

Applications of Mathematics

Li Sun; Guoping He; Yongli Wang; Changyin Zhou

An accurate active set Newton algorithm for large scale bound constrained optimization

Applications of Mathematics, Vol. 56 (2011), No. 3, 297–314

Persistent URL: <http://dml.cz/dmlcz/141488>

Terms of use:

© Institute of Mathematics AS CR, 2011

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

AN ACCURATE ACTIVE SET NEWTON ALGORITHM FOR
LARGE SCALE BOUND CONSTRAINED OPTIMIZATION*LI SUN, Tai'an and Shanghai, GUOPING HE, Qingdao,
YONGLI WANG, Qingdao, CHANGYIN ZHOU, Qingdao

(Received September 22, 2008)

Abstract. A new algorithm for solving large scale bound constrained minimization problems is proposed. The algorithm is based on an accurate identification technique of the active set proposed by Facchinei, Fischer and Kanzow in 1998. A further division of the active set yields the global convergence of the new algorithm. In particular, the convergence rate is superlinear without requiring the strict complementarity assumption. Numerical tests demonstrate the efficiency and performance of the present strategy and its comparison with some existing active set strategies.

Keywords: active set, bound constraints, Newton method, strict complementarity

MSC 2010: 90C30, 90C06

1. INTRODUCTION

In this paper, we are concerned with the solution of the following simple bound constrained minimization:

$$(1.1) \quad \min_{l \leq x \leq u} f(x),$$

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

Algorithms for solving problem (1.1) fall mostly in the active set category. In this class of methods, a working set is defined to estimate the set of active constraints at the solution and it is updated from iteration to iteration. Early methods for

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

bound constrained problems restrict the changing of the working set by dropping or adding only one constraint at each iteration [26]. This implies that if there are $k \leq n$ constraints active at the solution yet the initial point is in the interior of the feasible region, this method will require at least k iterations to converge. Obviously, it is not suitable for large scale problems.

In recent years, a number of algorithms have been proposed to add and drop several constraints in an iteration (see [7], [10], [12], [13], [16], [20], [21], [22], [30]).

The gradient projection method is a constructive method (see [7], [16], [20], [21]). Moré and Toraldo [21] used the gradient projection method to identify a suitable working face, and then used the conjugate gradient method to explore the face. However, its convergence is driven by the gradient projection with the step length satisfying a sufficient decrease condition and there is no numerical evidence of its effectiveness in the non-quadratic case. More recently, Hager and Zhang [16] indicated that more constraints could be added and dropped in a single iteration by using the adaptive nonmonotone projection search. Furthermore, when the objective function is strongly convex quadratic, their algorithm converges in a finite number of iterations, even when the strict complementarity assumption does not hold.

In recent years, researches have been done to accelerate the convergence rate. Many efficient methods such as Newton methods and trust region methods for unconstrained optimization have been successfully extended to handle the presence of bounds on the variables [4], [5], [19] and their local superlinear/quadratic convergence have been established [10], [12], [13], [20]. A trust region version of Newton's method for bound constrained problems is analyzed in [20]. It also uses a projected search during the subspace minimization phase, which allows adding more indices to the active set in one step. Global and superlinear convergence hold without the strict complementarity assumption.

The affine scaling interior point method of Coleman and Li (see [1], [2]) is a different approach to problem (1.1). This method is based on a reformulation of the necessary optimality conditions obtained by multiplication by a scaling matrix. In a recent work Heinkenschloss et al. [17] analyzed an interior point method without assuming strict complementarity, but they proved only local convergence.

We finally mention some algorithms based on the active set identification function. A guessing technique was first employed to predict which bounds are active at the solution in [10]. This procedure permits fast changes in the working set. The calculation of search direction d^k is based on an identification technique of the set of active constraints and the solution of KKT-like equation for (1.1). However, in order to prove the local superlinear convergence, the method in [10] must resort to a particular-defined multiplier function, which involves solving a product form of the linear system. Later, Facchinei and Lucidi [13] employed a similar identification

technique to estimate the active set. Unlike the algorithms for bound constrained problems that we have reviewed, it generates iterates that need not be feasible by employing a differentiable exact penalty function.

In this paper, we propose a new algorithm for the solution of problem (1.1). The new algorithm is based on the accurate active set identification technique proposed in [9], [24]. It generates feasible iterates. To compute the direction d^k , an identification technique is employed to predict which bounds are active at the solution. At each iteration k , we compute two sets of indices L^k, U^k of the variables that we suppose will be, respectively, at their lower and upper bounds at the solution. We also compute an estimate F^k of the variables we believe to be free. The inactive variable x_{F^k} is updated by the solution of a lower dimensional quadratic bound constrained problem while the active variables, x_{L^k} and x_{U^k} , are updated by subspace gradient and subspace modified gradient directions. Under mild conditions, the new algorithm is globally convergent. In particular, the convergence rate is proved to be superlinear without the strict complementarity 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 algorithm. Global convergence of the algorithm is proved in Section 4, and its local convergent property is given in Section 5. We also give some numerical tests in the last section.

We end this section with a few words on the notation. A superscript k is used to indicate the iteration number. For simplicity, 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 $|I| \times |I|$ sub-matrix of H consisting of elements H_{ij} , where $i, j \in I$. If w is an n -dimensional vector, we denote by w_I the sub-vector with components w_i , $i \in I$. Finally, we denote by $\|\cdot\|$ the Euclidean norm.

2. PROBLEM FORMULATION AND PRELIMINARIES

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

$$\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}.$$

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

Assumption 1. The level set $L^0 = \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\} \cap \Omega$ is compact.

This assumption will be used to guarantee that the algorithm generates a bounded sequence so that at least one accumulation point exists.

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

$$(2.1) \quad \begin{cases} \nabla f_i(\bar{x}) \geq 0 & \forall i \in \bar{L}, \\ \nabla f_i(\bar{x}) = 0 & \forall i \in \bar{F}, \\ \nabla f_i(\bar{x}) \leq 0 & \forall i \in \bar{U}, \end{cases}$$

where $\nabla f_i(\bar{x})$ is the i th component of the gradient vector of f at \bar{x} . We define $\bar{L} := \{i: \bar{x}_i = l_i\}$, $\bar{U} := \{i: \bar{x}_i = u_i\}$, $\bar{F} := \{1, \dots, n\} \setminus (\bar{L} \cup \bar{U})$. Strict complementarity is said to hold at \bar{x} , if $\nabla f_i(\bar{x}) > 0$ and $\nabla f_i(\bar{x}) < 0$, respectively, in the first and third implication of (2.1).

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

$$(2.2) \quad \begin{cases} \nabla f(\bar{x}) - \bar{\lambda} + \bar{\mu} = 0, \\ \bar{\lambda} \geq 0, (l - \bar{x})^\top \bar{\lambda} = 0, \\ \bar{\mu} \geq 0, (\bar{x} - u)^\top \bar{\mu} = 0, \\ l \leq \bar{x} \leq u, \end{cases}$$

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

In view of the definitions of \bar{L} , \bar{U} , and \bar{F} , the above KKT conditions are equivalent to the following system:

$$(2.3) \quad \begin{cases} (l_i + u_i - 2\bar{x}_i)\nabla f_i(\bar{x}) \geq 0 & \text{if } i \in \bar{L} \cup \bar{U}, \\ \nabla f_i(\bar{x}) = 0 & \text{if } i \in \bar{F}. \end{cases}$$

3. FRAMEWORK OF THE NEW ALGORITHM

3.1. The scheme of search direction

In order to make our algorithm suitable for large scale bound constrained problems, we define the sets of indices $L(x)$, $U(x)$, and $F(x)$ as follows:

$$(3.1) \quad \begin{aligned} L(x) &= \{i: l_i \leq x_i \leq l_i + \min[\varrho(x, \lambda, \mu), \varsigma]\}, \\ U(x) &= \{i: u_i - \min[\varrho(x, \lambda, \mu), \varsigma] \leq x_i \leq u_i\}, \\ F(x) &= \{1, \dots, n\} \setminus (L(x) \cup U(x)). \end{aligned}$$

We use the positive scalar $\varsigma \in (0, \tau)$, where $\tau = \min_{i=1, \dots, m} \frac{1}{3}(u_i - l_i)$, to guarantee that $L(x) \cap U(x) = \emptyset$, so that $F(x)$ is well defined. Similarly to [9], we introduce the

function $\varrho: \mathbb{R}^{3n} \rightarrow [0, \infty)$:

$$(3.2) \quad \varrho(x, \lambda, \mu) := \sqrt{\|\Phi(x, \lambda, \mu)\|},$$

where the operator $\Phi: \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$ is defined by

$$\Phi(x, \lambda, \mu) := \begin{pmatrix} \nabla_x L(x, \lambda, \mu) \\ \min(x - l, \lambda) \\ \min(u - x, \mu) \end{pmatrix},$$

and

$$L(x, \lambda, \mu) := f(x) - \lambda^\top(x - l) - \mu^\top(u - x)$$

is the Lagrangian function of problem (1.1).

It then follows from Theorem 3.7 in [9] that $\varrho(x, \lambda, \mu)$ is an identification function for the active set of a KKT point of (1.1). Furthermore, it can accurately identify the active constraints in a certain neighborhood of a solution.

In the new algorithm, the variables with indices in $L(x)$ or $U(x)$ are called active variables, while the variables with indices in $F(x)$ are called inactive variables. The algorithm uses the limited memory quasi-Newton method to update the inactive variables, while the active variables go directly to the boundary or go inside the feasible region.

Now let x^k be a current point at iteration k . Consider the sets $L^k = L(x^k)$, $U^k = U(x^k)$ and $F^k = F(x^k)$, and define the directions of the inactive and active variables by the following statements.

The search direction of the inactive variables $d_{F^k}^k \in \mathbb{R}^{m_k}$, where m_k is the number of elements in F^k , is the optimal solution of the quadratic program

$$(3.3) \quad \begin{aligned} \min m(d_{F^k}) &= \nabla f_{F^k}(x^k)^\top d_{F^k} + \frac{1}{2} d_{F^k}^\top B_{F^k}^k d_{F^k} \\ \text{s.t. } l_{F^k} - x_{F^k}^k &\leq d_{F^k} \leq u_{F^k} - x_{F^k}^k. \end{aligned}$$

Here $B_{F^k}^k = P_0^{k\top} B^k P_0^k$ is a symmetric positive definite matrix, where P_0^k is the matrix whose columns are $\{e_i: i \in F^k\}$, e_i is the i th column of the identity matrix in $\mathbb{R}^{n \times n}$, B^k is the Hessian matrix. For an arbitrary $v \in \mathbb{R}^n$, $v_{F^k} = (P_0^k)^\top v$.

To keep the matrix $B_{F^k}^k$ positive, we employ the Modified Cholesky Factorization (MC) algorithm proposed in § 4.4 in [15].

In our numerical test, we employ the sparse discrete Newton method, see § 4.8 in [15] for more details. It takes advantage of the known sparsity and structure in the Hessian matrix to reduce the requirements of computation and storage.

Finally, we define the search direction of the active variables. We denote the set $L^k \cup U^k$ by A^k and partition the active set A^k into three parts,

$$(3.4) \quad \begin{aligned} A_1^k &= \{i: (l_i + u_i - 2x_i^k)\nabla f_i(x^k) \geq 0 \text{ and } \{x_i^k = l_i \text{ or } x_i^k = u_i\}\}, \\ A_2^k &= \{i: (l_i + u_i - 2x_i^k)\nabla f_i(x^k) < 0 \text{ and } \{l_i \leq x_i^k \leq l_i + \min[\varrho(x, \lambda, \mu), \varsigma] \\ &\quad \text{or } u_i - \min[\varrho(x, \lambda, \mu), \varsigma] \leq x_i^k \leq u_i\}\}, \\ A_3^k &= \{i: (l_i + u_i - 2x_i^k)\nabla f_i(x^k) \geq 0 \text{ and } \{l_i < x_i^k \leq l_i + \min[\varrho(x, \lambda, \mu), \varsigma] \\ &\quad \text{or } u_i - \min[\varrho(x, \lambda, \mu), \varsigma] \leq x_i^k < u_i\}\}. \end{aligned}$$

Here A_1^k is the index set of variables which satisfy the KKT conditions. It is reasonable that we fix the variables with indices in A_1^k , as the corresponding steepest descent directions head towards the outside of the feasible region; A_2^k and A_3^k are the sets of active variables where the steepest descent directions move into the interior or toward the boundary of the feasible region. Thus we can use the steepest descent directions as a search direction in the former case, and truncate the steepest descent directions to ensure feasibility in the latter. We have to specify that the definition of (3.4) divides the active variables into two parts, one that satisfies the KKT conditions and the other that violates the KKT conditions. This kind of division enables us to obtain the global convergence of the algorithm.

Let P_j^k be the matrix whose columns are $\{e_i: i \in A_j^k\}$ for $j = 1, 2, 3$. The search direction at the k th iteration is defined by

$$(3.5) \quad d^k = P_0^k d_{F^k}^k - (P_2^k P_2^{k\top} \Theta^k + P_3^k P_3^{k\top} \Gamma^k) \nabla f(x^k).$$

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

$$(3.6) \quad \theta_i^k = \begin{cases} 0 & \text{if } i \notin A_2^k, \\ \frac{x_i^k - u_i}{\nabla f_i(x^k)} & \text{if } l_i \leq x_i^k \leq l_i + \min[\varrho(x^k, \lambda^k, \mu^k), \varsigma] \\ & \text{and } x_i^k - \nabla f_i(x^k) \geq u_i, \\ \frac{x_i^k - l_i}{\nabla f_i(x^k)} & \text{if } u_i - \min[\varrho(x^k, \lambda^k, \mu^k), \varsigma] \leq x_i^k \leq u_i \\ & \text{and } x_i^k - \nabla f_i(x^k) \leq l_i, \\ 1 & \text{otherwise,} \end{cases}$$

$$(3.7) \quad \gamma_i^k = \begin{cases} 0 & \text{if } i \notin A_3^k, \\ \frac{x_i^k - l_i}{\nabla f_i(x^k)} & \text{if } l_i < x_i^k \leq l_i + \min[\varrho(x^k, \lambda^k, \mu^k), \varsigma] \\ & \text{and } x_i^k - \nabla f_i(x^k) \leq l_i, \\ \frac{x_i^k - u_i}{\nabla f_i(x^k)} & \text{if } u_i - \min[\varrho(x^k, \lambda^k, \mu^k), \varsigma] \leq x_i^k < u_i \\ & \text{and } x_i^k - \nabla f_i(x^k) \geq u_i, \\ 1 & \text{otherwise.} \end{cases}$$

It is easy to establish the simple description of $d_{A^k}^k$,

$$(3.8) \quad d_i^k = \begin{cases} -\nabla f(x^k) & \text{if } l_i \leq x_i^k - \nabla f_i(x^k) \leq u_i, \\ l_i - x_i^k & \text{if } x_i^k - \nabla f_i(x^k) \leq l_i, \\ u_i - x_i^k & \text{if } x_i^k - \nabla f_i(x^k) \geq u_i, \end{cases}$$

where $i \in A^k$. For the purpose of proving the global convergence of the algorithm, we use $d_{A^k}^k$ in the form of (3.5).

From the definition of the search direction d^k , $d_{F^k}^k$, Θ^k , and Γ^k , we know that

$$(3.9) \quad l_i \leq x_i^k + d_i^k \leq u_i$$

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

Lemma 3.1. *Suppose that d^k is defined by (3.5). Then for all k ,*

$$(3.10) \quad \nabla f(x^k)^\top d^k \leq 0$$

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

Proof. Obviously, $d_{F^k} = 0$ is a feasible solution of the quadratic programming problem (3.3). Hence,

$$\nabla f_{F^k}(x^k)^\top d_{F^k}^k + \frac{1}{2} d_{F^k}^{k\top} B_{F^k}^k d_{F^k}^k \leq 0,$$

that is,

$$\nabla f_{F^k}(x^k)^\top d_{F^k}^k \leq -\frac{1}{2} d_{F^k}^{k\top} B_{F^k}^k d_{F^k}^k.$$

Since $B_{F^k}^k$ is positive definite, it follows that

$$\nabla f_{F^k}(x^k)^\top d_{F^k}^k \leq 0,$$

and the equality holds only if $d_{F^k}^k = 0$.

Define

$$\tilde{H}_k = P_2^k P_2^{k\top} \Theta^k + P_3^k P_3^{k\top} \Gamma^k,$$

and let $P \in \mathbb{R}^{n \times |A^k|}$ be the matrix whose columns are $\{e_i : i \in A^k\}$. Then

$$(3.11) \quad H_k = P^\top \tilde{H}_k P.$$

It is easy to see that H_k is semi-positive definite, and (3.11) gives

$$(3.12) \quad \nabla f_{A^k}(x^k)^\top d_{A^k}^k = -\nabla f_{A^k}(x^k)^\top H_k \nabla f_{A^k}(x^k) \leq 0.$$

This implies that (3.10) is true and $\nabla f(x^k)^\top d^k = 0$ only if $d^k = 0$. \square

3.2. The accurate active set Newton algorithm

Now, we are ready to present our algorithm for solving problem (1.1).

Algorithm 3.2 (AASN)

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

Step 1. Determine the Lagrange multipliers λ^k and μ^k by

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

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

Compute the identification function $\varrho(x^k, \lambda^k, \mu^k)$ by (3.2), $L^k = L(x^k)$, $U^k = U(x^k)$, $F^k = F(x^k)$ by (3.1), let P_0^k be the matrix whose columns are $\{e_i : i \in F^k\}$, and set $B_{F^k}^k = P_0^{k\top} B^k P_0^k$.

Step 2. Determine the search direction d^k by (3.5). If $d^k = 0$, stop.

Step 3. Find the smallest integer $i = 0, 1, \dots$, such that

$$f(x^k + 2^{-i} d^k) \leq f(x^k) + \sigma 2^{-i} \nabla f(x^k)^\top d^k.$$

Set $\alpha^k = 2^{-i}$, $x^{k+1} = x^k + \alpha^k d^k$.

Step 4. Set $B^{k+1} = \nabla^2 f(x^{k+1})$, $k := k + 1$, goto Step 1.

We should specify that L^k , U^k , F^k in Step 1 could be determined by different kinds of identification functions, which we will compare in our numerical tests.

4. GLOBAL CONVERGENCE ANALYSIS

In this section, we will show that Algorithm 3.2 (AASN) is globally convergent. To this end, we suppose that the following standard assumption holds.

Assumption 2. There exist positive scalars c_1, c_2 such that for all $k = 1, 2, \dots$, the matrix $B_{F^k}^k$ satisfies

$$(4.1) \quad c_1 \|z\|^2 \leq z^\top B_{F^k}^k z \leq c_2 \|z\|^2 \quad \forall z \in \mathbb{R}^{m_k}, \quad z \neq 0,$$

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

Lemma 4.1. *If Assumptions 1 and 2 hold and d^k is defined by (3.5), then for $x^k \in \Omega$ there exists a positive scalar $c > 0$ such that*

$$(4.2) \quad \nabla f(x^k)^\top d^k \leq -c \|d^k\|^2.$$

Proof. Since $B_{F^k}^k$ is a symmetric positive definite matrix, $d_{F^k}^k$ is the solution of (3.3) if and only if

$$\begin{aligned} l_i - x_i^k &= (d_{F^k}^k)_i \Rightarrow (\nabla f_{F^k}(x^k) + B_{F^k}^k d_{F^k}^k)_i \geq 0, \\ l_i - x_i^k < (d_{F^k}^k)_i < u_i - x_i^k &\Rightarrow (\nabla f_{F^k}(x^k) + B_{F^k}^k d_{F^k}^k)_i = 0, \\ (d_{F^k}^k)_i = u_i - x_i^k &\Rightarrow (\nabla f_{F^k}(x^k) + B_{F^k}^k d_{F^k}^k)_i \leq 0. \end{aligned}$$

As $x^k \in \Omega$, each lower (upper) bound of the quadratic program (3.3) is negative (positive), it follows that

$$\begin{aligned} \nabla f_{F^k}(x^k)^\top d_{F^k}^k + \frac{1}{2} (d_{F^k}^k)^\top B_{F^k}^k d_{F^k}^k &\leq \nabla f_{F^k}(x^k)^\top d_{F^k}^k + (d_{F^k}^k)^\top B_{F^k}^k d_{F^k}^k \\ &= (\nabla f_{F^k}(x^k) + B_{F^k}^k d_{F^k}^k)^\top d_{F^k}^k \leq 0. \end{aligned}$$

By Assumption 2, we have

$$(4.3) \quad \nabla f_{F^k}(x^k)^\top d_{F^k}^k \leq -c_1 \|d_{F^k}^k\|^2.$$

From the definition of d^k , we conclude that $d_i^k = 0$, if $i \in A_1^k$. For $i \in A_2^k \cup A_3^k$, we partition the choice of d_i^k into four parts.

- 1) If $\theta_i^k = 0$ or $\gamma_i^k = 0$, then $d_i^k = 0$ and $\nabla f_i(x^k) d_i^k \leq -(d_i^k)^2$.
- 2) If $\theta_i^k = 1$ or $\gamma_i^k = 1$, then $d_i^k = -\nabla f_i(x^k)$ and $\nabla f_i(x^k) d_i^k \leq -(d_i^k)^2$.
- 3) If $\theta_i^k = (x_i^k - l_i)/\nabla f_i(x^k)$ or $\gamma_i^k = (x_i^k - l_i)/\nabla f_i(x^k)$, then $d_i^k = l_i - x_i^k$, $\nabla f_i(x^k) \geq x_i^k - l_i$, so we obtain $\nabla f_i(x^k) d_i^k \leq -(d_i^k)^2$.
- 4) If $\theta_i^k = (x_i^k - u_i)/\nabla f_i(x^k)$ or $\gamma_i^k = (x_i^k - u_i)/\nabla f_i(x^k)$, then $d_i^k = u_i - x_i^k$, $\nabla f_i(x^k) \leq x_i^k - u_i$, which means $\nabla f_i(x^k) d_i^k \leq -(d_i^k)^2$.

Denote $c = \min(c_1, 1)$. It then follows that (4.2) holds. This completes the proof. \square

Lemma 4.2. *Suppose that Assumptions 1 and 2 hold, $x^k \in \Omega$ and d^k is defined by (3.5). Then x^k is a KKT point of f on Ω if and only if $d^k = 0$.*

Proof. First we suppose that $d^k = 0$.

If $i \in A^k$, then by (3.5) we have

$$P_2^k P_2^{k\top} \Theta^k \nabla f(x^k) = 0, \quad P_3^k P_3^{k\top} \Gamma^k \nabla f(x^k) = 0.$$

Since $\theta_i^k \neq 0$ for $i \in A_2^k$ and $\gamma_i^k \neq 0$ for $i \in A_3^k$, it follows that $\nabla f_i(x^k) = 0$ for $i \in A_2^k \cup A_3^k$. For $i \in A_1^k$, x^k satisfies the first case of (2.3).

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

If $i \in F^k$, we have

$$(4.4) \quad x_i^k > l_i + \min[\varrho(x^k, \lambda^k, \mu^k), \varsigma], \quad x_i^k < u_i - \min[\varrho(x^k, \lambda^k, \mu^k), \varsigma].$$

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 $l_{F^k} - x_{F^k}^k \leq \tilde{d}_{F^k} \leq u_{F^k} - x_{F^k}^k$ 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 $d_{F^k}^k = 0$ is the optimal solution of (3.3). We can prove in a similar way that $\nabla f_i(x^k)$ cannot be positive. Hence, $\nabla f_i(x^k) = 0$ for each $i \in F^k$. This proves 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 (2.3) and (3.4) we know that $A_2^k = \emptyset$, $A_3^k = \emptyset$. Therefore, $d_{A^k}^k = 0$.

On the other hand, $d = 0$ is a feasible solution of the quadratic programming problem (3.3). Since $\nabla f_{F^k}(x^k) = 0$ and $B_{F^k}^k$ is a positive definite matrix, it follows that

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

Hence, $d_{F^k}^k = 0$ is the optimal solution of the quadratic programming problem (3.3), and $d^k = (d_{A^k}^k, d_{F^k}^k) = 0$. \square

The following theorem shows the global convergence of the Algorithm AASN.

Theorem 4.3. *Suppose that Assumptions 1 and 2 hold and f is twice continuously differentiable in Ω . Assume that $x^k \rightarrow \bar{x}$ and d^k is defined by (3.5). Then \bar{x} is a KKT point of problem (1.1).*

Proof. If the sequence $\{x^k\}$ is finite with last point \bar{x} then by Lemma 4.1 and Lemma 4.2, \bar{x} is a KKT point of Problem (1.1). So supposing that the sequence is infinite, we have that

$$(4.5) \quad f(x^k + \alpha^k d^k) - f(x^k) \leq \alpha^k \nabla f(x^k)^\top d^k + o\|d^k\|^2 \leq -c\|d^k\|^2,$$

where the last inequality follows from Lemma 4.1.

Since the sequence $\{f(x^k)\}$ is decreasing and bounded from below, it follows from (4.5) that $\{d^k\} \rightarrow 0$.

Let \bar{x} be any accumulation point of $\{x^k\}$ generated by Algorithm AASN. Then there exists a subsequence $\{x^{k_i}\}$, $i = 1, 2, \dots$, such that

$$\lim_{i \rightarrow \infty} x^{k_i} = \bar{x}.$$

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

$$(4.6) \quad (l_j + u_j - 2\bar{x}_j) \nabla f_j(\bar{x}) < 0,$$

or there exists $j \in \bar{F}$ such that

$$(4.7) \quad \nabla f_j(\bar{x}) \neq 0.$$

If (4.6) holds for some $j \in \bar{A}$, then

$$(4.8) \quad j \in A_2^{k_i}.$$

Since $d^k \rightarrow 0$, this implies that $d_{A_2}^{k_i} \rightarrow 0$, and from (3.8) it follows that

$$(4.9) \quad \begin{aligned} \lim_{i \rightarrow \infty} \nabla f_j(x^{k_i}) &= 0 & \text{if } l_j \leq x_j^{k_i} - \nabla f_j(x^{k_i}) \leq u_j, \\ \lim_{i \rightarrow \infty} l_j - x_j^{k_i} &= 0 & \text{if } x_j^{k_i} - \nabla f_j(x^{k_i}) \leq l_j, \\ \lim_{i \rightarrow \infty} u_j - x_j^{k_i} &= 0 & \text{if } x_j^{k_i} - \nabla f_j(x^{k_i}) \geq u_j, \end{aligned}$$

which contradicts (4.6) for all sufficiently large i .

Now we come to prove that $\nabla f_{\bar{F}}(\bar{x}) = 0$. We recall that $d_{F^k}^k$ is the solution of the quadratic programming problem

$$\begin{aligned} \min \quad & \nabla f_{F^k}(x^k)^\top d_{F^k} + \frac{1}{2} d_{F^k}^\top B_{F^k}^k d_{F^k} \\ \text{s.t.} \quad & l_{F^k} - x_{F^k}^k \leq d_{F^k} \leq u_{F^k} - x_{F^k}^k. \end{aligned}$$

Since $d^k \rightarrow 0$, the continuity of the optimal solution of a strictly convex quadratic programming problem under perturbations (see Proposition 6.1 of [25]) implies that zero is the optimal solution of

$$\begin{aligned} \min \quad & \nabla f_{\bar{F}}(\bar{x})^\top d_{\bar{F}} + \frac{1}{2} d_{\bar{F}}^\top B_{\bar{F}} d_{\bar{F}} \\ \text{s.t.} \quad & l_{\bar{F}} - \bar{x}_{\bar{F}} \leq d_{\bar{F}} \leq u_{\bar{F}} - \bar{x}_{\bar{F}}. \end{aligned}$$

Hence, $\nabla f_{\bar{F}}(\bar{x}) = 0$ by a reasoning similar to that used in the proof of Lemma 4.2. \square

5. THE LOCAL SUPERLINEAR CONVERGENCE ANALYSIS

In this section we study the local behavior of AASN. The following assumption is essential for the local fast convergence analysis.

Assumption 3 (SSOSC). Let $(\bar{x}, \bar{\lambda}, \bar{\mu})$ be a KKT triplet for problem (1.1). We say that SSOSC holds at \bar{x} if

$$(5.1) \quad z^\top \nabla^2 f(\bar{x}) z > 0, \quad \forall z \in \{z \in \mathbb{R}^n : z_i = 0, i \in A_+\},$$

where A_+ is the strong active set, i.e. the index set of active constraints with positive multipliers.

Lemma 5.1. *Suppose that \bar{x} is a KKT point of problem (1.1) and Assumption 3 is satisfied. Then there exists a neighborhood Ω_1 of \bar{x} such that, for each x in this neighborhood Ω_1 ,*

$$(5.2) \quad \begin{aligned} L(x) &= \{i : l_i = \bar{x}_i\}, \\ U(x) &= \{i : \bar{x}_i = u_i\}, \\ F(x) &= \{i : \bar{\lambda}_i = 0 \text{ and } \bar{\mu}_i = 0\}. \end{aligned}$$

Lemma 5.1 shows that when (x, λ, μ) is sufficiently close to $(\bar{x}, \bar{\lambda}, \bar{\mu})$, the estimate of the active set is accurate, see Theorem 2.2 in [9].

Theorem 5.2. *Suppose that Assumptions 1 and 2 hold, f is twice continuously differentiable with Lipschitz continuous Hessian matrices and $(\bar{x}, \bar{\lambda}, \bar{\mu})$ is a KKT pair for problem (1.1) which satisfies Assumption 3. Then the sequence $\{x^k\}$ generated by AASN converges to \bar{x} , and the rate of convergence is superlinear.*

Proof. We notice that in the bound constrained optimization problem, the gradients of active constraints are linearly independent at \bar{x} , that is, linear independence assumption is always satisfied. As $(\bar{x}, \bar{\lambda}, \bar{\mu})$ is a KKT pair for problem (1.1) that satisfies Assumption 3, it follows that $(\bar{x}, \bar{\lambda}, \bar{\mu})$ is an isolate KKT point.

From Lemma 5.1 we know that for every $x^k \in \Omega_1$ the accurate active set is identified. That is, none of the bounds in L^k and U^k is eventually active at the solution and $x_{L^k}^k = \bar{x}_{L^k}$, $x_{U^k}^k = \bar{x}_{U^k}$.

Hence, for $x^k \in \Omega_1$, the quadratic programming problem can be written as

$$(5.3) \quad \begin{aligned} \min \tilde{m}(d_{F^k}) &= \nabla f_{\bar{F}}(\bar{x}_{\bar{L}}, x_{\bar{F}}^k, \bar{x}_{\bar{U}})^\top d_{\bar{F}} + \frac{1}{2} d_{\bar{F}}^\top B_{\bar{F}}^k d_{\bar{F}} \\ \text{s.t. } l_{\bar{F}} - x_{\bar{F}}^k &\leq d_{\bar{F}} \leq u_{\bar{F}} - x_{\bar{F}}^k. \end{aligned}$$

Since none of the bounds in problem (5.3) is eventually active at the solution, it follows that

$$d_{\bar{F}}^k = -B_{\bar{F}}^{k-1} \nabla f_{\bar{F}}(\bar{x}_{\bar{L}}, x_{\bar{F}}^k, \bar{x}_{\bar{U}})$$

and $d_{\bar{F}}^k$ is just the Newton direction for solving the system of equations

$$\nabla f_{\bar{F}}(\bar{x}_{\bar{L}}, x_{\bar{F}}^k, \bar{x}_{\bar{U}}) = 0.$$

Hence, from classical results on Newton methods, we have that

$$\lim_{k \rightarrow \infty} \frac{\|x^k + d^k - \bar{x}\|}{\|x^k - \bar{x}\|} = \lim_{k \rightarrow \infty} \frac{\|x_{\bar{F}}^k + d_{\bar{F}}^k - \bar{x}_{\bar{F}}\|}{\|x^k - \bar{x}\|} = 0.$$

Then by using Taylor's expansions of $f(x^k + d^k)$ and $f(x^k)$ at \bar{x} , similarly to Remark 3.1 in [8], we finally obtain that

$$f(x^k + d^k) - f(x^k) - \frac{1}{2} \nabla f(x^k)^\top d^k \leq o(\|d^k\|^2).$$

Therefore, the assertion on the acceptance of the unit step size follows from the fact that $\sigma < \frac{1}{2}$. □

6. NUMERICAL TESTS

Now, we will deal with numerical tests of AASN. In AASN, we choose $\sigma = 10^{-1}$ in all runs. The code is written in MATLAB with double precision. For each problem, the termination condition is the Euclidean norm of the search direction below 10^{-5} , namely, $\|d^k\| \leq 10^{-5}$. We have also included additional two stopping flags, that is, the maximum iteration $IT_{\max} = 1000$ in the main loop and the maximum iteration number $IT_{\text{inner}} = 25$ in the line search loop. For each test function, we use the same initial value x^0 .

We employ the technique from [11] for generating bound constrained optimization problems with known characteristics. Through this kind of strategy, the number and position of the active constraints, the Lagrange multipliers, and the shape of the feasible region can be easily controlled, see [11] for more details.

The accurate active set identification function employed in AASN can also be defined by

$$(6.1) \quad \varrho(x, \lambda, \mu) = \begin{cases} 0 & \text{if } r(x, \lambda, \mu) = 0, \\ \frac{1}{\log(r(x, \lambda, \mu))} & \text{if } r(x, \lambda, \mu) \in (0, 0.9), \\ \frac{-1}{\log(0.9)} & \text{if } r(x, \lambda, \mu) \geq 0.9, \end{cases}$$

where

$$\begin{aligned} r(x, \lambda, \mu) = & \|\nabla f(x) - \lambda + \mu\| + |\lambda^\top e| + |\mu^\top e| + \|[-\lambda]_+\| \\ & + \|[-\mu]_+\| + \|[l - x]_+\| + \|[u - x]_+\|. \end{aligned}$$

We write $[x]_+$ for the vector $\max\{0, x\}$, where the maximum is taken componentwise, and $e^\top = (1, 1, \dots, 1)$, $e \in \mathbb{R}^n$.

First, we compare AASN with two different accurate identification functions (6.1) and (3.2) with the test problems chosen from [3] and [27]. The dimensions of the test problems range from 5000 to 20000. Displayed in Table 1 are the CPU times to obtain the solutions through AASN with (6.1) and (3.2), which are denoted by CPU_r and CPU_p , respectively. In addition, we also list the iteration number IT_r and IT_p as well as the number of the function evaluation IF_r and IF_p for AASN with (6.1) and (3.2).

Numerical results demonstrate that both the identification functions work well. Now we list other two different active set identification functions.

| No. | n | CPU_r | CPU_p | IF/IT_r | IF/IT_p | FF_r | FF_p |
|-----|-------|----------------|----------------|------------------|------------------|---------------|---------------|
| P1 | 10000 | 5.8750 | 5.8280 | 20/10 | 20/10 | -2.6442e+004 | -2.6442e+004 |
| P2 | 20000 | 192.5620 | 175.9210 | 275/64 | 249/59 | 2.1777e-013 | 1.2333e-013 |
| P3 | 10000 | 24.6870 | 21.7350 | 312/67 | 280/70 | 1.2814e-013 | 7.7706e-014 |
| P4 | 10000 | 8.3590 | 9.3280 | 106/29 | 122/31 | 7.8455e-007 | 4.0860e-007 |
| P5 | 20000 | 47.0000 | 44.2650 | 121/30 | 101/25 | 6.7026e-017 | 6.4218e-016 |
| P6 | 5000 | 3.7970 | 3.8900 | 250/53 | 248/52 | 5.4971e-018 | 1.9420e-021 |
| P7 | 10000 | 23.7650 | 19.1720 | 337/67 | 258/55 | 4.6383e-017 | 1.3018e-016 |
| P8 | 5000 | 15.7660 | 13.5630 | 997/182 | 887/164 | 1.6490e-014 | 2.0461e-016 |
| P9 | 10000 | 54.7030 | 75.6410 | 934/161 | 1264/215 | 1.0365e-014 | 4.1858e-013 |
| P10 | 5000 | 8.5780 | 8.7650 | 517/86 | 517/86 | 6.9789e-014 | 2.1284e-017 |

Table 1. Comparison of AASN with (6.1) and (3.2).

In ALBFGS [30],

$$(6.2) \quad \begin{aligned} L(x) &= \{i: x_i \leq l_i + a_i(x)\nabla f_i(x)\}, \\ U(x) &= \{i: x_i \geq u_i + b_i(x)\nabla f_i(x)\}, \\ F(x) &= \{1, \dots, n\} \setminus (L(x) \cup U(x)), \end{aligned}$$

where $a_i(x)$ and $b_i(x)$ are nonnegative continuous functions such that if $x_i = l_i$ or $x_i = u_i$ then $a_i(x) > 0$ or $b_i(x) > 0$, respectively.

In ASNA [10],

$$(6.3) \quad \begin{aligned} L(x) &= \left\{i: x \leq l_i + \min \left[\zeta c(x) a_i(x) \lambda_i(x), \frac{u_i - l_i}{3} \right] \right\}, \\ U(x) &= \left\{i: x \geq u_i - \min \left[\zeta c(x) b_i(x) \mu_i(x), \frac{u_i - l_i}{3} \right] \right\}, \\ F(x) &= \{1, \dots, n\} \setminus (L(x) \cup U(x)), \end{aligned}$$

where ζ is a positive constant; $a(x)$, $b(x)$, and $c(x)$ are the functions,

$$a(x) = \alpha - l + x, \quad b(x) = \beta + u - x, \quad c(x) = f(x^0) - f(x).$$

We remark that α and β are arbitrarily fixed before starting the algorithm; $\lambda(x)$ and $\mu(x)$ are the multiplier functions [10]. The identification functions (6.2) and (6.3) belong to the approximation identification function.

Secondly, test results on 10 problems are listed. Different identification functions (6.2), (6.3) will be employed in AASN, the subscripts m and f denote the test results of AASN with (6.3) and (6.4) respectively in Tab. 2.

| No. | CPU _m | CPU _f | IF/IT _m | IF/IT _f | FF _m | FF _f |
|-----|------------------|------------------|--------------------|--------------------|-----------------|-----------------|
| P1 | 12.8750 | 10.0630 | 35/11 | 30/10 | -2.6442e+004 | -2.6442e+004 |
| P2 | 125.5630 | 120.8120 | 179/46 | 182/47 | 8.0222e-016 | 2.1777e-013 |
| P3 | 30.4060 | 239.2500 | 184/48 | 1021/193 | 4.4443e-014 | 1.4659e-013 |
| P4 | 14.5000 | 9.5780 | 169/39 | 155/36 | 2.3096e-015 | 4.9272e-016 |
| P5 | 106.3280 | 100.2810 | 233/53 | 217/49 | 9.6295e-016 | 9.2125e-016 |
| P6 | 4.2660 | 4.2350 | 279/56 | 285/60 | 5.0090e-013 | 2.8948e-021 |
| P7 | 17.7820 | 16.7970 | 255/55 | 257/54 | 3.0064e-016 | 4.6730e-019 |
| P8 | 18.4380 | 31.4520 | 1236/201 | 2156/300 | 1.9870e-018 | 5.6397e-013 |
| P9 | 69.3290 | 62.3370 | 1142/206 | 1041/203 | 1.6346e-021 | 1.7052e-012 |
| P10 | 35.1560 | 25.3960 | 2059/250 | 1660/236 | 1.7745e-017 | 6.9989e-014 |

Table 2. Numerical results of AASN with (6.2) and (6.3).

Numerical tests show that (3.1), (6.1), (6.2), and (6.3) are indeed “good” estimates of \bar{L} , \bar{F} . We can see that the accurate identification technique is efficient. It should be noted that the superlinear convergence of ASNA occurs under the assumption of strict complementarity, while the same results of AASN can be obtained without this assumption.

In ASNA, the search direction $d_{A^k}^k$ of the active variables is defined by

$$(6.4) \quad d_i^k = \begin{cases} x_i^k - l_i & \text{if } i \in L^k, \\ x_i^k - u_i & \text{if } i \in U^k. \end{cases}$$

However, we found that this simple modification can, in some cases, increase IF and IT, especially IF. This indicates that this form of $d_{A^k}^k$ in (6.4) cannot provide sufficient decrease as compared with $d_{A^k}^k$ in (3.8).

7. CONCLUSIONS

An accurate active set Newton method is analyzed in this paper. The active set strategy which belongs to the accurate active set identification allows quick change in the working set. It is suitable for solving large scale problems. We divide the active variables into two parts, the one that satisfies the KKT conditions and the other that violates the KKT conditions. This kind of division enables us to obtain the global convergent property of our algorithm. Numerical results show that AASN is practical and efficient.

We know that Support Vector Machine (SVM) training may be posed as a large quadratic program with bound constraints and a single linear equality constraint. The active set strategies discussed in this paper are closely related to decomposition methods currently popular for SVM training [14], [18]. Consequently, how to extend the active set strategies to SVM remains to be considered in the future.

Acknowledgement. We would like to thank Professor Y. L. Lai for his considerable help in the preparation of this paper, and thank the anonymous referees for their helpful comments.

References

- [1] *T. F. Coleman, Y. Li*: On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds. *Math. Program.* *67* (1994), 189–224.
- [2] *T. F. Coleman, Y. Li*: An interior trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.* *6* (1996), 418–445.
- [3] *A. R. Conn, N. I. M. Gould, Ph. L. Toint*: Testing a class of methods for solving minimization problems with simple bounds on the variables. *ACM Trans. Math. Softw.* *7* (1981), 17–41.
- [4] *A. R. Conn, N. I. M. Gould, Ph. L. Toint*: Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM J. Numer. Anal.* *25* (1988), 433–460.
- [5] *A. R. Conn, N. I. M. Gould, Ph. L. Toint*: Correction to the paper on global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM J. Numer. Anal.* *26* (1989), 764–767.
- [6] *J. E. Dennis, J. J. Moré*: A characterization of superlinear convergence and its application to quasi-Newton methods. *Math. Comput.* *28* (1974), 549–560.
- [7] *Z. Dostál*: A proportioning based algorithm with rate of convergence for bound constrained quadratic programming. *Numer. Algorithms* *34* (2003), 293–302.
- [8] *F. Facchinei*: Minimization of SC^1 functions and the Maratos effect. *Oper. Res. Lett.* *17* (1995), 131–137.
- [9] *F. Facchinei, A. Fischer, C. Kanzow*: On the accurate identification of active constraints. *SIAM J. Optim.* *9* (1998), 14–32.
- [10] *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.
- [11] *F. Facchinei, J. Júdice, J. Soares*: Generating box-constrained optimization problems. *ACM Trans. Math. Softw.* *23* (1997), 443–447.
- [12] *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.
- [13] *F. Facchinei, S. Lucidi, L. Palagi*: A truncated Newton algorithm for large scale box constrained optimization. *SIAM J. Optim.* *12* (2002), 1100–1125.
- [14] *R. E. Fan, P. H. Chen, C. J. Lin*: Working set selection using second order information for training support vector machines. *J. Mach. Learn. Res.* *6* (2005), 1889–1918.
- [15] *P. E. Gill, W. Murray, M. H. Wright*: *Practical Optimization*. Academic Press, London, 1981.
- [16] *W. W. Hager, H. Zhang*: A new active set algorithm for box constrained optimization. *SIAM J. Optim.* *17* (2006), 526–557.
- [17] *M. Heinkenschloss, M. Ulbrich, S. Ulbrich*: Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption. *Math. Program.* *86* (1999), 615–635.
- [18] *S. S. Keerthi, E. G. Gilbert*: Convergence of a generalized SMO algorithm for SVM classifier design. *Mach. Learn.* *46* (2002), 351–360.
- [19] *M. Lescrenier*: Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold. *SIAM J. Numer. Anal.* *28* (1991), 476–495.

- [20] *C. J. Lin, J. J. Moré*: Newton's method for large bound-constrained optimization problems. *SIAM J. Optim.* *9* (1999), 1100–1127.
- [21] *J. J. Moré, G. Toraldo*: On the solution of large quadratic programming problems with bound constraints. *SIAM J. Optim.* *1* (1991), 93–113.
- [22] *Q. Ni, Y. Yuan*: A subspace limited memory quasi-Newton algorithm for large-scale nonlinear bound constrained optimization. *Math. Comput.* *66* (1997), 1509–1520.
- [23] *J. Nocedal, S. J. Wright*: *Numerical Optimization*. Springer, New York, 2006.
- [24] *C. Oberlin, S. J. Wright*: Active set identification in nonlinear programming. *SIAM J. Optim.* *17* (2006), 577–605.
- [25] *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.
- [26] *B. T. Polyak*: The conjugate gradient method in extremal problems. *U.S.S.R. Comput. Math. Math. Phys.* *9* (1969), 94–112.
- [27] *K. Schittkowski*: *More Test Examples for Nonlinear Programming Codes*. Lecture Notes in Economics and Mathematical Systems, Vol. 282. Springer, Berlin, 1987.
- [28] *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.
- [29] *L. Sun, L. Fang, G. P. He*: An active set strategy based on the multiplier function or the gradient. *Appl. Math.* *55* (2010), 291–304.
- [30] *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.

Authors' addresses: *L. Sun* (corresponding author), College of Information Sciences and Engineering, Shandong Agricultural University, 271018 Ta'ian, P. R. China, and Department of Mathematics, Shanghai Jiaotong University, No. 800 Dong Chuan Road, 200240 Shanghai, P. R. China, e-mail: sunlishi@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; *Y. Wang*, College of Information Science and Engineering, Shandong University of Science and Technology, 266510 Qingdao, P. R. China, e-mail: wylb@263.net; *C. Zhou*, College of Information Science and Engineering, Shandong University of Science and Technology, 266510 Qingdao, P. R. China, e-mail: zhoucy@sdust.edu.cn.