Journal of Computational Mathematics, Vol.17, No.3, 1999, 307–314.

Subspace Search Method for Quadratic Programming with Box Constraints[*1)]

Zi-luan Wei
(*ICMSEC, Chinese Academy of Sciences, Beijing 100080, China*)

Abstract

A subspace search method for solving quadratic programming with box constraints is presented in this paper. The original problem is divided into many independent subproblem at an initial point, and a search direction is obtained by solving each of the subproblem, as well as a new iterative point is determined such that the value of objective function is decreasing. The convergence of the algorithm is proved under certain assumptions, and the numerical results are also given.

Key words: Subspace search method, Quadratic programing, Matrix splitting

1. Introduction

In this paper, we consider the problem of minimizing a quadratic convex programming with box constrained variables:

$$\begin{aligned} &\text{Min} f(x) \\ &\text{s.t.} x \in \Omega \end{aligned} \tag{1.1}$$

where $\Omega = \{\ x \in R^n\colon\ l \le x \le u\}$, $f(x) = \dfrac{1}{2}x^T Hx + b^T x$, and H is an n by n symmetric positive definite matrix, and $b,\ l,\ u$ are given constant vectors in R^n.

This problem arises in several areas of applications, such as optimal control and disign engineering, linear least square problem with bounded variables and implementation of robust method for nonlinear programming, etc. Many successful algorithms for solving this type of large scale problem have been studied based on active set strategies. A popular approach is to use an active-set algorithm that solves a sequence of subproblems of the form

$$\text{Min } f(x+d) \quad \text{s.t. } d_i = 0, \quad i \in V_k \tag{1.2}$$

where V_k is the index set of active constraints, indicating the set of varaibles that would remain fixed at one of their bounds. Obviously, it is necessary to identify a candidate active set, and to solve the problem (1.2) exactly in the active set algorithm. Especially,

[*] Received October 4, 1996.

[1)] This research supported partially by The National Natural Science Foundation of China (No.19771079) and State Key Laboratory of Scientific and Engineering Computing.

obtaining the exact minimizer of (1.2) may require many conjugate gradient iterations, and adding constaints at a time to the working set may lead to an excessive number of iterations for large scale problem. In order to avoid the above disadvantages, a different type of algorithm, based on the gradient projection, and combination of the gradient projection with conjugate gradient, have been proposed by several authors. These algorithms have finite convergence if the problem is strictly convex and the solution is nondegenerate[12]. A similar algorithm combines conjugate gradient with gradient projection technique, and uses a new strategy for the decision of leaving the current face and make it possible to obtain finite convergence even for a singular Hessian and in the presence of dual degeneracy[7]. A primal-dual interior point algorithm is also used to solve large problem (1.1), and the numerical experiments have shown that the algorithm requires only a few steps and is very efficient[9].

In this paper, we present a subspace search method for solving the problem (1.1). The main steps of the algorithm are to divide the problem (1.1) into independent subproblems at an initial feasible point and solve each of these subproblems to obtain a search direction, and then to determine a new feasible iterative point such that the objective function is decreasing. The convergence of the algorithm is proved under certain assumptions. The main feature of the algorithm is that large scale problom (1.1) can be transformed into many small independent subproblems, and all the subproblems can be solved simultaneously.

This paper is organized as follows. In Section 2 we describe the algorithm. The convergence results are proved under certain assumptions and numerical results are also given in Section 3.

2. Derivation of the Algorithm

Now we consider the problem (1.1). Without loss of generality, assume that vector $x \in R^n$ can be divided into $(x_1^T, x_2^T, \cdots, x_t^T)$, and $x_i \in R^{n_i}$, and that $n_1 = n_2 = \cdots = n_t$ and $tn_i = n$. Accordingly, matrix H and vectors b, l, u can be also subdivided into $t \times t$ block submatrices $H_{ij} (H_{ij} \in R^{n_i \times n_i}, i, j = 1, 2, \cdots, t)$ and subvectors b_i, l_i, u_i $(b_i, l_i, u_i \in R^{n_i}$, $i = 1, 2, \cdots, t)$, respectively. Therefore, the objective function $f(x)$ can be rewritten as follows.

$$f(x) = \frac{1}{2} \sum_{i=1}^{t} \sum_{j=1}^{t} x_i^T H_{ij} x_j + \sum_{i=1}^{t} b_i^T x_i \tag{2.1}$$

Assume that an initial vector $\bar{x} \in \Omega$ is a strictly interior point, that is, $l < \bar{x} < u$, and that x belongs to the neighborhood of $\bar{x}$, then we have

$$x = \bar{x} + (x - \bar{x}) \tag{2.2}$$

Substituting (2.2) into (2.1), it is easy to derive that

$$f(x) = \frac{1}{2} \sum_{i=1}^{t} \bar{x}_i^T \hat{b}_i + \sum_{i=1}^{t} (x_i - \bar{x}_i)^T \bar{b}_i + \frac{1}{2} \sum_{i=1}^{t} \sum_{j=1}^{t} (x_i - \bar{x}_i)^T H_{ij} (x_j - \bar{x}_j) \tag{2.3}$$

where $\hat{b}_i = \Big(\sum_{j=1}^{t} H_{ij}\bar{x}_j + 2b_i\Big)$, $\bar{b}_i = \Big(\sum_{j=1}^{t} H_{ij}\bar{x}_j + b_i\Big)$. Clearly, $f(x)$ is non-separable in the expression (2.3), so the problem (1.1) can not be split into t independent subproblems. One possibility to overcome this difficulty is to define the following functions

$$\varphi_i(x_i, \bar{x}) = \frac{1}{2}\bar{x}_i^T \hat{b}_i + \frac{1}{2}(x_i - \bar{x}_i)^T H_{ii}(x_i - \bar{x}_i) + (x_i - \bar{x}_i)^T \bar{b}_i \tag{2.4}$$

and

$$\varphi(x, \bar{x}) = \sum_{i=1}^{t} \varphi_i(x_i, \bar{x}) \tag{2.5}$$

It follows from (2.5) that $\varphi(x, \bar{x})$ is a quadratic function defined on $X_1 \times X_2 \times \cdots X_t$ (here $X_i = \{x_i \in R^{n_i}\colon\ l_i \le x_i \le u_i\}$, $i = 1, 2, \cdots, t$), and it is easy to verify that

$$f(\bar{x}) = \varphi(\bar{x}, \bar{x}) \tag{2.6}$$

$$\nabla f(\bar{x}) = \nabla \varphi(\bar{x}, \bar{x}) \tag{2.7}$$

and

$$f(x) - \varphi(x, \bar{x}) = \frac{1}{2}\sum_{i=1}^{t}\sum_{j\neq i}(x_i - \bar{x}_i)H_{ij}(x_j - \bar{x}_j) \tag{2.8}$$

If $\varphi(x, \bar{x})$ is used to replace $f(x)$ in (1.1), thus, the original problem (1.1) can be replaced locally by the following problem

$$\min\ \varphi(x, \bar{x}) \quad \text{s.t.} \quad x \in \Omega \tag{2.9}$$

It is obvious that $\varphi(x, \bar{x})$ is separable function, hence, minimizing (2.9) is equivalent to solve the following t independent subproblems

$$\min\ \varphi_i(x_i, \bar{x}) \quad \text{s.t.} \quad x_i \in X_i \tag{2.10}$$

Suppose that $\hat{x}$ is an optimal solution to the problem (2.10), it is easy to see that the value of objective function $f(x)$ is decreasing at $\bar{x}$ along the direction $d = \hat{x} - \bar{x}$. If matrix H can be divided into H_1 and H_2 ($H = H_1 + H_2$) such that

$$y^T(H_1 - H_2)y > 0 \tag{2.11}$$

holds for any $y (y \neq 0) \in R^n$, where H_1 is a block diagonal matrix of H consisting of t diagonal blockes of H (from H_{11} to H_{tt}), and $H_2 = H - H_1$. Obviosly, if H is a symmetric nonsingular M-matrix[3] or block-diagonal dominant matrix[2], then it is easy to verify that H can be split into H_1 and H_2 such that (2.11) holds for any $y (y \neq 0) \in R^n$. As a result, a new iterative point $x = \hat{x}$ can be generated such that $f(\hat{x}) < f(\bar{x})$. Based on the above description we are able to construct the following subspace search algorithm for solving the problem (1.1).

Algorithm A:

Let $x^0 \in \Omega$ be a given initial point, $\varepsilon > 0$ be some prescribed accuracy. And $k := 0$, then x^{k+1} is obtained by the following steps.

(i) Let $\bar{x} = x^k$, and solve the problem (2.10) for $i = 1, 2, \cdots, t$, and obtain an optimal solution $\hat{x}^k$.

(ii) If $\|\hat{x}^k - x^k\|_2 \leq \varepsilon$, then stop, and $\hat{x}^k$ is an approximate solution for the problem (1.1), Otherwise, let $x^{k+1} = \hat{x}^k$ and $k := k + 1$, and return to (i).

It follows from the definition of algorithm A that the main computational work is from solving problem (2.10) for $i = 1, 2, \cdots, t$ at each iteration. There are some different algorithms, which can be used to solve the problem (2.10).

3. Convergence Results

This section deals with the convergence of the algorithm A. We prove that sequence x^k generated by the algorithm A converges to an optimal solution x^* of problem (1.1) under certain assumptions. Several lemmas are introduced in order to prove the convergent conclusion.

Lemma 3.1. *Suppose that H is an n by n symmetric positive definite matrix, and b, l, u are given constant vectors in R^n, then x^* is an optimal solution of problem* (1.1) *if and only if $x^* \in \Omega$ and there is $y^* \geq 0$, $z^* \geq 0$ such that*

$$Hx^* + b = z^* - y^*, \quad (u - x^*)^T y^* = 0, \quad (x^* - l)^T z^* = 0. \tag{3.1}$$

hold.

Lemma 3.2. *Suppose that $\varphi(x, \bar{x})$ is defined by* (2.5), *and that x^* is an minimizer of $\varphi(x, \bar{x})$ on $X_1 \times X_2 \times \cdots \times X_t$, then*

$$\varphi(x^*, \bar{x}) \leq \varphi(\bar{x}, \bar{x}) - \frac{1}{2} \sum_{i=1}^{t} d_i^T H_{ii} d_i \tag{3.2}$$

where $d_i = x_i^* - \bar{x}_i, i = 1, 2, \cdots, t$.

Proof. It follows from the assumption that $\varphi(x, \bar{x})$ is a separable function, so x^* can be obtain from (2.10) for $i = 1, 2, \cdots, t$. A necessary and sufficient condition for x_i^* being a solution of (2.10) is

$$H_{ii}(x_i^* - \bar{x}_i) + \bar{b}_i = q^i, \quad i = 1, 2, \cdots, t. \tag{3.3}$$

where

$$q_r^i \begin{cases} > 0, & (x_i^*)_r = (l_i)_r \\ < 0, & (x_i^*)_r = (u_i)_r \\ 0, & (l_i)_r < (x_i^*)_r < (u_i)_r \end{cases} \tag{3.4}$$

and $(\ )_r$ denotes the r-th component of a vector. From (3.3) and (3.4), it is easy to verify that

$$d_i^T H_{ii} d_i + d_i^T \bar{b}_i < 0, \quad i = 1, 2, \cdots, t. \tag{3.5}$$

On the other hand, $\varphi(x, \bar{x})$ is a quadratic function, so one can easily derive that

$$\varphi(x^*, \bar{x}) = \varphi(\bar{x}, \bar{x}) + d^T \nabla\varphi(\bar{x}, \bar{x}) + \frac{1}{2} d^T \nabla^2 \varphi(\bar{x}, \bar{x}) d$$

$$=\varphi(\bar{x},\bar{x})+\sum_{i=1}^{t}d_i^T[H_{ii}d_i+\bar{b}_i]-\frac{1}{2}\sum_{i=1}^{t}d_i^TH_{ii}d_i \tag{3.6}$$

By (3.5) and (3.6), it is straightforward to obtain

$$\varphi(x^*,\bar{x})\le\varphi(\bar{x},\bar{x})-\frac{1}{2}\sum_{i=1}^{t}d_i^TH_{ii}d_i$$

which proves the conclusion of the lemma.

Lemma 3.3. *Suppose that $\bar{x}\in\Omega$ and H is divided H_1 and H_2 such that* (2.11) *satisfies, and that $\hat{x}$ is generated by the algorithm A with the starting point $\bar{x}$, then*

$$f(\hat{x})<f(\bar{x}) \tag{3.7}$$

holds.

Proof. It follows from the assumptions that $\hat{x}$ is obtained from (i) of Algorithm A. By (2.8), it is clear that

$$|f(\hat{x})-\varphi(\hat{x},\bar{x})|=\frac{1}{2}\Big|\sum_{i=1}^{t}\sum_{j\ne i}d_i^TH_{ij}d_j\Big|$$

Hence

$$f(\hat{x})\le\varphi(\hat{x},\bar{x})+\frac{1}{2}\Big|\sum_{i=1}^{t}\sum_{j\ne i}d_i^TH_{ij}d_j\Big| \tag{3.8}$$

From (3.2), (3.8) and (2.11), it is straightforward to show that

$$f(\hat{x})\le\varphi(\bar{x},\bar{x})-\frac{1}{2}(d^TH_1d-d^TH_2d)=f(x)-\frac{1}{2}d^T(H_1-H_2)d \tag{3.9}$$

This implies that (3.7) holds, which proves the lemma.

Theorem 3.4. *Suppose that H is a nonsingular M-matrix or block-diagonal dominant matrix and that H is divided into H_1 and H_2 such that $H=H_1+H_2$ and $x^0\in\Omega$ is an initial point. Assume that sequence $\hat{x}^k$ and x^k are generated by the algorithm A, then there exists an integer $k_0>0$ such that $\hat{x}^{k_0}=x^{k_0}$ ($\hat{x}^{k_0}$ is an optimal solution of problem* (1.1)) *or*

$$\lim_{k\to\infty}(\hat{x}^k-x^k)=0 \tag{3.10}$$

and the accumulation point of x^k is a solution of the problem (1.1).

Proof. From the assumptions that H is a nonsingular M-matrix or block-diagonal dominant matrix, then, it is easy to show that H can be divided into H_1 and H_2 ($H=H_1+H_2$) such that (2.11) holds for any $y(y\ne0)\in R^n$, where H_1 is a block diagonal matrix of H consisting of t diagonal blockes of H. As the result, the algorithm A is well defined. Assume that sequence $\hat{x}^k$ and x^k is generated by the algorithm A. If there exists an integer k_0 such that $\hat{x}^{k_0}=x^{k_0}$, then, from (3.3)–(3.4), it is easy to see that $\hat{x}^{k_0}$ (or x^{k_0}) is an optimal solution for the problem (1.1).

If the above description is not true, from (2.11) and (3.9), it is striaghtforward to verify that the sequence $f(x^k)$ is convergent. Thus,

$$\lim_{k\to\infty} \| d^k\|_2 = \lim_{k\to\infty} \|\hat{x}^k - x^k\|_2 = 0 \tag{3.11}$$

which implies that (3.10) holds. Let x^* be a limit of a convergent subsequence of the sequence x^k, and let $\hat{x}^*$ be an accumulation point of the corresponding subsequence of $\hat{x}^k$, By (3.10) or (3.11), it is obvious that $x^* = \hat{x}^*$. Hence

$$\nabla\varphi(x^*, \bar{x}) = \nabla\varphi(\hat{x}^*, \bar{x}) = \nabla f(x^*)$$

From (3.1), the necessary and sufficient conditions of optimality for problem (1.1) hold at x^*, which proves the conclusion of the theorem.

The following corollary is a direct result of Theorem 3.4.

Corollary 3.5. *Suppose that $x^0 \in \Omega$ is an initial point and H is a symmetric positive definite martrix with diagonal dominant, and that H is divided into H_1 and H_2 such that $H = H_1 + H_2$ and H_1 is a block diagonal matrix, and sequence $\hat{x}^k$ and x^k are generated by the algorithm A. Then there exists an integer $k_0 > 0$ such that $\hat{x}^{k_0} = x^{k_0}$ ($\hat{x}^{k_0}$ is an optimal solution of problem* (1.1)) *or*

$$\lim_{k\to\infty} (\hat{x}^k - x^k) = 0 \tag{3.12}$$

and the accumulation point of x^k is a solution of the problem (1.1).

Now we discuss the numerical experiments of our algorithm. The algorithm A was implemented in double precision Fortran, and the code was tested on randomly generated problems. The matrix H is defined by $H = S + vv^T$, The elements of $v \in R^n$ are generated randomly in the interval $(-1, 1)$, and $v^tv = 1$, S is a diagonal matrix, and $S = \alpha I$ ($\alpha > 0$) and I is an identify matrix. Without loss of generality, assume that $l = 0, u = e, e \in R^n$ and all the elements in e are equal to 1, and the number of active constraints at the optimal solution x^* is chosen as $\frac{n}{2}$, and half of the active variables have positive gradient, and the other half have negative. So the optimal solution x^* is chosen such that

$$(\nabla f(x^*))_i = \begin{cases} 1, & x_i^* = 0 \\ -1, & x_i^* = 1 \\ 0, & 0 < x_i^* < 1 \end{cases} \tag{4.7}$$

Finally we set $b = -Hx^* + \nabla f(x^*)$.

If $|f(x^k) - f(x^*)| < \varepsilon$ is used for the termination rule, and set $\varepsilon = 10^{-6}$ and $\alpha = 1.0$, then the computational results are shown in Table 1 for n (from 100 to 1000).

where n, n_i, t and NI denote the size of the problems, the size of subproblem, number of the subproblems and number of iteration for each problem, respectively.

Table 1.

n	n_i×t	NI	$\|x^k - x^*\|_2$	n	n_i×t	NI	$\|x^k - x^*\|_2$
100	10×10	12	0.40632×10^{-3}	600	15×40	15	0.50520×10^{-3}
	5×20	13	0.51831×10^{-3}		30×20	14	0.71878×10^{-3}
	20×5	13	0.29122×10^{-3}		40×15	14	0.54251×10^{-3}
200	20×10	12	0.56617×10^{-3}	700	35×20	15	0.43237×10^{-3}
	40×5	10	0.43943×10^{-3}		70×10	13	0.64135×10^{-3}
	50×4	9	0.40737×10^{-3}		20×35	15	0.57402×10^{-3}
300	30×10	12	0.49600×10^{-3}	800	40×20	15	0.45446×10^{-3}
	10×30	13	0.58339×10^{-3}		20×40	15	0.59770×10^{-3}
	50×6	11	0.31306×10^{-3}		50×16	14	0.67480×10^{-3}
	20×15	13	0.41242×10^{-3}		80×10	13	0.65477×10^{-3}
	15×20	13	0.52632×10^{-3}		10×80	15	0.50097×10^{-3}
400	40×10	13	0.40337×10^{-3}	900	30×30	16	0.45173×10^{-3}
	20×20	14	0.48924×10^{-3}		60×15	15	0.46720×10^{-3}
	50×8	12	0.50962×10^{-3}		15×60	15	0.73217×10^{-3}
	25×16	14	0.40236×10^{-3}		10×90	15	0.59535×10^{-3}
	16×25	14	0.54003×10^{-3}		90×10	14	0.44213×10^{-3}
500	50×10	13	0.54079×10^{-3}	1000	100×10	14	0.45201×10^{-3}
	25×20	14	0.66207×10^{-3}		50×20	15	0.33196×10^{-3}
	20×25	11	0.29122×10^{-3}		20×50	16	0.48603×10^{-3}
	10×50	14	0.75915×10^{-3}		25×40	16	0.49682×10^{-3}
	100×5	11	0.29018×10^{-3}		10×100	15	0.59320×10^{-3}

Table 2.

α	NI1	$\|x^k - x^*\|_2$	NI2	$\|x^k - x^*\|_2$
0.90	16	0.11285×10^{-2}	18	0.10277×10^{-2}
0.80	21	0.11397×10^{-2}	24	0.11242×10^{-2}
0.75	26	0.90089×10^{-3}	30	0.98295×10^{-3}
0.70	32	0.11379×10^{-2}	39	0.10953×10^{-2}
0.65	45	0.10712×10^{-2}	58	0.12360×10^{-2}

The second and third column, and the fourth and fifth column denote the numerical results for $n = 500$ $(n_i = 25, t = 20)$ and $n = 900(n_i = 30, t = 30)$, respectively.

The author wishes to thank Prof. Y.Yuan for many valuable suggestion and comments.

References

[1] D.P. Bertsekas, Projected Newton method for optimization problems with simple constraints, *SIAM J. Control Optimization,* **20** (1982), 221–246.

[2] Zhongzhi Bai, Parallel matrix multisplitting block relaxation methods, *Math. Numer. Sinica,* **17** (1995), 238–252 (in Chinese).

[3] A. Berman , R.J. Plemmons, Nonnegative Matrices in the Mathematical Sciences, Academic Press, New York, 1979.

[4] T.F. Coleman, L.A. Hulbert, A direct active set algorithm for large sparse quadratic program with simple bounds, *Mathematical Programming,* **45** (1989), 373–406.

[5] A.R. Conn, N.I.M. Gould , Ph.L. Toint, Testing a class of methods for solving minimization problems with simple bounds on variables, *Math. Comp.,* **50** (1988), 399–430.

[6] J.C. Dunn, On the convergence of projected gradient processes to singular critical point, *J. Optim. Theory Appl.,* **55** (1987), 203–216.

[7] A. Friedlander, J.M. Martinez, On the maximization of a concave quadratic function with box constraints, *SIAM J. Optimization,* **4** (1994), 177–192.

[8] R. Fletcher, Practical Methods of Optimization, 2-nd ed., John Wiley, Chichester, New York, 1987.

[9] C.G. Han, P.M. Pardalos, Y. Ye, Computational aspects of an interior point algorithm for quadratic programming problem with box constraints, in Large Scale Numerical Optimization, T.Coleman and Y.Ye, eds., Society for Industrial and Applied Mathematics, Philadephia, PA, 1990.

[10] O.L. Mangasarian, Parallel gradient distribution in unconstrained optimizatin, *SIAM J. Control and Optimization,* **4** (1995), 1916–1925.

[11] J.J. More, G. Toraldo, Algorithm for bound constrained quadratic programming problems, *Numerical Math.,* **55** (1989), 277–400.

[12] ——, On the solution of large quadratic programming problems with bound constraints, *SIAM J. Optimization,* **1** (1991), 93–113.

[13] M.P. Mekenna, J.P. Mesirov, S.A. Zenios, Data parallel quadratic programming on box constrained problem, *SIAM J. Optimization,* **4** (1995), 570–589.

[14] R.S. Varga, Matrix Iterative Analysis, Pientice-Hall, Englewood Cliffs, N.J., 1962.

[15] Y. Yuan, Numerical Methods for Nonlinear Programming, Shanghai Scientific and Technical Publishers, 1993 (in Chinese).