvectors $N_i^\star, i = 1, 2, \ldots, s$ for some positive integer s such that (a) $K_i, i = 1, 2, \ldots, m$ and $N_i^\star, i = 1, 2, \ldots, s$ are independent and (b) K_0 is a linear combination of $K_i, i = 1, 2, \ldots, m$ and $N_i^\star, i = 1, 2, \ldots, s$.*

Proof (1). Since $X^\star$ is of full rank, there exists a neighborhood of it where all symmetric matrices are positive definite. Furthermore, there exists a positive real number $\delta > 0$, such that, for all symmetric matrix H satisfying

$$\| H \|_F = \sqrt{HH'} \leq \delta, \quad \mathrm{tr}(K_i H) = 0, i = 1, 2, \ldots, L,$$

$(X^\star + H)$ is an admissible solution for the constraints (13) and $\mathrm{tr}(K_0(X^\star + H)) \geq \mathrm{tr}(K_0 X^\star)$. This leads to $\mathrm{tr}(K_0 H) = 0$. Therefore, K_0 is a linear combination of $K_i, i = 1, 2, \ldots, L$. Obviously, the cost of any admissible point is also the corresponding linear combination of c_i, and hence a constant.

(2). Let $N^\perp$ be defined as:

$$N^\perp = \{Y \in \Re^{(p+q)\times(p+q)} : Y^\top = Y, YN_i = 0, i = 1, \ldots, r.\}.$$

Then, it is straight forward to show that there is a positive real number $\tilde{\delta} > 0$ such that, for all $H \in N^\perp$, if $\| H \|_F \leq \tilde{\delta}$, then, $X^\star + H$ is positive semi-definite. Following the argument for **(1)** we obtain the validity of **(2)**.

(3). First, we claim that for two optimal solutions X_1 and X_2, $\frac{1}{2}(X_1 + X_2)$ is also an optimal solution and, furthermore, only a common zero eigenvector of X_1 and X_2 can be its zero eigenvector.

The first part of the claim is clearly implied by the convexity of the problem and its constraints. For the second part, assume V is a zero eigenvector of $\frac{1}{2}(X_1 + X_2)$. Then, $V^\top(X_1+X_2)V = 0$. Since both X_1 and X_2 are positive semi-definite, $V^\top X_1 V = 0$ and $V^\top X_2 V = 0$. Therefore, V is a common zero eigenvector of X_1 and X_2.

Hence, if the property of (3) does not hold, there exists a group of optimal solutions $Z_i, i = 1, 2, \ldots, h$ for some positive integer h such that their common eigenvectors, denoted as $U_j, j = 1, \ldots, g$, together with $K_i, i = 1, 2, \ldots, m$ can not be organized as a linear representation for K_0. Similarly, we can show that $\frac{1}{h}(Z_1 + Z_2 + \ldots + Z_h)$ is an optimal solution and $U_j, j = 1, \ldots, g$ are a group of its zero eigenvectors. This is a contradiction to the results in (2). Hence the proof is complete. □

It is possible that the optimal solution to the convex programming problem defined by (12) and (13) has p zero eigenvectors. If it is the case, based on Theorem 1, the optimal solution of the convex problem and that of the optimization problem defined by (12), (13) and (14) coincide. However, numerical results conducted by the authors imply that it is not always the case. In the next two sections, an algorithm will be developed to search for a solution to the problem subject to a rank constraint.

3 RANK PRESERVING FLOW ALGORITHM

Let the set $P(q)$ denote all $(p+q)$-dimensional positive semi-definite matrices of rank q. From Proposition 1.1 in [5, page 134], we know that $P(q)$ is a connected smooth manifold and its tangent space is calculated as:

$$T_Z P(q) = \{\Delta Z + Z\Delta^\top \mid \Delta \in \Re^{(p+q)\times(p+q)}\}.$$

First of all, let us show that the cost function (12) has compact sublevel sets. Since K_0 is in the form of $\begin{pmatrix} Q_0 & \frac{1}{2}B_0^\top \\ \frac{1}{2}B_0 & 0 \end{pmatrix}$, one can always choose a positive definite matrix C of an appropriate dimension such that $\bar{K} := \begin{pmatrix} Q_0 & \frac{1}{2}B_0^\top \\ \frac{1}{2}B_0 & C \end{pmatrix}$ is positive definite. Therefore, $\mathrm{tr}(K_0 Z) = \mathrm{tr}(\bar{K}Z) - \mathrm{tr}(C)$, bearing in mind that Z is in the form of $\begin{pmatrix} Y & X \\ X^\top & I_q \end{pmatrix}$. Hence, the fact that $\mathrm{tr}(K_0 Z)$ is bounded implies that Z is bounded.

Based on the compactness of sublevel sets of the cost function J_0 defined by (12), the following two properties concerning the optimal point set hold:

- The set of all optimal point contains at most finite number of connected closed branches. These branches are isolated.
- Any algorithm, as long as it guarantees the descent of the cost function J_0, will converge to one of those connected branches.

Now we are going to compute the gradient of the cost function. At any point $Z \in P(q)$, decompose $\Re^{(p+q)\times(p+q)}$ as $S \bigoplus S^\perp$ such that S is the kernel of the linear map:

$$\pi : \Re^{(p+q)\times(p+q)} \mapsto \Re^{(p+q)\times(p+q)}, \qquad \pi(\Delta) = \Delta Z + Z\Delta^\top. \tag{15}$$

Denote Pr as the corresponding projection such that $Pr(S) = 0$. Define a Riemannian metric as:

$$\ll \Delta_1, \Delta_2 \gg := 2\mathrm{tr}\{[Pr(\Delta_1)]^\top Pr(\Delta_2)\}. \tag{16}$$

Since

$$DJ_0\,|_Z\,(\Delta) = \mathrm{tr}\{K_0(\Delta Z + Z\Delta^\top)\} = 2\mathrm{tr}(ZK_0\Delta),$$

the gradient of J_0 associated with the Riemannian metric defined by (16) is calculated as:

$$\mathrm{grad}J_0 = K_0 Z^2 + Z^2 K_0. \tag{17}$$

The projected gradient onto the constrained surface by (13) can be calculated as:

$$\mathrm{Grad}J_0 = (K_0 - \sum_{i=1}^{L} k_i K_i)Z^2 + Z^2(K_0 - \sum_{i=1}^{L} k_i K_i), \tag{18}$$

where k_i satisfy:

$$\sum_{i=1}^{L} k_i \mathrm{tr}\{[Pr(K_i Z)]^\top Pr(K_j Z)\} = \mathrm{tr}\{[Pr(K_0 Z)]^\top Pr(K_j Z)\}, \quad j = 1, 2, \ldots, L.$$

The associated negative gradient flow is defined as:

$$\dot{Z} = -\mathrm{Grad}J_0. \tag{19}$$

Along any trajectory of this flow, the cost function J_0 always decreases until arriving an equilibrium point.

4 NUMERICAL SIMULATION

In this section, we conduct a numerical experiment using the gradients developed in Sections 3. The parameters for the problem defined by (1) and (2) are chosen by random number generator *randn* in Matlab as:

$$Q0 = \begin{pmatrix} 0.3180 & 1.6065 & -0.9235 \\ -0.5112 & 0.8476 & -0.0705 \\ -0.0020 & 0.2681 & 0.1479 \end{pmatrix},$$

$$Q1 = \begin{pmatrix} 1.5578 & 1.1226 & 0.4142 \\ -2.4443 & 0.5817 & -0.9778 \\ -1.0982 & -0.2714 & -1.0215 \end{pmatrix},$$

$$Q2 = \begin{pmatrix} -0.5077 & -0.7262 & -0.2091 \\ 0.8853 & -0.4450 & 0.5621 \\ -0.2481 & -0.6129 & -1.0639 \end{pmatrix},$$

$$B0 = \begin{pmatrix} -0.5571 \\ -0.3367 \\ 0.4152 \end{pmatrix}; \quad B1 = \begin{pmatrix} 0.3177 \\ 1.5161 \\ 0.7494 \end{pmatrix}, \quad C_1 = 0;$$

$$B2 = \begin{pmatrix} 0.3516 \\ 1.1330 \\ 0.1500 \end{pmatrix}, \quad C_2 = 0,$$

for $X \in \Re^3, m = 2$. The parameter matrices K_0, K_1, K_2 is calculated based on these parameters.

For the Rank Preserving Flow, we compute the Projection operator first.
$vec(\pi\Delta) = \mathcal{A}vec(\Delta) := (Z^\top \otimes I + (I \otimes Z)P(4,4))vec(\Delta)$, where $vec(\Delta)$ is the column vector where the column vectors of Δ are stacked in order, $\otimes$ is the matrix Kronecker product, and $P(4,4)$ is the fourth-order permutation matrices defined as:

$$P(4,4) := \sum_{i=1}^{4} \sum_{j=1}^{4} E_{i,j} \otimes E_{i,j}^\top,$$

and $E_{i,j}$ is the matrix that the (i,j)-th component is 1 and the component elsewhere is zero. Therefore,

$$vec(Pr(S)) = \mathcal{A}^{+}\mathcal{A}vec(S),$$

where $\mathcal{A}^{+}$ is the pseudo-inverse of the matrix $\mathcal{A}$.

Then, use ODE23 in Matlab to search for an optimal solution. The initial condition is chosen as

$$Z0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

In Figure 1, we can see that, eventhough the cost function is not convex, the algorithm still converge quickly. The solution obtained is $X = (0.0286, -0.0305, -0.2124)^\top$. The corresponding cost is $J_0 = -0.0888$.

Another possible approach is to penalize the rank of the symmetric matrix Z in the semidefinite programming problem by some means so as to obtain the minimal rank one. This approach may result in a convex problem. Further detail is to be investigated.

References

[1] Y. Nesterov and A. Nemirovsky (1994), Interior-point polynomial methods in convex programming. *Studies in Applied Mathematics, SIAM*, 13.

[2] I. Thng, A. Cantoni and Y. H. Leung (1994), Analytical solutions to the optimization of a quadratic cost function subject to linear and quadratic equality constraints. to appear in *Applied Mathematics and Optimizations*.

[3] F. Jarre (1991), On the convergence of the method of analytic centers when applied to convex quadratic programs. *Mathematical Programming* 49, 341-358.

[4] S. Mehrotra and J. Sun (1991), A method of analytic centers for quadratically constrained convex quadratic programs. *SIAM J. Numer. Anal.* Vol. 28, No. 2, pp. 529-544.

[5] U. Helmke and J. B. Moore (1994), *Optimization and Dynamical Systems* Springer-Verlag, London.

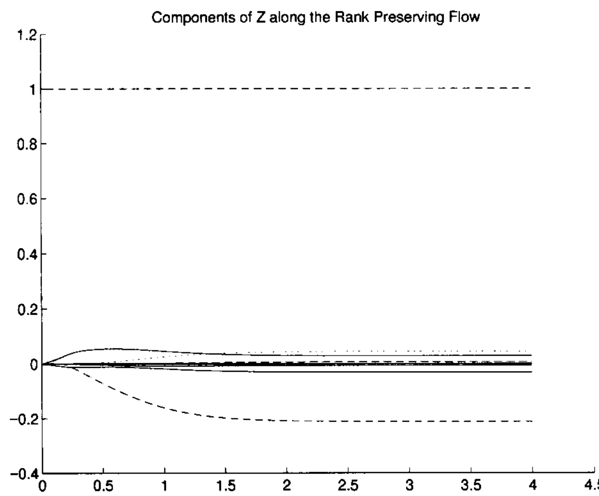

Figure 1. The convergence of all components of Z along the Rank Preserving Flow.

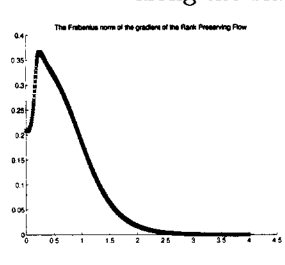

Figure 2. The Frobenius norm of the gradient along the Rank Preserving Flow

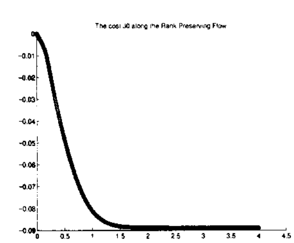

Figure 3. The cost J_0 along the Rank Preserving Flow