Applications of Mathematics

Li Sun; Liang Fang; Guoping He

An active set strategy based on the multiplier function or the gradient

Applications of Mathematics, Vol. 55 (2010), No. 4, 291–304

Persistent URL: http://dml.cz/dmlcz/140401

AN ACTIVE SET STRATEGY BASED ON THE MULTIPLIER FUNCTION OR THE GRADIENT*

Li Sun, Tai'an, Liang Fang, Tai'an, Guoping He, Qingdao

(Received February 21, 2008)

Abstract. We employ the active set strategy which was proposed by Facchinei for solving large scale bound constrained optimization problems. As the special structure of the bound constrained problem, a simple rule is used for updating the multipliers. Numerical results show that the active set identification strategy is practical and efficient.

Keywords: active set, bound constraints, large scale problem

MSC 2010: 90C30, 90C06

1. Introduction

The bound constrained problems are probably the simplest kind of constrained nonlinear programming problems, and they often arise in practice. Actually, most unconstrained problems encountered in applications are only meaningful if the variables belong to some prefixed range of values and should therefore be viewed as bound constrained problems. We are concerned with the solution of simple bound constrained minimization problems of the form

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } l \leqslant x \leqslant u \end{aligned} \tag{1.1}$$

where $x \in \mathbb{R}^n$. The objective function $f(x)$ is assumed to be twice continuously differentiable, l and u are given bound vectors in $\mathbb{R}^n$.

* The work was supported in part by the National Science Foundation of China (10571109, 10901094), Natural Science Foundation of Shandong (Y2008A01) and Technique Foundation of STA (2006GG3210009).

We begin with an overview of the development of active set methods. In this class of methods, a working set estimates the set of active constraints at the solution and it is updated from iteration to iteration. In general, only a single active constraint can be added to or dropped from the working set at each iteration, and this can slow down the convergence rate, especially when dealing with large-scale problems.

In recent years, a number of algorithms have been proposed to add and drop many constraints in an iteration. Moré and Toraldo [10] use the gradient projection method to identify a suitable working face, followed by the conjugate gradient method to explore the face, but its convergence is driven by the gradient projection with the step length satisfying the sufficient decrease condition. Z. Dostál in [3] proposes a proportioning based algorithm which preserves the finite termination property.

Another line of active set research, stemming from the work of Facchinei, has dealt with the study of identification function. Below, we summarize some features of these different techniques.

- The approximate active set identification [7]. Based on a multiplier function, the estimate of the active set $A(x)$ satisfies $I_+ \subseteq A(x) \subseteq I_0$, where I_0 is the index set of the active constraints at the solution and I_+ is the index set of strongly active constraints, i.e. the index set of active constraints with positive multipliers.
- The accurate active set identification [4]. On the basis of identification function, Facchinei-Fisher-Kanzow established a strategy that can identify the accurate active constraints in a certain neighborhood Ω_1 of the optimal solution [4], that is, $A(x) = I_0$, $i \in \Omega_1$. An algorithm in [2] employs this strategy successfully in SSLE.

In this paper we analyze the approximate active set identification strategy. The main features of our QNAS algorithm are shown below.

- QNAS algorithm generates feasible iterates.
- To compute the direction d^k, an identification strategy is employed to predict the active set. The active set identification function is based on the multiplier functions as in [8]. In particular, the identification function works well with the information of the gradient of the objective function.
- QNAS algorithm possesses the global convergent property under the standard assumption.

The paper is organized as follows. In the next section some basic definitions and assumptions are stated. In Section 3, we discuss the construction of the QNAS algorithm, whose global convergence is proved in Section 4. The numerical tests and the conclusion are given in Section 5 and the last section.

At the end of this section, we fix the notation. A superscript k is used to indicate iteration numbers. Furthermore, we often omit the arguments and write, for example,

f^k instead of $f(x^k)$. If H is an $n \times n$ matrix with elements H_{ij}, $i, j = 1, \dots, n$, and I is an index set such that $I \subseteq \{1, \dots, n\}$, we denote by H_I the $|I| \times |I|$ submatrix of H consisting of elements H_{ij}, $i \in I$, $j \in I$. If w is an n vector, we denote by w_I the subvector with components w_i, $i \in I$. Finally, by $\|\cdot\|$we denote the Euclidean norm.

2. Problem formulation and preliminaries

In what follows we indicate by Ω the feasible set of Problem (1.1), that is,

$$\Omega = \{x \in \mathbb{R}^n \colon\ l \leqslant x \leqslant u\}.$$

To guarantee that no unbounded sequences are produced by the minimization process, we make the following standard assumption.

Assumption 1. The feasible set Ω is bounded.

A vector $\overline{x} \in \Omega$ is said to be a stationary point for Problem (1.1) if for every $i = 1, \dots, n$,

$$\begin{cases} l_i = \overline{x}_i \Rightarrow \nabla f_i(\overline{x}) \geqslant 0, \\ l_i < \overline{x}_i < u_i \Rightarrow \nabla f_i(\overline{x}) = 0, \\ \overline{x}_i = u_i \Rightarrow \nabla f_i(\overline{x}) \leqslant 0, \end{cases} \tag{2.1}$$

where $\nabla f_i(\overline{x})$ is the ith component of the gradient vector of f at $\overline{x}$. Strict complementarity is said to hold at $\overline{x}$ if $\nabla f_i(\overline{x}) > 0$ and $\nabla f_i(\overline{x}) < 0$ in the first and third implication of (2.1).

It is well known that the KKT conditions for $\overline{x}$ to solve Problem (1.1) are

$$\begin{cases} \nabla f(\overline{x}) - \overline{\lambda} + \overline{\mu} = 0, \\ \overline{\lambda} \geqslant 0,\ \ (l - \overline{x})^T \overline{\lambda} = 0, \\ \overline{\mu} \geqslant 0,\ \ (\overline{x} - u)^T \overline{\mu} = 0, \\ l \leqslant \overline{x} \leqslant u, \end{cases} \tag{2.2}$$

where $\overline{\lambda} \in \mathbb{R}^n$ and $\overline{\mu} \in \mathbb{R}^n$ are the KKT multipliers.

3. A framework of the algorithm

3.1. Identifying the active constraints

In order to make our algorithm suitable for large-scale bound constrained problems, we define the sets of indices L^k, U^k, F^k of the current iterate x^k estimated to be active, respectively, at their lower bound, upper bound, or estimated to be free:

$$
\begin{aligned}
L^k &= \Big\{i\colon\ x_i^k \leqslant l_i + \min\Big[\varsigma\lambda_i(x^k), \frac{u_i - l_i}{3}\Big]\Big\},\\
U^k &= \Big\{i\colon\ x_i^k \geqslant u_i - \min\Big[\varsigma\mu_i(x^k), \frac{u_i - l_i}{3}\Big]\Big\},\\
F^k &= \{1,\ldots,n\} \setminus (L^k \cup U^k).
\end{aligned} \tag{3.1}
$$

Here ς is a positive constant, in our numerical tests we choose $\varsigma = 10^{-5}$, and $\lambda(x)$, $\mu(x)$ are two multiplier functions [8] defined as

$$\lambda_i(x) = [(u_i - x_i)^2 + (x_i - l_i)^2]^{-1}(x_i - u_i)^2\nabla f_i(x), \tag{3.2}$$

$$\mu_i(x) = -\,[(u_i - x_i)^2 + (x_i - l_i)^2]^{-1}(l_i - x_i)^2\nabla f_i(x). \tag{3.3}$$

We try to employ the identification techniques which allows one to identify exactly the active constraints at the solution without requiring strict complementarity [4] in QNAS, but using this partition of the variables does not guarantee that $L^k \cap U^k = \emptyset$ at each iteration k, which will lead to a misunderstanding when defining the direction.

Next, we investigate the possibility of reducing the computational costs of the active set estimation. The basic idea is to follow a more classical approach, namely, to obtain an approximation of $\overline{\lambda}$ and $\overline{\mu}$ at each iteration, thus avoiding the necessity of using the multiplier functions, which need the computation of $n \times n$ linear system, see (3.2) and (3.3). Considering the first equality of (2.2), we obtain the approximate multipliers easily as follows:

$$\lambda_i^k = \begin{cases} \nabla f_i(x^k) & \text{if } x_i^k = l_i,\\ 0 & \text{otherwise;} \end{cases} \tag{3.4}$$

$$\mu_i^k = \begin{cases} -\nabla f_i(x^k) & \text{if } x_i^k = u_i,\\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

It is easy to see that the estimated multipliers can be determined directly by the gradient of the objective function as the special structure of the bound constrained problems. Employing (3.4) and (3.5) instead of the multiplier function, we obtain the following partition of L^k, U^k, F^k:

$$L^k = \Big\{i\colon\ x_i^k \leqslant l_i + \min\Big[\varsigma \nabla f_i(x^k), \frac{u_i - l_i}{3}\Big]\Big\}, \tag{3.6}$$
$$U^k = \Big\{i\colon\ x_i^k \geqslant u_i + \min\Big[\varsigma \nabla f_i(x^k), \frac{u_i - l_i}{3}\Big]\Big\},$$
$$F^k = \{1, \ldots, n\} \setminus (L^k \cup U^k).$$

The active set identification function (3.6) is similar to that described in [5].

3.2. The scheme of search direction

We indicate the estimation of the active set $L^k \cup U^k$ by A^k. In order to obtain the search direction for the active variables, we partition the active set A^k into three parts,

$$A_1^k = \{i\colon\ (l_i + u_i - 2x_i^k)\nabla f_i(x^k) \geqslant 0 \text{ and } \{x_i^k = l_i \text{ or } x_i^k = u_i\}\}, \tag{3.7}$$
$$A_2^k = \Big\{i\colon\ (l_i + u_i - 2x_i^k)\nabla f_i(x^k) < 0 \text{ and } \Big\{l_i \leqslant x_i^k \leqslant l_i + \min\Big[\varsigma \lambda_i(x), \frac{u_i - l_i}{3}\Big] \text{ or } u_i - \min\Big[\varsigma \mu_i(x), \frac{u_i - l_i}{3}\Big]\Big\} \leqslant x_i^k \leqslant u_i\Big\},$$
$$A_3^k = \Big\{i\colon\ (l_i + u_i - 2x_i^k)\nabla f_i(x^k) \geqslant 0 \text{ and } \Big\{l_i < x_i^k \leqslant l_i + \min\Big[\varsigma \lambda_i(x), \frac{u_i - l_i}{3}\Big] \text{ or } u_i - \min\Big[\varsigma \mu_i(x), \frac{u_i - l_i}{3}\Big]\Big\} \leqslant x_i^k < u_i\Big\}.$$

Here A_1^k is the index set of variables, where the corresponding steepest descent directions head towards the outside of the feasible region. Therefore, it is reasonable that we fix the variables with indices in A_1^k. Further, A_2^k is the index set of the variables, where the steepest descent directions point into the interior of the feasible region, and therefore we can use the steepest direction as a search direction in the corresponding subspace. Finally, A_3^k is the set of active variables, where the steepest decent directions point towards the boundary. Thus the steepest descent directions in this subspace should be truncated to ensure feasibility.

Let P_0^k be the matrix whose columns are $\{e_i;\ i \in F^k\}$, and P_j^k the matrix whose columns are $\{e_i;\ i \in A_j^k\}$ for $j = 1, 2, 3$, where e_i is the ith column of the identity matrix in $\mathbb{R}^{n\times n}$. The search direction at the kth iteration is defined by

$$d^k = P_0^k d_{F^k}^k - (P_2^k P_2^{k^T} \Theta^k + P_3^k P_3^{k^T} \Gamma_k) \nabla f(x^k). \tag{3.8}$$

Here $\Theta^k = \mathrm{diag}(\theta_1^k, \dots, \theta_n^k)$ and $\Gamma^k = \mathrm{diag}(\gamma_1^k, \dots, \gamma_n^k)$ with

$$\theta_i^k = \begin{cases} 0 & \text{if } i \notin A_2^k, \\ \dfrac{x_i^k - u_i}{\nabla f_i(x^k)} & \text{if } l_i \leqslant x_i^k \leqslant l_i + \min[\varrho(x^k, \lambda^k, \mu^k), \varsigma] \text{ and } x_i^k - \nabla f_i(x^k) \geqslant u_i, \\ \dfrac{x_i^k - l_i}{\nabla f_i(x^k)} & \text{if } u_i - \min[\varrho(x^k, \lambda^k, \mu^k), \varsigma] \leqslant x_i^k \leqslant u_i \text{ and } x_i^k - \nabla f_i(x^k) \leqslant l_i, \\ 1 & \text{otherwise,} \end{cases}$$

$$\gamma_i^k = \begin{cases} 0 & \text{if } i \notin A_3^k, \\ \dfrac{x_i^k - l_i}{\nabla f_i(x^k)} & \text{if } l_i < x_i^k \leqslant l_i + \min\Big[\varsigma\lambda_i(x), \dfrac{u_i - l_i}{3}\Big] \text{ and } x_i^k - \nabla f_i(x^k) \leqslant l_i, \\ \dfrac{x_i^k - u_i}{\nabla f_i(x^k)} & \text{if } u_i - \min\Big[\varsigma\mu_i(x), \dfrac{u_i - l_i}{3}\Big] \leqslant x_i^k < u_i \text{ and } x_i^k - \nabla f_i(x^k) \geqslant u_i, \\ 1 & \text{otherwise.} \end{cases}$$

It is easy to conclude that the simple description of $d_{A^k}^k$ is

$$d_i^k = \begin{cases} -\nabla f_i(x^k) & \text{if } l_i \leqslant x_i^k - \nabla f_i(x^k) \leqslant u_i, \\ l_i - x_i^k & \text{if } x_i^k - \nabla f_i(x^k) \leqslant l_i, \\ u_i - x_i^k & \text{if } x_i^k - \nabla f_i(x^k) \geqslant u_i, \end{cases} \tag{3.9}$$

where $i \in A^k$.

The search direction for the inactive variables is chosen as $d_{F^k}^k$, where $d_{F^k}^k$ is the optimal solution of the strictly convex quadratic programming problem

$$\begin{aligned} \min m(d_{F^k}) = \nabla f_{F^k}(x^k)^T d_{F^k} + \frac{1}{2} d_{F^k}^T B_{F^k}^k d_{F^k} \\ \text{s.t. } l_{F^k} - x_{F^k}^k \leqslant d_{F^k} \leqslant u_{F^k} - x_{F^k}^k \end{aligned} \tag{3.10}$$

where $B_{F^k}^k \in \mathbb{R}^{m_k \times m_k}$ is the reduced approximation of the Hessian matrix, m_k is the number of elements in F^k, $B_{F^k}^k = P_0^{k^T} B^k P_0^k$. The approach to updating B^k is based on the recursive BFGS update that discard information corresponding to that part of inactive set that is not changed.

The definition of the search direction (3.8) and that of d_{F^k} in (3.10) and d_{A^k} in (3.9) ensure that

$$l_i \leqslant x_i^k + d_i^k \leqslant u_i$$

holds for $i = 1, \dots, n$.

Lemma 3.1. *If d^k is defined by (3.8), then it satisfies*

$$\nabla f(x^k)^T d^k \leqslant 0 \tag{3.11}$$

and the equality holds only if $d^k = 0$.

P r o o f. Obviously, $d_{F^k} = 0$ is a feasible solution of the quadratic program (3.10), hence

$$\nabla f_{F^k}(x^k)^T d^k_{F^k} + \frac{1}{2} d^{k^T}_{F^k} B^k_{F^k} d^k_{F^k} \leqslant 0, \tag{3.12}$$

$$\nabla f_{F^k}(x^k)^T d^k_{F^k} \leqslant -\frac{1}{2} d^{k^T}_{F^k} B^k_{F^k} d^k_{F^k}.$$

Since $B^k_{F^k}$ is positive definite, so

$$\nabla f_{F^k}(x^k)^T d^k_{F^k} \leqslant 0.$$

Define

$$\tilde{H}_k = P^k_1 P^{k^T}_1 + P^k_2 P^{k^T}_2 \Theta^k + P^k_3 P^{k^T}_3 \Gamma^k$$

and

$$H_k = [P^k_1, P^k_2, P^k_3]^T \tilde{H}_k [P^k_1, P^k_2, P^k_3]. \tag{3.13}$$

It is easy to see that H_k is positive definite. Because $P^{k^T}_1 d^k = 0$, (3.13) gives

$$\nabla f_{A^k}(x^k)^T d^k_{A^k} = -d^{k^T}_{A^k} H_k^{-1} d^k_{A^k} \leqslant 0. \tag{3.14}$$

This indicates that (3.11) is true and that $\nabla f(x^k)^T d^k = 0$ only if $d^k = 0$. □

3.3. The active set quasi-Newton algorithm

Now, we are ready to give the active set quasi-Newton algorithm (QNAS) for solving Problem (1.1).

Step 0. Choose $\sigma \in (0, \frac{1}{2})$, $x^0 \in \mathbb{R}^n$, where x^0 satisfies $l \leqslant x^0 \leqslant u$, compute $f(x^0)$, $\nabla f(x^0)$ and set $k = 0$.

Step 1. Determine the search direction by (3.8), if $d^k = 0$, stop.

Step 2. Find the smallest integer $i = 0, 1, \ldots$ such that

$$f(x^k + 2^{-i} d^k) \leqslant f(x^k) + \sigma 2^{-i} \nabla f(x^k)^T d^k$$

and set $\alpha^k = 2^{-i}$, $x^{k+1} = x^k + \alpha^k d^k$. Determine L^{k+1}, U^{k+1}, and F^{k+1} by (3.1) or (3.6).

Step 3. Update B^{k+1}, $B^{k+1}_{F^{k+1}} = P^{k+1^T}_0 B^{k+1} P^{k+1}_0$, $k := k + 1$, goto *Step 1.*

4. Global convergence analysis

The KKT conditions (2.2) are equivalent to

$$\begin{cases} (l_i + u_i - 2\overline{x}_i)\nabla f_i(\overline{x}) \geqslant 0 & \text{if } i \in \overline{L} \cup \overline{U}, \\ \nabla f_i(\overline{x}) = 0 & \text{if } i \in \overline{F}. \end{cases} \tag{4.1}$$

Here $\overline{L} := \{i\colon\ \overline{x}_i = l_i\}$, $\overline{U} := \{i\colon\ \overline{x}_i = u_i\}$, $\overline{F} := \{1, \ldots, n\} \setminus (\overline{L} \cup \overline{U})$.

Assumption 2. There exist positive scalars c_1, c_2 such that any matrix $B^k_{F^k}$, $k = 1, 2, \ldots$ satisfies

$$c_1\|z\|^2 \leqslant z^T B^k_{F^k} z \leqslant c_2\|z\|^2 \quad \forall\, z \in \mathbb{R}^{m_k},\ \ z \neq 0. \tag{4.2}$$

Here m_k is the number of elements in F^k.

Lemma 4.1. *If Assumptions* 1, 2 *hold,* $x^k \in \Omega$, *and* d^k *is the direction defined by* (3.8), *then*

$$\nabla f(x^k)^T d^k \leqslant -c\|d^k\|^2. \tag{4.3}$$

P r o o f. From (3.12) and (4.2) we have that

$$\nabla f_{F^k}(x^k)^T d^k_{F^k} \leqslant -\frac{c_1}{2}\|d^k_{F^k}\|^2. \tag{4.4}$$

From the definition of $d^k_{A^k}$ in (3.9) we conclude that

1) $d^k_i = -\nabla f_i(x^k)$ if $l_i \leqslant x^k_i - \nabla f_i(x^k) \leqslant u_i$, so $\nabla f_i(x^k) d^k_i \leqslant -(d^k_i)^2$;
2) $d^k_i = l_i - x^k_i$ if $\nabla f_i(x^k) \geqslant x^k_i - l_i$, which means $\nabla f_i(x^k) d^k_i \leqslant -(d^k_i)^2$;
3) $d^k_i = u_i - x^k_i$ if $\nabla f_i(x^k) \leqslant x^k_i - u_i$, hence $\nabla f_i(x^k) d^k_i \leqslant -(d^k_i)^2$.

Define $c = \min\bigl(\frac{c_1}{2}, 1\bigr)$; this implies that (4.3) holds, which completes the proof. □

Lemma 4.2. *If Assumptions* 1, 2 *hold,* $x^k \in \Omega$, *and* d^k *is the direction defined by* (3.9), *then*

$$d^k = 0 \iff x^k \text{ is a KKT point of } f \text{ on } \Omega.$$

P r o o f. First we suppose that $d^k = 0$.
If $i \in A^k$, then according to (3.8) we have

$$P^k_2 P^{k^T}_2 \nabla f(x^k) = 0, \quad P^k_3 P^{k^T}_3 \Gamma_k \nabla f(x^k) = 0.$$

Because $\gamma_i^k \neq 0$ for $i \in A_3^k$, it follows that

$$P_j^{k^T} \nabla f(x^k) = 0, \quad j = 2, 3.$$

Therefore, $\nabla f_i(x^k) = 0$ if $i \in A_2^k \cup A_3^k$. By the definition of the multiplier functions (3.2) and (3.3), we have $\lambda_i(x^k) = 0$ and $\mu_i(x^k) = 0$ for $i \in A_2^k \cup A_3^k$.

For $i \in A_1^k$, if $x_i^k = l_i$, then $\nabla f_i(x^k) \geqslant 0$ by (3.2), and we have $\lambda_i(x^k) \geqslant 0$. Analogously, if $x_i^k = u_i$, we have $\mu_i(x^k) \geqslant 0$.

To establish that x^k is a KKT point of f on Ω, it is sufficient to prove that $l_i < x_i^k < u_i$ and $\nabla f_i(x^k) = 0$ for each $i \in F^k$. If $i \in F^k$, we have

$$x_i^k > l_i + \min\Big[\varsigma\lambda_i(x), \frac{u_i - l_i}{3}\Big], \quad x_i^k < u_i - \min\Big[\varsigma\mu_i(x), \frac{u_i - l_i}{3}\Big]. \tag{4.5}$$

Suppose that there exists an $i \in F^k$ such that $\nabla f_i(x^k) < 0$. Then for sufficiently small $\varepsilon > 0$, the vector $\tilde{d}_{F^k}$ defined by

$$\tilde{d}_j = \begin{cases} 0 & \text{if } j \in F^k \setminus \{i\}, \\ \varepsilon & \text{if } j = i \end{cases}$$

satisfies $l_{F^k} - x_{F^k}^k \leqslant \tilde{d}_{F^k} \leqslant u_{F^k} - x_{F^k}^k$, and

$$m(\tilde{d}_{F^k}) = \nabla f_i(x^k)\varepsilon + \frac{1}{2}\varepsilon^2 B_{ii}^k < 0.$$

This is impossible, since $d_{F^k} = 0$ is the optimal solution of (3.10). We could prove in a similar way that $\nabla f_i(x^k)$ cannot be positive. Hence, $\nabla f_i(x^k) = 0$ for each $i \in F^k$. By (3.4) and (3.5), we have $\lambda_i^k = 0$, $\mu_i^k = 0$, $i \in F^k$.

The statements mentioned above prove that x^k is a KKT point of f on Ω.

Now suppose that x^k is a KKT point of f on Ω. From (3.7) and (4.1) it follows that $A_2^k = \emptyset$, $A_3^k = \emptyset$, therefore $d_{A^k} = 0$.

On the other hand, $d_{F^k} = 0$ is a feasible solution of the quadratic programming problem (3.10). Since $\nabla f_{F^k}(x^k) = 0$ and $B_{F^k}^k$ is a positive definite matrix,

$$m(d_{F^k}) = \frac{1}{2} d_{F^k}^T B_{F^k}^k d_{F^k} \geqslant 0.$$

Hence, $d_{F^k} = 0$ is the optimal solution of the quadratic programming problem (3.10). □

Theorem 4.3. *Suppose that Assumptions* 1, 2 *are satisfied. Assume that* f *is twice continuously differentiable in* Ω, $d^k \to 0$, *and that* $x^k \to \overline{x}$, *where* d^k *is the direction defined by* (3.8). *Then* $\overline{x}$ *is a KKT point of Problem* (1.1).

Proof. Since $\overline{x}$ is the accumulation point of $\{x^k\}$, there exists a subsequence $\{x^{k_i}\}$, $i = 1, 2, \ldots$, such that

$$\lim_{i\to\infty} x^{k_i} = \overline{x}. \tag{4.6}$$

Define $\overline{A} = \{i\colon\ \overline{x}_i = l_i \text{ or } \overline{x}_i = u_i\}$. If $\overline{x}$ is not a KKT point, there exists $j \in \overline{A}$ such that

$$(l_j + u_j - 2\overline{x}_j)\nabla f_j(\overline{x}) < 0 \tag{4.7}$$

or there exists $j \notin \overline{A}$ such that

$$\nabla f_j(\overline{x}) \neq 0. \tag{4.8}$$

If (4.7) holds for some $j \in \overline{A}$, then $j \in A_2(x^{k_i})$ for all sufficiently large i.

But $\lim\limits_{k\to\infty} \|P_2^k \nabla f(x^k)\| = 0$ shows that

$$\nabla f_j(\overline{x}) = 0, \quad j \in A_2(\overline{x}),$$

which contradicts (4.7). So it remains to prove that $\nabla f_{\overline{F}}(\overline{x}) = 0$. We recall that d_F^k is the solution of the quadratic programming problem

$$\begin{gathered}\min \nabla f_F(x^k)^T d_F + \frac{1}{2} d_F^T B_F^k d_F\\ \text{s.t. } l_F - x_F^k \leqslant d_F \leqslant u_F - x_F^k.\end{gathered}$$

Since $d^k \to 0$, the continuity of the optimal solution of a strictly convex quadratic programming problem under perturbations implies that zero is the optimal solution of

$$\begin{gathered}\min \nabla f_F(\overline{x})^T d_F + \frac{1}{2} d_F^T \overline{B}_F d_F\\ \text{s.t. } l_F - \overline{x}_F \leqslant d_F \leqslant u_F - \overline{x}_F.\end{gathered}$$

Hence, $\nabla f_F(\overline{x}) = 0$ by reasons similar to those used in the proof of Lemma 4.2. $\square$

5. Numerical tests

In this section some numerical results are reported. The code was written in MATLAB with double precision. For each problem, the termination condition is the Euclidean norm of the search direction below 10^{-5}, namely,

$$\|d^k\| \leqslant 10^{-5}.$$

In QNAS, we choose $\varsigma = 10^{-5}$, $\sigma = 10^{-1}$ in all runs. In order to compare (3.1) and (3.6) in identifying the active set, we use the technique in [6] for generating bound constrained optimization problems with known characteristics. The test problems were chosen from [13].

Let an unconstrained problem

$$\min_{x\in\mathbb{R}^n} g(x) \tag{5.1}$$

be given, where g is a twice continuously differentiable function. Let $\overline{x}$ be a local minimum of this unconstrained problem. The bound constrained problem we will generate has the same solution $\overline{x}$. We start by choosing an arbitrary partition of the index set $\{1,\dots,n\}$ into three subsets $\overline{L}$, $\overline{F}$ and $\overline{U}$. They are the sets of indices of the variables that are at a lower bound, free, and at an upper bound at $\overline{x}$, respectively.

We choose the vectors l and u to satisfy the relationships

$$\begin{aligned} l_{\overline{L}} &= \overline{x}_{\overline{L}} < u_{\overline{L}},\\ l_{\overline{F}} &< \overline{x}_{\overline{F}} < u_{\overline{F}},\\ l_{\overline{U}} &< \overline{x}_{\overline{U}} = u_{\overline{F}}. \end{aligned} \tag{5.2}$$

Now consider the objective function

$$f(x) = g(x) + \sum_{i\in\overline{L}} h_i(x_i) - \sum_{i\in\overline{U}} h_i(x_i), \tag{5.3}$$

where $h_i\colon \mathbb{R}\to\mathbb{R}$, $i\in\overline{L}\cup\overline{U}$, are twice continuously differentiable nondecreasing functions.

It follows from (5.2) and (5.3) that $\overline{x}$ is a local minimum of the bound constrained optimization problem

$$\begin{aligned} &\min f(x)\\ &\text{s.t. } l\leqslant x\leqslant u. \end{aligned} \tag{5.4}$$

If $\overline{x}$ is just a stationary point of (5.1), since $\nabla h_i(\overline{x}) \geqslant 0$ for $i \in \overline{L} \cup \overline{U}$, then $\overline{x}$ is a stationary point of problem (5.4) as well.

The possible choices for the function h_i can be

$$(5.5) \qquad \begin{aligned} &(1) \quad \varpi_i(x_i - \overline{x}_i), \\ &(2) \quad \kappa_i(x_i - \overline{x}_i)^3 + \varpi_i(x_i - \overline{x}_i), \\ &(3) \quad \kappa_i(x_i - \overline{x}_i)^{7/3} + \varpi_i(x_i - \overline{x}_i), \end{aligned}$$

where κ_i, ϖ_i are nonnegative constants. Considering the KKT conditions at $\overline{x}$, it is easy to see that the Lagrange multipliers associated with the constraints $l_{\overline{L}} \leqslant x_{\overline{L}}$ and $u_{\overline{U}} \geqslant x_{\overline{U}}$ are ϖ_i for $i \in \overline{L} \cup \overline{U}$.

By this kind of strategy, the number and position of the constraints and of the active constraints, the Lagrange multipliers, and the shape of the feasible region can be easily controlled.

The number of iterations (IT), the final function value (FF) and the CPU time (CPU) to obtain the solution through QNAS with (3.6) and (3.1) are given in the form of IT/FF/CPU in Tab. 1. We observe that the identification of (3.1) and (3.6) both work well, while (3.1) needs the additional computation of an $n \times n$ linear system.

	n	QNAS with (3.1)	QNAS with (3.6)
TP1	10000	16/9.6531e + 02/22.9690	20/9.3013e + 02/17.5000
TP2	10000	21/1.8705e – 007/11.9840	24/6.3826e – 011/10.3590
TP3	10000	187/1.8495e – 007/301.4690	193/8.8499e – 007/253.0150
TP4	5000	39/4.3915e – 008/18.7030	36/1.6289e – 007/16.5940
TP5	5000	80/1.748e – 014/55.2500	77/4.9702e – 017/52.8590
TP6	5000	206/2.2923e – 007/53.4220	207/2.0347e – 008/48.2970
TP7	5000	426/4.4702e – 007/135.9530	368/2.3013e – 007/99.1880
TP8	5000	59/3.6916e – 014/12.3280	87/3.7323e – 014/17.7190
TP9	5000	62/1.3083e – 015/18.1410	65/6.9232e – 022/20.7660

Table 1. Test results on 9 Test Problems.

6. Conclusion and the future work

An active set quasi-Newton method is analyzed in this paper. The active set strategy which belongs to the approximate active set identification allows quick change in the working set, it is suitable for solving large scale problems. As the special structure of the KKT system of the bound constrained optimization, the multipliers can be determined directly by the gradient. Numerical results show that QNAS is practical and efficient. However, QNAS requires the strict complementarity assumption to obtain the superlinear convergence rate as shown in [5]. Consequently, how to employ the accurate active set identification [4] in QNAS or how to obtain a feasible search direction of the inactive variables instead of solving the strictly quadratic programming problem (3.10) remains to be investigated in future.

Acknowledgments. We would like to thank the anonymous referees for their helpful comments.

References

[1] *J. V. Burke, J. J. Moré, G. Toraldo*: Convergence properties of trust region methods for linear and convex constraints. Math. Program. *47* (1990), 305–336.

[2] *L. F. Chen, Y. L. Wang, G. P. He*: A feasible active set QP-free method for nonlinear programming. SIAM J. Optim. *17* (2006), 401–429.

[3] *Z. Dostál*: A proportioning based algorithm with rate of convergence for bound constrained quadratic programming. Numer. Algorithms *34* (2003), 293–302.

[4] *F. Facchinei, A. Fischer, C. Kanzow*: On the accurate identification of active constraints. SIAM J. Optim. *9* (1998), 14–32.

[5] *F. Facchinei, J. Júdice, J. Soares*: An active set Newton algorithm for large-scale nonlinear programs with box constraints. SIAM J. Optim. *8* (1998), 158–186.

[6] *F. Facchinei, J. Júdice, J. Soares*: Generating box-constrained optimization problems. ACM Trans. Math. Softw. *23* (1997), 443–447.

[7] *F. Facchinei, S. Lucidi*: Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. J. Optimization Theory Appl. *85* (1995), 265–289.

[8] *F. Facchinei, S. Lucidi, L. Palagi*: A truncated Newton algorithm for large scale box constrained optimization. SIAM J. Optim. *12* (2002), 1100–1125.

[9] *D. C. Liu, J. Nocedal*: On the limited memory BFGS method for large scale optimization. Math. Program. *45* (1989), 503–528.

[10] *J. J. Moré, G. Toraldo*: On the solution of large quadratic programming problems with bound constraints. SIAM J. Optim. *1* (1991), 93–113.

[11] *Q. Ni, Y. Yuan*: A subspace limited memory quasi-Newton algorithm for large-scale nonlinear bound constrained optimization. Math. Comput. *66* (1997), 1509–1520.

[12] *G. Di Pillo, F. Facchinei, L. Grippo*: An *RQP* algorithm using a differentiable exact penalty function for inequality constrained problems. Math. Program. *55* (1992), 49–68.

[13] *K. Schittkowski*: More test examples for nonlinear programming codes. Lecture Notes in Economics and Mathematical Systems, Vol. 282. Springer, Berlin, 1987.

[14] *L. Sun, G. P. He, Y. L. Wang, L. Fang*: An active set quasi-Newton method with projected search for bound constrained minimization. Comput. Math. Appl. *58* (2009), 161–170.

[15] *L. Sun, G. P. He, Y. L. Wang, C. Y. Zhou*: An accurate active set Newton method for large scale bound constrained optimization. Appl. Math. Accepted.

[16] *Y. L. Wang, L. F. Chen, G. P. He*: Sequential systems of linear equations method for general constrained optimization without strict complementarity. J. Comput. Appl. Math. *182* (2005), 447–471.

[17] *Y. H. Xiao, Z. X. Wei*: A new subspace limited memory BFGS algorithm for large-scale bound constrained optimization. Appl. Math. Comput. *185* (2007), 350–359.

[18] *C. Y. Zhou, G. P. He, Y. L. Wang*: A new constraints identification technique-based QP-free algorithm for the solution of inequality constrained minimization problems. J. Comput. Math. *24* (2006), 591–608.

Authors' addresses: *L. Sun* (corresponding author), College of Information Sciences and Engineering, Shandong Agricultural University, 271018, Tai'an, P. R. China, e-mail: sunlishi@hotmail.com; *L. Fang*, Department of Mathematics and System Science, Taishan University, 271021, Tai'an, P. R. China; e-mail: fangliang3@hotmail.com; *G. He*, College of Information Science and Engineering, Shandong University of Science and Technology, 266510, Qingdao, P. R. China, e-mail: hegp@263.net.