A New Hyperplane Method for Solving Linear Inequality Constrained Optimization

Yueting Yang*, Yunlong Lu, Xiaowei Jiang

Department of Mathematics, Beihua University, Jilin 132013, China

Abstract

For solving linear inequality constrained optimization, a new hyperplane method is introduced. The trust region approach for unconstraint optimization is used to minimize objective function on the hyperplane defined by the current iterative point. If a separate indicator shows that it is not worthwhile to find a better point in the current hyperplane, then a line search along a chopped direction will be used. The proposed method can overcome the zigzagging phenomena. Global convergence of the algorithm is proved. The numerical results are presented to show the property and efficiency of the proposed method.

Keywords: Linear Inequality Constraints Optimization; Trust Region Method; Global Convergence

1 Introduction

We consider the following linear inequality constraints optimization

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad c_i(x) \triangleq a_i^T x - b_i \leq 0, \quad i = 1, \ldots, m.
\end{align*}
\]  

(1)

where \( f(x) : \mathbb{R}^n \to \mathbb{R} \) is twice continuously differentiable, \( a_i \in \mathbb{R}^n \) \( (i = 1, \cdots, m) \), \( b = (b_1, \cdots, b_m)^T \in \mathbb{R}^m \). Let \( L = \{1, 2, \cdots, m\} \), \( I(x) = \{i \in L : c_i(x) = 0\} \) and \( A = (a_1, \cdots, a_m) \). \( D = \{x \in \mathbb{R}^n : A^T x - b \leq 0\} \) denotes the feasible set of Problem 1.

Many practical and interesting problems can be expressed as (1). There are kinds of methods for solving (1) [1]-[5], in which the projected gradient and the active set methods are used widely. The former is originally developed by Rosen (1960) [1]. The latter attempts to locate the solution of an equality constraints subproblem to find the search direction. In this paper, we propose a new hyperplane method for solving (1). Namely, at the current iterative point \( x^k \in D \), a quadratic model of \( f \) is constructed by using \( g_k = \nabla f(x^k) \) and \( B^k \), they are the gradient and an approximation Hessian matrix of \( f(x) \) at \( x^k \), respectively. Then the trust region method is applied to finding the minimal point of \( f(x) \) in the hyperplane defined by \( x^k \). Till a separate indicator

*Project supported by the National Nature Science Foundation of China (No. 20070421125).
*Corresponding author.

Email address: yyt2858@163.com (Yueting Yang).
says that it is not worthwhile anymore, the current hyperplane is abandoned along a direction with line search, then some old active constraints become non-active constraints and some new constraints become active constraints. In the process of minimizing within a hyperplane, if the algorithm hits a bound, several new constraints are incorporated.

The paper is organized as follows. In the next section we derive a sufficient and necessary condition for judging the K-T points. The new algorithm and its global convergence are proposed in Section 3. The efficiency of the method is tested via numerical results in Section 4.

## 2 Optimal Conditions

Throughout this paper, the following two basic assumptions are used

**H1** The feasible set $D$ is nonempty, i.e. $D \neq \emptyset$.

**H2** For arbitrary $x \in D$, the vectors $\{a_i, i \in I(x)\}$ are linearly independent.

For iteration point $x^k$, we denote $I_k = I(x^k)$ and $A_k = (a_j, j \in I_k) \in R^{n \times |I_k|}$. Without lose of generality, we assume $I_k$ is nonempty. (Otherwise, $x^k$ is the internal point of feasible set, the algorithm for unconstraint optimization can be used to find $x^{k+1}$.) From the assumption **H2**, $A_k^T A_k$ is positive definite. Define $\bar{F}_{I_k}$ as a hyperplane of the feasible region, that is

$$F_{I_k} = \{x \in D : c_j(x) = 0, \ j \in I_k; \ c_j(x) < 0, \ j \in L \setminus I_k\}.$$  

The closure $\bar{F}_{I_k}$ is denoted by $\bar{F}_{I_k}$. Let $[F_{I_k}]$ be the smallest linear manifold that contains $F_{I_k}$, and $S_{I_k}$ be the subspace obtained by the parallel translation of $[F_{I_k}]$. We define the below matrices

$$Q_k = (A_k^T A_k)^{-1} A_k^T, \quad P_k = I - A_k Q_k,$$  

where $I$ is a $n$ order identity matrix. $Q_k$ and $P_k$ are called chopped and project matrices respectively. Applying $Q_k$ and $P_k$, two feasible descent directions of $f(x)$ at $x^k$ will be constructed in the below. It is easy to derive that

$$Q_k A_k = I, \quad P_k^2 \equiv P_k, \quad P_k A_k = 0, \quad P_k^T = P_k.$$  

For expression convenient, we also define two transition vectors

$$q_k = (q^j_k, j \in I_k)^T = -Q_k g_k, \quad v_k = (v^j_k, j \in I_k)^T, \quad v_k^j = \begin{cases} 0 & q^j_k \geq 0 \\ q^j_k & q^j_k < 0 \end{cases}.$$  

**Definition 1** Let

$$g_l(x^k) = -P_k g_k, \quad g_c(x^k) = Q_k^T v_k, \quad g_p(x^k) = g_l(x^k) + g_c(x^k).$$  

$g_l(x^k)$ is called the inner anti-gradient of $-g_k$ on $S_{I_k}$. $g_c(x^k)$ is called the chopped vector. And $g_p(x^k)$ is called the projection vector of $-g_k$ in the feasible set for the Problem 1.

**Lemma 1** 1) If $g_l(x^k) \neq 0$, then $g_l(x^k)$ is a feasible descent direction of the Problem 1;

2) If $g_c(x^k) \neq 0$, then $g_c(x^k)$ is a feasible descent direction of the Problem 1.
Lemma 2 Let $x^k$ be a feasible point of the Problem 1 and $g_I(x^k) = 0$. If $q_k \geq 0$. Then $x^k$ is the K-T point of the Problem 1.

The Lemma 2 is just a sufficient condition that $x^k$ is the K-T point of the Problem 1. In the following, we derive the sufficient and necessary condition.

Theorem 1 $x^k$ is the K-T point of the Problem 1 if and only if $g_p(x^k) = 0$.

3 Algorithm and Its Convergence

According to the Theorem 1, we propose an efficient algorithm for solving the Problem 1. When the line search is applied along the chopped vector $g_c(x^k)$, the Wolfe-Powell conditions are chosen to compute step length, that is,

\[
 f(x^k + \alpha d_k) \leq f_k + \beta_1 \alpha d_k^T g_k, \\
 |g(x^k + \alpha d_k)^T d_k| \leq -\beta_2 g_k^T d_k, 
\]

where $0 < \beta_1 < 1/2 < \beta_2 < 1$.

Algorithm 1

Choose an initial point $x^0 \in D$, $\eta \in (0, 1)$, $0 < \beta_1 < \beta_2 < 1$, set $k := 1$.

Step 1 If $\|g_p(x^k)\| = 0$, then $x^k$ is a K-T point of the problem 1, stop; if $\|g_c(x^k)\| \|g_p(x^k)\| \geq \eta$, goto Step 2 and compute $x^{k+1}$, else use the sub-algorithm 2 to compute $x^{k+1}$.

Step 2 Compute the step length by line search along $g_c(x^k)$. Denote that

\[
 \bar{\alpha} = \min\{\frac{b_i - a_i^T x_k}{a_i^T g_c(x_k)}, a_i^T g_c(x_k) > 0, i \in L \setminus I_k\}.
\]

Set $\alpha = 1$ as the initial step length. If $\alpha$ satisfies (7) and (8), set $\alpha_k = \min\{\alpha, \bar{\alpha}\}$ and $x^{k+1} = x^k + \alpha_k g_c(x^k)$. Choose a new value by the line search for $\alpha$ and repeat test (7) and (8).

Lemma 3 Let

\[
 K = \{k \in N \mid \frac{\|g_c(x^k)\|}{\|g_p(x^k)\|} \geq \eta\}. 
\]

Denote $\alpha_* = \inf\{\alpha_k, k \in K\}$. Assume that $\{x^k, k \in K\}$ is the sequence of points generated by the Algorithm 1. If the infinite sequence $\{x^k, k \in K\}$ such that $x^k \to x^*$ and $x^*$ is not the K-T point of the Problem 1. Then

\[
 g(x^*)^T g_c(x^*) < 0, \quad a_j^T g_c(x^*) \leq 0 \\
 \alpha_* > 0, \quad k \in K. 
\]
If the condition (9) is violated, a trust region sub-algorithm on the given face $F_k$ will be used. Many approaches of the sub-algorithm maybe chosen. In general, the sub-algorithm should have following properties:

1. **P1** If $x^k \in F_k$, then $x^{k+1} \in \bar{F}_I$;
2. **P2** $f(x^{k+1}) < f(x^k)$;
3. **P3** If $\{x^k, x^{k+1}, x^{k+2} \ldots \} \subset F_k$ is a set of infinitely many iterates generated by the sub-algorithm, then $g_I(x^k) \to 0$.

To sum up, if properties P1, P2, P3 are satisfied by the sub-algorithm, the main Algorithm 1 is well defined. The below sub-algorithm (Algorithm 2) sketches a trust region algorithm. To adapt large scale optimization, the conjugate gradient (CG) method is applied to the quadratic subproblem on the current face $F_k$.

$$ \min q(\delta) = \frac{1}{2} \delta^T B_k \delta + g_k^T \delta $$

subject to $\|\delta\| \leq \rho_k$. (13)

where $\rho_k$ is trust region radius, $B_k$ is the approximation Hessain matrix of $f(x)$ at $x^k$. $B_k$ will be updated by the modified quasi-Newton BFGS formula which is investigated in [7].

**Algorithm 2 (Trust region sub-algorithm)**

**Step 1** Starting with $\delta^{(0)} = 0$, apply the CG method to (13). This method generates a search direction $d^{(j)}$ and a new iterate $\delta^{(j+1)} = \delta^{(j)} + d^{(j)}$ at the j-th iteration. This process is interrupted in any of below three cases, and a solution of (13) is obtained denote by $\delta_{\text{trial}}$.

1. $\nabla q(\delta^{(j)}) = 0$, $\|\delta\| \leq \rho_k$, then $\delta_{\text{trial}} = \delta^{(j)}$;
2. $d^{(j)}$ is a negative curvature direction. Set $\delta_{\text{trial}} = \delta^{(j)} + \alpha_j d^{(j)}$, where $\alpha_j$ satisfies $\|\delta^{(j)} + \alpha_j d^{(j)}\| = \rho_k$.
3. $\|\delta^{(j+1)}\| > \rho_k$ or $x^k + \delta^{(j+1)} \notin F_k$, in this case $\delta_{\text{trial}}$ is defined by the following mode:

Define $\delta'$ is the projection of $\delta^{(j+1)}$ in the region $\{\delta : \|\delta\| \leq \rho_k, x^k + \delta \in F_k\}$,

Define $\delta'' = \delta^{(j)} + \alpha_j d^{(j)}$, where $\alpha_j$ is the largest step such that $\|\delta''\| \leq \rho_k$ and $x^k + \delta'' \in F_k$ all holds.

If $q(\delta') \leq q(\delta'')$, then $\delta_{\text{trial}} = \delta'$, else $\delta_{\text{trial}} = \delta''$.

**Step 2** Set $x_{\text{trial}} = x^k + \delta_{\text{trial}}$, and calculate $\Delta_{\text{pred}} = -q(\delta_{\text{trial}})$, $\Delta_{\text{act}} = f(x^k) - f(x_{\text{trial}})$.

If $\Delta_{\text{act}} < \gamma \Delta_{\text{pred}}$, set $\rho_k = \mu \|\delta_{\text{trial}}\|$ and goto Step 1; If $\gamma \Delta_{\text{pred}} \leq \Delta_{\text{act}} \leq \Delta_{\text{pred}}$, set $x^{k+1} = x_{\text{trial}}$, $\rho_{k+1} = \rho_k$; If $\Delta_{\text{act}} > \Delta_{\text{pred}}$, set $x^{k+1} = x_{\text{trial}}$, $\rho_{k+1} = \phi \rho_k$.

In the following, the global convergence theorems are presented and proved.

**Theorem** 2. *The Algorithm 1 either terminate to a K-T point $x^k$ in limit steps or any limit point of sequence $\{x^k\}$ is the K-T point of the Problem 1.*

**Proof** Note that the process of Algorithm 1 and the Theorem 1. We assume, without loss of generality, that the infinite sequence $\{x^k\}$ generated by Algorithm 1 satisfies $x^k \to x^*$. From the property P2, (7), the Lemma 1 and the Theorem 1 we know, $\{f(x^k)\}$ is monotone decreasing and $f(x^k) \to f(x^*)$, $k \to \infty$. (14)
Below two cases may occur:

Case 1. For the set \( K \), there are infinite elements such that \( k \in K, k \to \infty, x^k \to x^* \); Assume, on the contrary, that \( x^* \) is not the K-T point of the Problem 1. Then for sufficiently large \( k \),

\[
f(x^{k+1}) - f(x^k) \leq \beta_1 \alpha_k g_k^T g_c(x^k) < \frac{1}{2} \beta_1 \alpha_k (x^*)^T g_c(x^*) < 0.
\] (15)

by using (7), (11) and (12). But \( \lim_{k \to \infty} (f(x^{k+1}) - f(x^k)) = 0 \). This is a contradiction. Hence \( x^* \) is the K-T point of the Problem 1.

Case 2. \( K \) is a finite set. In this case, there are \( k_0 \in N \) and a face \( F_{l_k} \) such that \( x^k \in F_{l_k} \) for all \( k \geq k_0 \). This means \( \{x^k, x^{k+1}, x^{k+2}, \ldots \} \subset F_{l_k} \) is an infinite sequence generated by the algorithm 2 for all \( k \geq k_0 \). Then from property P3, \( \lim_{k \to \infty} \|g_1(x^k)\| = 0 \) and (9) is violated for all \( k \geq k_0 \).

Namely, \( (1 - \eta)\|g_p(x^k)\| < \|g_1(x^k)\| \). Consequently, \( \lim_{k \to \infty} \|g_p(x^k)\| = 0 \). Thus \( x^* \) is the K-T point of (1) from the Theorem 1.

Suppose the assumption H2 holds, if \( x^* \) is a K-T point and there exists \( i \in I(x^*) \) at least, such that \( a_i^T x^* - b_i = 0 \) and \( q_i^* = 0 \). We say \( x^* \) is degenerate.

**Theorem 3** Assume K-T points of the Problem 1 are non-degenerate. Then there exists \( \hat{k} \in N \) such that \( x^k \in F_{l_k} \) for all \( k \geq \hat{k} \). Moreover, all the limit points of the sequence \( \{x^k\} \) belong to \( F_{l_k} \) and are K-T points.

### 4 Numerical Results

In the section, we devote to the numerical experiment to test the proposed method. Our experiments are implemented on a PC with 1.8 MHz Pentium IV and 1 G SDRAM using MATLAB 6.5. The test aims to assess the characteristic and efficiency of the algorithm 1 by tracking the iterative process. In the below numerical results, the data \( k, n_f, t, f_*, x_* \) stand for iterative number, function and gradient evaluations, CPU times(\text{"s\}), the computed optimal function value; the optimal solution, respectively. In addition, \( F_{l_k} \) stands for the hyperplane defined by \( x_* \) (the active constraint at \( x_* \)).

**Problem 1** HS118.test ([8])  
Initial point: \( x^0 = (20, 55, 15, 20, 60, 20, 20, 60, 20, 20, 60, 20, 20, 60, 20)^T \);

Optimal solution: \( x^* = (8, 49, 3, 1, 56, 0, 1, 63, 6, 3, 70, 12, 5, 77, 18)^T \);

Optimal value: \( f^* = 664.8204500 \).

Starting at \( x^0 \), the numerical results of the proposed algorithm are listed in the following.

| Table 1: The numerical results of the Problem 1 |
|-----------------|-----------------|
| \( k/n_f/t/f_/\|\|g_p(x_*)\|\) | 18/45/0.2810/664.820500/1.7370 × 10^{-14} |
| \( x_* \) | (8, 49, 3, 1, 56, 0, 1, 63, 3, 70, 12, 5, 77, 18)^T |
| \( F_{l_k} \) | (1, 10, 12, 14, 16, 20, 22, 24, 25, 27, 28, 29, 30, 34, 40) |

We label 59 constraints of the Problem 1 as serial number. These serial numbers are only used to show clearly the variation of the corresponding hyperplane at each iterative point.
It can be observed by tracking the iterative process that the chopped vector $g_c$ is only used at the 17-th iterative. Namely, the line search along direction $g_c(x^{16})$ is implemented at iterative point $x^{16}$. And a hyperplane $(4): x_7 - x_4 + 7 \leq 13$ is abandoned. The other iterative implement the trust region sub-algorithm. Moreover, all iterative points is on the bound of feasible region. At the process of implementing the trust region sub-algorithm, when the algorithm hits a bound, several new constraints are incorporated. For instance, two constraints $(4, 20)$ are incorporated at the 7-th iterative, which implies the superiority to normal active set method.

**Problem 2** HS110.test ([8])
Initial point: $x^0 = (9, 9, 9, 9, 9, 9, 9, 9)^T$;
Optimal solution: $x^* = (9.35025655, \cdots, 9.35025655)$;
Optimal value: $f^* = -45.77846971$.

The numerical results of the problem 2 is given in the following.

Table 2: The numerical results of the Problem 2

| $k$/n | $f$/t | $f_*$/$\|g_p(x_*)\|$ | $21/40/0.2400/ - 45.77846970744630/9.3429 \times 10^{-9}$ |
|-------|-------|-------------------------|
| $x_*$ | $x^{21}$ | $(9.35026583, \cdots, 9.35026583)^T$ |
| $F_{Ie}$ | | $\langle \rangle$ |

For solving the Problem 2, the line search along the chopped vector is not implemented. Throughout the iterative process, the trust region sub-algorithm for solving unconstrained optimization is completed in the interior of feasible set.

**Problem 3** liswet1.mod ([9])
Initial point: $x^0 = (0, 0, \cdots, 0)^T$.

This is a variable dimension problem. We will only study the problem with 10 dimension. The following table lists the preliminary numerical results, in which $dim = 10$ means the dimension of the problem is 10.

Table 3: The numerical results of the Problem 3 ($dim = 10$)

| $k$/n | $f$/t | $f_*$/$\|g_p(x_*)\|$ | $16/36/0.1700/8.94609704490 \times 10^{-3}/6.8575 \times 10^{-9}$ |
|-------|-------|-------------------------|
| $x_*$ | | $(0.084147, 0.36479, 0.46046, 0.55614, 0.65181, 0.74749, 0.84316, 0.93883, 0.98402, 0.9456)^T$ |
| $F_{Ie}$ | | $\langle 2, 3, 4, 5, 6 \rangle$ |

**References**


