Computer Science Journal of Moldova, vol.**8**, no.2(23), 2000

Primal-Dual Method of Solving Convex Quadratic Programming Problems

V. Moraru

Abstract

This paper presents a primal-dual method for solving quadratic programming problems. The method is based on finding an exact solution of a finite sequence of unconstrained quadratic prigraamming problems and on finding an aproximative solution of constrained minimization problem with simple constraints. The subproblem with simple constraints is solved by the interior-reflective Newton's method [6].

Key words: Quadratic Programming, Primal-Dual Method, Newton's Method, Semidefinite Programming.

1 Introduction

The quadratic programming problems are very important for constrained optimization. Among the known methods of solving of a general nonlinear programming problem, a special place is occupied by Newton and quasi-Newton methods. They are based on sequential quadratic programming [1-3]. In these methods, the optimal solution is being determined as a consequence on an iterative process, during which a quadratic programming problem is solved at each step. That is why the efficiency of the methods of solving quadratic programming problems based on sequential quadratic programming (SQP), depends mainly on the efficiency of the algorithm of solving quadratic programming subproblems.

There is a great number of algorithms of solving quadratic programming problems. A relative complete bibliography of these methods can be found in [4]. In the case when we have a quadratic programming problem with a big number of variables and constraints, it is necessary to execute a relatively big number of arithmetical operations for the solution of the problem. In such situations, it is more convenient to solve a finite succession of quadratic programming problems without any constrained or with simple bounds, instead of the solution of the considered problem. Such a method is presented in this work.

We consider a quadratic programming problem written in the following form:

$$f(x) = \frac{1}{2}x^T Hx + g^T x \rightarrow \min \tag{1.1}$$

$$\text{subject to linear constraints } Ax \leq b,$$

where H is a symmetric matrix, positive definite of the $n \times n$ dimension, A is a $m \times n$ matrix, g, x and b are column vectors, g and $x \in \Re^n$, and $b \in \Re^m$. The symbol $'T'$ denotes the operation of transposition.

It is well known (see for example [2,3,5]) that the optimal solution x_* of the problem (1.1) is defined by the relation:

$$x_* = -H^{-1}(A^T \lambda_* + g). \tag{1.2}$$

The Lagrange multipliers vector $\lambda_* = (\lambda_*^1, \lambda_*^2, \ldots, \lambda_*^m)^T$ is the solution of the dual problem:

$$\varphi(\lambda) = \frac{1}{2}\lambda^T D\lambda + c^T \lambda \rightarrow \min \tag{1.3}$$

$$\text{subject to } \lambda^i \geq 0, i = 1, 2, \ldots, m,$$

where $D = AH^{-1}A^T, c = b + AH^{-1}g$.

If the inverse matrix H was known, then the quadratic programming problem (1.1) would be equivalent to the problem (1.3) which has simple constraints. In [4], a method is proposed for the solution of the problem (1.1), in which the inversion of matrices is avoided. The method results from joining the solution of some systems of linear equations with the same matrix H with the method of selecting the active constraints.

Below, another algorithm is proposed for the solution of the considered problem (1.1), in which the matrix inversion is also avoided. Solving $(m+1)$ quadratic programming problems without constrains, we obtain a quadratic programming problem with simple constraints equivalent to the dual problem (1.3). These methods are described in Section 2. In Section 3, an approximate method of the Newton's type for the solution of the quadratic programming problem with simple constraints is described, which is an adaptation of the version proposed by Coleman and Li in [6]. Finally, in Section 4, the extension to Semidefinite Programming will be discussed.

Now we will briefly describe the notation used in this paper. All vectors are column vectors. The superscript notation x^i referes to an element of the vector x. A subscript k is used to denote iteration numbers. Superscript T denotes transposition. $\Re^n$ denotes the space of n-dimensional real column vectors. For symmetric matrix A we use $A \succeq 0$ to denote that A is positive semidefinite, and $A \succ 0$ to denote that A is positive definite. We write $A \succeq B$ to denote that $A - B \succeq 0$. We use $Tr\,(A)$ to denote trace of square matrix A, namely the sum of the diagonal elements of A. Given two matrices A and B, we define the inner product of A and B as $\langle A, B\rangle = Tr\left(A^T B\right)$. $vec(A)$ is a column vector whose entries come from a given matrix A by stacking up columns of A.

2 Description and Motivation of the Method

We consider the quadratic programming problem having the form (1.1). The method proposed for the solving of this problem consists in the following:

Step 1 The free minimum point x_0 of the quadratic function is determined:

$$f(x) = \frac{1}{2} x^T H x + g^T x. \tag{2.1}$$

The calculation of the minimum x_0 implies the solving of the system of linear equation: $Hx = -g$; however, a method of unconstrained optimization such as that of conjugate directions can be used.

Step 2. The vector

$$d = Ax_0 = (a_1^T x_0, a_2^T x_0, \dots, a_s^T x_0)^T$$

is being determined by the point x_0, where a_i^T is the line i of the matrix A. If $d \leq b$ then $x_* = x_0$ is the optimal solution and the problem (1.1) is solved, otherwise it is passed on to the following step.

Step 3. The free minimum points $x_1, x_{2,} \dots, x_m$ of the respective quadratic functions are calculated:

$$f_i(x) = \frac{1}{2} x^T H x - a_i^T x, i = 1, 2, \dots, m. \tag{2.2}$$

This may by achieved by solving the linear equation systems: $Hx_i = a_i$, $i = 1, 2, \dots, m$ with the same matrix H, or by applying other methods of unconstrained optimization.

Step 4. Using the solution obtained above, at step 3, we construct the matrix $W = (w_{ij})$ with the dimensions $m \times m$. This matrix has the elements $w_{ij} = a_i^T x_j, 1 \leq i, j \leq m$.

Step 5. The quadratic programming problem with simple constraints is solved:

$$\min_{\lambda} \{\varphi(\lambda) = \frac{1}{2} \lambda^T W \lambda + (b - d)^T \lambda \mid \lambda \geq 0\}. \tag{2.3}$$

Step 6. The optimum solution is found:

$$x_* = x_0 - \sum_{i=1}^{m} \lambda_*^i x_i, \tag{2.4}$$

where $\lambda_* = (\lambda_*^1, \lambda_*^2, \dots, \lambda_*^m)^T$ – is the optimum in (2.1).

The validity of this method is justified by the following theorems and lemmas.

Lemma 1 *2.1 Let the matrix W is symmetric and positive semidefinite with the diagonal elements $w_{ii} > 0$, $i = 1, 2, ..., m$. If $m \leq n$ and the vectors $a_1, a_2, ..., a_m$ are linear independent then* $\det(W) \neq 0$ *(the inverse matrix W^{-1} exists) and the matrix W is positive definite.*

Proof. Therefore, we obtain

$$w_{ij} = a_i^T x_j = x_i^T H x_j = x_j^T H x_i = a_j^T H^{-1} H x_i = a_j^T x_i = w_{ji},$$

for any i, j. Hence $W^T = W$.

If we denote $X = [x_1, x_2, ..., x_m]$ - a matrix with its dimensions $m \times m$, with columns consisting the vectors $x_1, x_2, ..., x_m$, then according to what was written above we can write: $W = X^T H X$. So W is a positive semidefinite matrix with $w_{ii} = x_i^T H x_i > 0$, because the matrix H is definite positive as being.

If we have a system of linear independent vectors $\{a_i\}_{i=1}^{i=m}$ then the $rank\,(X) = m$ and, as a consequence, $rank\,(W) = m$ and $W \succ 0$. The lemma is proved. ■

Lemma 2 ***2.2*** *The quadratic programming problems (1.3) and (2.3) are equivalent.*

Proof. According to the proposed algorithm $HX = A^T$ and so $W = X^T H X = X^T A^T = AH^{-1}A^T$.

We also notice that $d = -AH^{-1}g$ (see step 1), so, as a consequence, we have $c = b + AH^{-1}g = b - d$. The lemma is proved. ■

Corollary 3 *The quadratic programming problem (2.3) has an unique optimal solution.*

We notice that x_* that has been calculated using formula (2.4) coincides with the one that was calculated using formula (1.2). This justifies the proposed algorithm. This result may be proved directly and is formulated in the theorem that follows:

Theorem 4 ***2.1*** *Let $x_1, x_2, \ldots, x_m$ be the points determined by $m+1$ successive minimizations of the quadratic functions (2.1)-(2.2) and let λ_* be the optimal solution of the quadratic programming problem with simple constraints (2.3). Then x_* determined from formula (2.4) is an optimal solution of the quadratic programming problem (1.1).*

3 Solving a Quadratic Programming Problem with Simple Bounds

In this part of the paper will be shown how a quadratic programming problem of the following form can be solved:

$$\begin{aligned} \varphi(\lambda) &= \frac{1}{2}\lambda^T W \lambda + c^T \lambda \to \min \\ \text{subject to} \quad \lambda &\geq 0 \, , \end{aligned}$$

where W is a positive semidefinite matrix of the dimension $m \times m$, and $c \in \Re^m$. A necessary and sufficient condition for λ_* to be the optimal solution of the considered problem is the existence of $\mu_* \in \Re^m$ so that the Karush - Kuhn - Tucker conditions can be satisfied:

$$\nabla \varphi(\lambda_*) - \mu_* = 0, \tag{3.1}$$

$$\lambda_*^T \mu_* = 0, \tag{3.2}$$

$$\lambda_* \geq 0, \mu_* \geq 0. \tag{3.3}$$

It follows from the above relations that if $\lambda_*^i > 0$ then $\frac{\partial \varphi(\lambda_*)}{\partial \lambda^i} = 0$, and if $\lambda_*^i = 0$ then $\frac{\partial \varphi(\lambda_*)}{\partial \lambda^i} \geq 0$. Taking these into consideration, we can say that λ_* is a solution of the following system of equations and inequalities:

$$\left. \begin{array}{r} \lambda^1 \frac{\partial \varphi(\lambda)}{\partial \lambda^1} = 0, \\ \vdots \\ \lambda^m \frac{\partial \varphi(\lambda)}{\partial \lambda^m} = 0, \\ \lambda^i \geq 0, i = 1, 2, \ldots, m. \end{array} \right\} \tag{3.4}$$

But the system (3.4) can have solutions that do not satisfy relations (3.1)-(3.3). For example, an alien solution is $\overline{\lambda}^i = 0$ and $\frac{\partial \varphi(\overline{\lambda})}{\partial \lambda^i} < 0$ for a certain $i \in \{1, 2, ..., m\}$. As follows, we are going to use a procedure that has at its basis the idea from [6], to replace system (3.4) by another suitable one, whose solving would lead to the optimal solution of the given problem.

Let $V(\lambda) = diag\left(v_1(\lambda), v_2(\lambda), ..., v_m(\lambda)\right)$ be a diagonal matrix having the the diagonal elements defined as follows:

$$v_i(\lambda) = \begin{cases} \left(\lambda^i\right)^{\frac{1}{2}}, \text{ if } \frac{\partial\varphi(\lambda)}{\partial\lambda^i} \geq 0, \\ 1, \text{ if } \frac{\partial\varphi(\lambda)}{\partial\lambda^i} < 0. \end{cases}$$

Then the system (3.4), as it was shown in [6], can be reduced to the following equivalent system:

$$\left.\begin{array}{r} V(\lambda)^2 \nabla\varphi(\lambda) = 0, \\ \lambda^i \geq 0, i = 1, 2, \ldots, m. \end{array}\right\} \tag{3.5}$$

As follows, we are going to apply formally Newton's method for solving the system of equation (3.5) . This requires Jacobian matrix:

$$\Im(\lambda) = \left[V(\lambda)^2 \nabla\varphi(\lambda)\right]' =$$
$$= V(\lambda)^2 W + diag\left(\frac{\partial v_1^2(\lambda)}{\partial\lambda^1}, \frac{\partial v_2^2(\lambda)}{\partial\lambda^2}, \cdots, \frac{\partial v_m^2(\lambda)}{\partial\lambda^m}\right) \times$$
$$\times diag\left(\frac{\partial\varphi(\lambda)}{\partial\lambda^1}, \frac{\partial\varphi(\lambda)}{\partial\lambda^2}, \cdots, \frac{\partial\varphi(\lambda)}{\partial\lambda^m}\right). \tag{3.6}$$

The vector function $V(\lambda)^2 \nabla\varphi(\lambda)$ is continuous. We can notice that if $\frac{\partial\varphi(\lambda)}{\partial\lambda^i} = 0$ for a certain $i \in \{1, 2, \ldots, m\}$ then the function $v_i^2(\lambda)$ is not differentiable and we can not determine the Jacobian matrix $\Im(\lambda)$. It is suggested in this case in [6] to write artificially $\frac{\partial v_i^2(\lambda)}{\partial\lambda^i} = 0$. For this, we consider a diagonal matrix

$$E(\lambda) = diag\left(e_1(\lambda), e_2(\lambda), \ldots, e_m(\lambda)\right)$$

in which the diagonal elements represent the product $\frac{\partial v_i^2(\lambda)}{\partial\lambda^i} \times \frac{\partial\varphi(\lambda)}{\partial\lambda^i}$:

$$e_i(\lambda) = \begin{cases} \frac{\partial\varphi(\lambda)}{\partial\lambda^i}, \text{ if } \frac{\partial\varphi(\lambda)}{\partial\lambda^i} > 0, \\ 0, \text{ otherwise,} \end{cases}$$

for $i = 1, 2, \ldots, m$. In other words, if $\frac{\partial\varphi(\lambda)}{\partial\lambda^i} = 0$ then we presume that $\frac{\partial v_i^2(\lambda)}{\partial\lambda^i} = 0$ and the Jacobian matrix takes the following form:

$$\Im(\lambda) = \left[V(\lambda)^2 \nabla\varphi(\lambda)\right]' = V(\lambda)^2 W + E(\lambda).$$

Let $\lambda_k \geq 0$ be the current approximation to the solution of problem considered. Taking into consideration all written above, a Newton's step for solving system $V(\lambda)^2 \nabla\varphi(\lambda) = 0$ is defined as follows:

$$\left[V(\lambda_k)^2 W + E(\lambda_k)\right](\lambda - \lambda_k) = -V(\lambda_k)^2 \nabla\varphi(\lambda_k). \tag{3.7}$$

We denote $\overline{\lambda}_k$ as the solution of the linear system (3.7). It is clear that the calculation of $\overline{\lambda}_k$ according to (3.7) is possible only when matrix $V(\lambda_k)^2 W + E(\lambda_k)$ is nonsingular. Matrix W, by assumption, is positive semidefinite. We obviously have

$$z^T\left(V(\lambda)^2 W + E(\lambda)\right)z = z^T V(\lambda) W V(\lambda) z + z^T E(\lambda) z \geq$$

$$\geq \sum_{i=1}^{m} e_i(\lambda)\left(z^i\right)^2 > 0$$

for any $z \in R^m$ and any $\lambda \geq 0$. This means that matrix $V(\lambda)^2 W + E(\lambda)$ is positive definite $\forall\lambda \geq 0$ and, as a consequence, is nonsingular. In this way a solution of system(3.7) is guaranteed to exist. Having this solution, a new approximation $\lambda_{k+1} = \lambda_k + \alpha_k\left(\overline{\lambda}_k - \lambda_k\right)$ is calculated, where step α_k is determined according to the following relation:

$$\alpha_k = \min\left\{1, \frac{-\lambda_k^i}{\overline{\lambda}_k^i - \lambda_k^i} \mid \overline{\lambda}_k^i - \lambda_k^i < 0\right\}.$$

See [6] for details.

4 Extension to Semidefinite Quadratic Programming Problem

Semidefinite programming has applications in diverse domains. A survey of semidefinite optimization problem and of its applications in convex constrained optimization, control theory and combinatorial optimization is given in [7]. Most of the succes is related to the links between the Lagrangian and semidefinite relaxation, as studies in [8].

The above method can be extended to the following semidefinite quadratic programming problem:

$$f\left(x\right)=\frac{1}{2}x^{T}Hx+g^{T}x\rightarrow \min$$

$$\text{subject to } \sum_{i=1}^{n}x^{i}A_{i}\succeq B, \tag{4.1}$$

where A_i, B are $m\times m$ symmetric matrices. The matrices $A_1, A_2, \ldots, A_n$ are further assumed to be linearly independent.

We define the Lagrangian for the problem (4.1):

$$L\left(x,\Lambda\right)=f\left(x\right)-\langle\Lambda,\sum_{i=1}^{n}x^{i}A_{i}-B\rangle,$$

where Λ is a $m\times m$ symmetric positive semidefinite matrix of "Lagrange multipliers".

The point x_* is an optimal solution for (5.1) if and only if there exists a nonzero symmetric $m\times m$ matrix $\Lambda_*\succeq 0$ such that the pair (x_*, Λ_*) satisfies (see, e.g., [9])

$$\left.\begin{array}{r}\nabla_{x}L\left(x_{*},\Lambda_{*}\right)=0,\\ \langle\Lambda_{*},\sum\limits_{i=1}^{n}x_{*}^{i}A_{i}-B\rangle=0,\\ \sum\limits_{i=1}^{n}x_{*}^{i}A_{i}-B\succeq 0.\end{array}\right\}$$

Let $\widetilde{A}(x) = \sum\limits_{i=1}^{n} x^i A_i - B$. Using the substitution $\Lambda_* = \widetilde{\Lambda}_*^2$, where $\widetilde{\Lambda}_*$is a symmetric positive semidefunite matrix, we can rewrite the optimality conditions as

$$Hx_* + g - \begin{pmatrix} Tr\left(\widetilde{\Lambda}_* A_1 \widetilde{\Lambda}_*\right) \\ Tr\left(\widetilde{\Lambda}_* A_2 \widetilde{\Lambda}_*\right) \\ \cdots \\ Tr\left(\widetilde{\Lambda}_* A_n \widetilde{\Lambda}_*\right) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{pmatrix}, \tag{4.2}$$

$$Tr\left(\widetilde{\Lambda}_* \widetilde{A}(x_*) \widetilde{\Lambda}_*\right) = 0, \quad \widetilde{A}(x_*) \succeq 0.$$

For simplification, we use $u\left(\widetilde{\Lambda}, A\right)$ to denote the vector

$$\left(Tr\left(\widetilde{\Lambda} A_1 \widetilde{\Lambda}\right), Tr\left(\widetilde{\Lambda} A_2 \widetilde{\Lambda}\right), \ldots, Tr\left(\widetilde{\Lambda} A_n \widetilde{\Lambda}\right)\right)^T.$$

By (4.2)

$$x_* = -H^{-1}\left(g - u\left(\widetilde{\Lambda}_*, A\right)\right).$$

The corresponding dual problem is

$$\left.\begin{array}{c} \inf_x \left(\frac{1}{2} x^T H x + g^T x - Tr\left(\widetilde{\Lambda} \widetilde{A}(x) \widetilde{\Lambda}\right)\right) \to \max \\ \text{subject to } \ \widetilde{\Lambda} = \widetilde{\Lambda}^T \end{array}\right\}. \tag{4.3}$$

Using (4.2) and remembering that

$$\langle \Lambda, \sum_{i=1}^{n} x^i A_i - B \rangle = \left[u\left(\widetilde{\Lambda}, A\right)\right]^T H^{-1} u\left(\widetilde{\Lambda}, A\right) -$$

$$-g^T H^{-1} u\left(\widetilde{\Lambda}, A\right) - \langle \Lambda, B \rangle,$$

we show that the problem (4.3) can be represented in the following way:

$$-\frac{1}{2}\left[u\left(\widetilde{\Lambda},A\right)\right]^T H^{-1}u\left(\widetilde{\Lambda},A\right)-g^T H^{-1}u\left(\widetilde{\Lambda},A\right)+$$
$$+\langle\Lambda,B\rangle-g^T H^{-1}g\to\max$$

$$\text{subject to } \Lambda\succeq 0,\quad \widetilde{\Lambda}=\widetilde{\Lambda}^T.$$

By M we denote the matrix of the $n\times n^2$ dimension:

$$M=\begin{pmatrix}[vec\,(A_1)]^T\\ [vec\,(A_2)]^T\\ \vdots\\ [vec\,(A_n)]^T\end{pmatrix}.$$

We obtain $u\left(\widetilde{\Lambda},A\right)=Mvec\,(\Lambda)$. Thus we have reformulated the dual problem (4.3) to the following problem:

$$-\frac{1}{2}\left[vec\,(\Lambda)\right]^T M^T H^{-1}Mvec\,(\Lambda)+$$
$$+\left[vec\,(B)+M^T H^{-1}\right]^T vec\,(\Lambda)-g^T H^{-1}g\to\max$$
$$\text{subject to } \Lambda\succeq 0,\Lambda=\Lambda^T.$$

References

[1] Boggs P.T., Tolle J.W. *Sequential quadratic programming.* Acta Numerica, 4, pp. 1-50, 1995.

[2] Pshenichnyi B.N., Danilin Yu.M. *Numerical Methods in Extremal Problems*, Mir Publishers, Moscow, 1977.

[3] Pshenichnyi B.N. *Metod linearizatii*, Nauka, Moscow, 1983 (in Russian).

[4] Moraru V. *An Algorithm for Solving Quadratic Programming Problems.* Computer Science Journal of Moldova, vol.5, No. 2, 1997, pp.223-235.

[5] Bazaraa M.C.,Shetty C.M. *Nonlinear Programming. Theory and Algorithms.* John Wiley & Sons, Inc., 1979.

[6] Coleman T.F., Li Y. *On the Convergence of Interior-Reflective Newton Methods for Nonlinear Minimization Subject to Bounds.* Math. Programming, Vol. 67, 1994, pp. 189-224.

[7] Vandenberge L., Boyd S. *Semidefinite Programming.* SIAM Review, Vol. 38, 1996, pp.49-95.

[8] Schor N.Z. *Quadratic Optimization Problems.* Tekhnicheskaya Kibernetika, No, 1, 1987, pp. 128-139.

[9] Nesterov Yu., Nemirovskii A. *Interior Point Polynomial Algorithm in Convex Programming: Theory and Applications.* SIAM, Philadelphia, 1994.

V.Moraru,
Department of Computing Mathematics and Programming
Technical University of Moldova
Bd. Stefan cel Mare No.168,
2012 Cishinau, Moldova,
e-mail: *moraru@mail.utm.md*

Received July 30, 2000