Computer Science Journal of Moldova, vol.**5**, no.2(14), 1997

An Algorithm for Solving Quadratic Programming Problems

V. Moraru

Abstract

Herein is investigated the method of solution of quadratic programming problems. The algorithm is based on the effective selection of constraints. Quadratic programming with constraints-equalities are solved with the help of an algorithm, so that matrix inversion is avoided, because of the more convenient organization of the Calculus. Optimal solution is determined in a finite number of iterations. It is discussed the extension of the algorithm over solving quadratic non-convex programming problems.

Key words: Quadratic Programming, Optimization, Active Constraints.

1 Introduction

In this paper an algorithm will be described for solution of quadratic programming problems of the form:

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

$$\text{subject to } Ax \geq b, \tag{2}$$

where H is a symmetric matrix, nonsingular of the $n \times n$ dimension. A is a $m \times n$ matrix, g, x and b are column vectors, g and $x \in R^n$ and $x \in R^n$. The symbol T indicates transposition operator.

For the last years quadratic programming problems have been of a great interest and are utilized intensively to solve problems of constrained optimization. In the majority of Newton or quasi-Newton methods of solution of nonlinear programming problems with constraints it is necessary to solve problems of the type (1)–(2) at every step, where g and H are respectively the gradient and Hesse matrix of the objective function[1–5] or of Lagrange function[3–5].

Let it be x_* an optimal solution of the problem (1)–(2). Then there exists Lagrange multipliers vector

$$\lambda_* = (\lambda_*^1, \lambda_*^2, \ldots, \lambda_*^m)^T,$$

so that Kuhn-Tucker conditions are satisfied:

$$\left.\begin{aligned} Hx_* + g = A^T\lambda_*, \\ (Ax_* - b)^T\lambda_* = 0, \\ Ax_* \geq b, \\ \lambda_* \geq 0. \end{aligned}\right\} \tag{3}$$

If the matrix H is positively semi-definite then Kuhn-Tucker conditions (3) are necessary and sufficient to x_* be a global minimum point.

Let it be the set of indexes

$$I(x) = \{i | a_i^T x = b^i\},$$

where a_i^T is row i of matrix A.

$I(x)$ gives us the set of active constraints in a point $x \in R^n$.

If the matrix H is indefinite, then conditions (3) are added following:

$$\lambda_*^i > 0, \forall i \in I(x_*)$$

and

$$p^T Hp > 0, \forall p \in \{p \in R^n | p \neq 0, a_i^T p = 0, i \in I(x_*)\} \tag{4}$$

this means that H is positively definite on a linear variety defined by active constraints.

A great number of methods is designed to solve quadratic programming problems in a finite number of steps. One of the most popular schemes of solution of quadratic programming problems is based on direct solution of a system of inequations and equations (3). Today there exist many methods to solve Kuhn-Tucker system. Beale [6], Frank and Wolfe [7], Wolfe [8], Shetty [9], Lemke [10], Cottle and Dantzig [11] and others have generalized and modified simplex method from linear programming to solve Kuhn-Tucker system (3).

There is another way to solve quadratic programming problems based on the idea of effective selection of constraints and the solution at every step quadratic programming problems with constraint equalities. Pshenichniy and Danilin [1], Pshenichniy [2], Best and Ritter [12] use conjugate direction method to solve sub-problems that appear during the work process. Theil and van de Panne [13], Boot [14], Fletcher [15], Gill and Murray [3,4] reduce the general quadratic programming problem to the solution of a number of constraint equalities problems. It is worth mentioning Gill and Murray works (see other references [3,4]) where they use largely so-called null subspace method.

In this work an algorithm is presented to solve quadratic programming problems where at every step a sub-problem with effective constraints is solved. In the second part a convenient method to solve quadratic programming problems with constraint equalities. This method reduces the determination of Kuhn-Tucker system solution to solving $s+1$ systems of linear equations with the same $H$of $n\times n$ and of a system of s linear equations with s unknown variables, where s is the number of constraint equalities and $s \leq n$. In the third part an algorithm of solving problems of the form (1)–(2) that is based on the idea of effective constraints selection will be discussed. Tte extension of this algorithm to non-convex quadratic programming is effected in the fourth part herein.

2 Minimization of quadratic functions on a linear variety

Let us minimize quadratic function (1) subject to

$$a_i^T = b^i, i = 1, 2, \ldots, s. \tag{5}$$

Now we will assume that matrix H is positively definite, i.e. function $f(x)$ is strictly convex. We will also assume that $s \leq n$ and that vectors $a_1, a_2, \dots a_s$ are linearly independent.

Due to Kuhn-Tucker conditions this problem is reduced to finding $x \in R^n$ and $\lambda \in R^s$ that would satisfy the system of linear equations:

$$\left.\begin{aligned} Hx - A^T\lambda = -g, \\ Ax = b, \end{aligned}\right\} \tag{6}$$

where A is a matrix of $s \times n$ dimensions with rows $a_i^T = 1, 2, \dots, s$ and $b = (b^1, b^2, \dots, b^s)^T$.

With $rang(A) = s$, system (6) of $n+s$ equations and $n+s$ unknown variables has the only solution (x_*, λ_*) that is the stationary point for Lagrange function associated to the problem (1), (5):

$$L(x, \lambda) = f(x) - \lambda^T(Ax - b).$$

In the case when $n + s$ is very large we will have a system of very large dimension which is undesirable and avoided in practice. The total number of necessary arithmetic operations to find the solution of the system (6) is approximately equal to $\frac{2(n+s)^3}{3}$.

Computation of x_* and λ_* could be carried out separately. For this matter (see [3–5]) the inverse matrix H^{-1} and matrix product $AH^{-1}A^T$ are calculated. Then Lagrange multipliers vector λ_* is given by the system:

$$(AH^{-1}A^T)\lambda = AH^{-1}g + b \tag{7}$$

and the optimal solution x_* is defined by

$$x_* = H^{-1}(A^T\lambda_* - g). \tag{8}$$

The inversion of the matrix H is equivalent the solution of n systems of liner equations and needs $\approx \frac{4n^3}{3}$ arithmetic operations. To find the optimal solution x_* of the problem(1),(5) according to the relations (7) and (8) implies in total $\approx \frac{2}{3}(5n^3 + s^3)$.

Determination of Lagrange multipliers vector λ_* and of the optimal solution x_* needs the determination of the matrix H^{-1} with the help of which the matrix $AH^{-1}A^T$ (or matrix AH^{-1}) is built for subsequent solution of the system of linear equations (7). We know that matrix inversion is a costly operation. It needs approximately two times more computing memory than the solution of a system of linear equations. Besides the calculation of an inverse matrix is made with approximations that are fatal (see, for example [16,17]).

As we could see, this method generally is not efficient. Now we are going to describe a method of minimization of the quadratic function (1) on a linear variety (5). This algorithm is also based on a solution the system of linear equations (6) and in which the matrix inversion is avoided because of a specific organization of the calculations. This method leads to the solution of $s+1$ systems of linear equations with the only matrix H, of a system of s equations and is very efficient when $s << n$. The advantage of this system is great when solving quadratic programming problems with constraints inequalities (1)–(2). Embedding from a linear variety to another one here imposes no more than solving of two systems of linear equations: one of n and other of s equations.

The elaborated algorithm to solve the system of linear equations (6) (i.e. of the problem (1)–(5) was presented in [18,19] and consists of the following operations:

Step 1. The system of linear equations is solved

$$Hy_0 = -g, \tag{9}$$

so that the point of free-minimum of the quadratic function $f(x)$ is determined. If $Ay_0 = b$ then $x_* = y_0$ is an optimal solution and the problem (1)–(5) is solved. Otherwise the next steps are followed.

Step 2. The vectors $y_1, y_2, \ldots, y_s$ are determined, solving s systems of linear equations with the only matrix H:

$$Hy_i = a_i, i = 1, 2, \ldots, s. \tag{10}$$

Step 3. Matrix $V = (v_{ij}$ is created, of $s \times s$ dimensions and with elements $v_{ij} = y_i^T a_j, 1 \leq i, j \leq s$.

Step 4. The following vector is formed

$$d = Ay_0 = (a_1^T y_0, a_2^T y_0, \ldots, a_s^T y_0)^T$$

with s components $d^i = a_i^T y_0, i = 1, 2, \ldots, s$.

Step 5. The solution λ_* of the linear equations system is determined

$$V\lambda = b - d. \tag{11}$$

Step 6. The optimal solution is calculated

$$x_* = y_0 + Y\lambda_* = y_0 + \sum_{j=1}^{s} \lambda_*^j y_j,$$

where $Y = (y_1, y_2, \ldots, y_s)$ is a matrix of $n \times s$ dimensions, its columns are vectors $y_j, j = 1, 2, \ldots, s$.

Let's now analyze the presented algorithm. Matrix V is symmetric and positively definite. It is true that:

$$y_i^T a_j = y_i^T H y_j = y_j^T H y_i = y_j^T a_i, \forall i, j,$$

so, V is modified Gram matrix in which the scalar product is determined by the positively defined matrix H. It is easily seen that $V = Y^T HY$ and $rang(Y) = s$ where from $rang(V) = s$. Thus the system (11) has the only solution λ_*.

It is immediately verified that

$$a_i^T x_* = a_i^T y_0 + \sum_{j=1}^{s} \lambda_*^j a_i^T y_j = d^i + b^i - d^i = b^i, i = 1, 2, \ldots, s,$$

and this means that x_* is the only feasible solution.

Let it be x any feasible solution, i.e. $a_i^T x = b_i, i = 1, 2, \ldots, s$. As a result:

$$[\nabla f(x_*)]^T (x - x_*) = [Hx_* + g]^T (x - x_*) = [HY\lambda_*]^T (x - x_*) =$$

$$= \left[\sum_{j=1} s\lambda_*^j H y_j \right]^T (x - x_*) = \sum_{j=1} s\lambda_*^j a_j^T (x - x_*) = 0.$$

So, x_* is the optimal solution of the problem (1),(5).

Remarks:

1. Solving of linear equations systems (9)–(10), with the only matrix H, one needs $\frac{n^3}{3} + sn^2$ arithmetic operations. Matrices H and V are positively definite. As a result, systems of equations (9)–(11), could be solved with stable numerical algorithms [16, 17], Cholesky method for instance. In this method matrices H and V are presented in the form $H = LL^T$ and $V = RR^T$. Cholesky factors L and R are inferior triangular matrices of $n \times n$ and respectively $s \times s$ dimensions and are calculated only once at the beginning and could be stored instead of matrix H.

2. Systems (7) and (11), are equivalent, because $V = AH^{-1}A^T$ and $d = -AH^{-1}g$. It has to be remarked that the inverse matrix of H is not calculated. We denote that $d = -Y^T g$ is a formula to be used in the fourth part of the work.

3 The case of problems with constraint inequalities

Now we will return to the problem (1)–(2), where f is a strictly convex function, i.e. H is a positively definite matrix. Constraint inequalities create new difficulties because it's unknown beforehand which of the problem constraints are verified as equalities by the optimal solution x_*. If the set of indexes

$$I(x_*) = \{i | a_i^T x_* = b^i\}$$

was known we would determine the optimal solution minimizing the quadratic function (1) on the linear variety

$$a_i^T x = b^i, i \in I(x_*).$$

This linear variety could be looked for in a systematic mode with the help of effective constraint selection method with the following main idea found in [1–4].

Let it be x_k feasible solution of the problem (1)–(2),. The steps of an iteration are:

1. Determination of $I(x_k)$ — the set of indexes of those constraints that are verified as equalities by the point x_k.

2. Calculation of p_k — Newtonian direction that origins from x_k and Lagrange multipliers λ_k, solving the quadratic programming problem of the form:

$$\left.\begin{array}{c} f(x_k+p)=\frac{1}{2}p^TH_p+(Hx_k+g)^T\,p+f(x_k)\rightarrow\ min \\ \text{subject to}\quad a_i^Tp=0, i\in I(x_k). \end{array}\right\} \tag{12}$$

3. Calculation of a new feasible solution

$$x_{k+1}=x_k+\alpha_kp_k,$$

   where α_k is a step length on the direction p_k and is always chosen so that x_{k+1} point is feasible, $0<\alpha_k\leq 1$.

4. Increment of $k=k+1$ and returning to the step 1.

5. After a finite number of steps the set of active constraint indexes remains unchanged and $p_k=0$. If Lagrange multipliers $\lambda_k^i\geq 0, i\in I(x_k)$, then x_k is an optimal solution of the problem (1)–(2). If there exits j such that $\lambda_k^j<0$ is determined a new set of indexes $I(x_k)=I(x_k)\backslash\{j\}$ and the process is repeated from the beginning.

6. In this order arbitrary point x_0 is taken from the domain of feasible solutions and linear variety of x_0 is determined, i.e. $I(x_0)$ is determined. One of the methods to calculate x_0 is to use phase 1 of the simplex algorithm (see [3–4]).

After we find the start point x_0 the quadratic programming problem is solved (12), that is reduced to solution of systems of linear equations:

$$\left.\begin{array}{r} Hy_0=-g_0,\\ Hy_i=a_i, i\in I(x_0),\\ V\lambda_0=-d, \end{array}\right\} \tag{13}$$

where $g_0 = Hx_0 + g, d^i = a_i^T y_0$, and $v_{ij} = y_i^T a_j = a_i^T y_j, i, j \in I(x_0)$.

Direction p_0 is given by the formula:

$$p_0 = y_0 + \sum_{j \in I(x_0)} \lambda_o^j y_j. \tag{14}$$

Then $x_1 = x_0 + \alpha_0 p_0$, where α_0 is determined that the new point x_1 would satisfy constraints (2). If $a_i^T p_0 \geq 0$ for any $i \notin I(x_0)$, then we will have $\alpha_0 = 1$. For $i \notin I(x_0)$ and $a_i^T p_0 < 0$ the approximation x_1 remains an feasible solution, if

$$\alpha_0 = \min \left\{ \frac{b^i - a_i^T x_0}{a_i^T p_0} | i \notin I(x_0), a_i^T p_0 < 0 \right\}.$$

If there exists i such that $a_i^T p < 0$ at the point x_1 one or more constraints become active. When this happens the last constraints are included in the set $I(x_1)$ and the determination of a new direction p_1 begins. In order to determine p_1 we start from (13), where g_0 is substituted with $g_1 = Hx_1 + g$ and the systems of linear equations

$$\left. \begin{aligned} H\overline{y}_0 &= -g_1, \\ Hy_s &= a_s, \end{aligned} \right\}$$

are solved again; here $s \notin I(x_0)$ and the constraint s is active at the point x_1. This is easily performed if we have, for instance, Cholesky factorization $H = LL^T$.

It is worth mentioning that it's unnecessary to solve directly the system of equations $H\overline{y}_0 = -g_1$, because the solution of this system can be obtained from:

$$\overline{y}_0 = y_0 - \alpha_0 p_0. \tag{15}$$

It is true that $H(\overline{y}_0 - y_0) = H(x_0 - x_1)$ is an equality that allows us to obtain (15).

We get the lagrange multiplier vector λ_1 solving the system of equations $\overline{V}\lambda_1 = -\overline{d}$, where $\overline{d} = (d^T, a_s^T \overline{y}_0)^T$ and new matrix $\overline{V}$ is obtained with the aid of matrix V with addition of new rows and columns:

$$\overline{V} = \begin{pmatrix} V & u \\ u^T & v_{ss} \end{pmatrix}.$$

Here vector u has the components $u^i = a_s^T y_i, \in I(x_0)$ and $v_{ss} = a_s^T y_s$. If Cholesky factorization $V = RR^T$ is known, then the matrix $\overline{R}$ from $\overline{V} = \overline{RR}^T$ decomposition is given

$$\overline{R} = \begin{pmatrix} R & 0 \\ w & r_{ss} \end{pmatrix}.$$

It is immediately verified that vector w is the solution of the system of linear equations $Rw = u$, and $r_{ss} = \sqrt{v_{ss} - w^T w}$. Matrix V is positively definite. Due to this fact $v_{ss} > w^T w$, i.e. we can calculate r_{ss}.

If $p_k = 0$, the condition is satisfied

$$\nabla f(x_k) = A^T \lambda_k \tag{16}$$

and we are at the minimum point on the linear variety obtained from intersection of active constraints associated with x_k. If all $\lambda_k^i \geq 0$, the point x_k represents the optimal solution of the problem (1),(5), because Kuhn-Tucker conditions are satisfied. When at least one of $\lambda_k^i < 0$, the constraint that gives us the negative component is declined, making instead of the matrix V in (13) a matrix $\overline{V}$ obtained from V eliminating j row and column.

We emphasize that if in (16) we have two or more negative components of the vector λ_k, the corresponding constraints will be eliminated from the set of active constraints in turn, simultaneous elimination of two constraints could lead us an inadmissible direction p_{k+1}.

Active constraint selection strategy guarantees that the embedding from a linear variety to another one will decrease the value of objective function. As the number of constraints is finite it results that a finite number of steps will lead us to finding the optimal solution of the problem in consideration.

4 Minimization of indefinite quadratic functions

Let us consider now the problem (1)–(2) where H is any quadratic matrix, symmetric and indefinite. We will suppose that conditions (3)–(4) are satisfied, what guarantees the identity and the existence of the minimum for the considerate problem.

In order to apply the above presented algorithm it's necessary that the matrix V is positive definite. As we see $V = AH^{-1}A^T = Y^T HY$, but in general case matrixes H and V are not positively definite nor in the solution neighborhood. In reality the optimal solution of the problem (12) doesn't change if we substitute the matrix H with another one of the form (see [3–5, 20]):

$$\hat{H} = H + \sum_{i \in I(x_k)} \sigma_i a_i a_i^T, \sigma_i \geq 0,$$

where σ_i are arbitrary non-negative real numbers.

If $I(x_k) = I(x_*)$ then at the base of the relations (4) are numbers $\sigma_i \geq 0$ so that the matrix $\hat{H}$ is positive definite even if H is singular or indefinite. Determination of set $I(x_*)$ of constraints as egalities of optimal solution x_*, is generally a difficult problem. We could apply linearization method[2,5] starting with matrix $\hat{H} = I$ to determine $I(x_*)$. We are going to demonstrate its practical application in our next paper.

References

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

[2] Pshenichnyi B.N. Metod linearizatii. Nauka, Moscow, 1983 (Russian).

[3] Numerical Methods for Constrained Optimization. Edited by P.E.Gill and W.Murrey. Academic Press, 1974.

[4] Gill P.E., Murray W., Wright M. Practical Optimization. Academic Press, 1981.

[5] Bertsekas D. Constrained Optimization and Lagrange Multiplier Methods. Academic Press, 1982.

[6] Beale E., M., L. On Optimizing a Convex Function Subject to Linear Inequalities. Journal of Royal Statistical Society, No.17, 1955.

[7] Frank M., Wolfe P. An Algorithm for Quadratic Programming. Naval Research Logistics Quarterly, No.3, 1956, p.95–110.

[8] Wolfe P. The Simplex Method for Quadratic Programming. Econometrica, No.27, 1959, p.382–398.

[9] Shetty C.M. A simplified Procedure for Quadratic Programming. Operations Research, No.11, 1963, p.248–260.

[10] Lemke C.E. Bimatrix Equilibrium Points and Mathematical Programming. Management Science, No.11, 1965, p.681–689.

[11] Cottle R.W., Dantzig G.B. Complementary Pivot Theory of Mathematical Programming. Linear Algebra and Applications, No.1, 1968, p.103–125.

[12] Best M.J., Ritter K. A Quadratic Programming Algorithm. Zeitschrift fur Operations Research, vol.32, 1988, p.271–297.

[13] Theil H., van de Panne C. Quadratic Programming as an Extension of Conventional Quadratic Maximization. Management Science, No.7, 1961, p.1–20.

[14] Boot J.C.G. Notes on Quadratic Programming: The Kuhn-Tucker and Theil-van de Panne Conditions, Degeneracy and Equality Constraints. Management Science, No.8, 1961, p.85–98.

[15] Flecher R.A General Quadratic Programming Algorithm. J. Inst. Maths. Applics, No.7, 1971, p.76–91.

[16] Bucur C.M., Popeea C.A., Simion Gh.Gh. Matematici speciale. Calcul numeric. Ed. Didactică şi Pedagogică, Bucureşti, 1983.

[17] Rice J.R. Matrix Computations and Mathematical Software. McGraw-Hill booc Company, New York, 1981.

[18] Moraru V. Asupra unei metode in programarea pătratică. Conferinţa tehnico-ştiinţifică. U.T.M., 1994, p.10–11.

[19] Moraru V. On the Construction of Optimization methods of quasi-Newton type. Proceedings of the 5 th International Conference on Optimization of Electric and Electronic Equipment. The Transilvania University of Braşov. Braşov, May 15–17, 1996, p.23–28.

[20] Moraru V. Quasi-Newton Methods for Solving Nonlinear Programming Problems. Computer Science Journal of Moldova, vol.3, No.3, 1995, p.263–277.

V.Moraru,
Department of Computing Mathematics
and Programming,
Technical University of Moldova,
168, Bd. Stefan cel Mare,
2012 Kishinev, Moldova

Received June 2, 1997