An improved gradient projection method for constrained optimization problems[1]

Zhijun Luo[1], Lirong Wang[2]

[1] The Department of Mathematics and Econometrics,
Hunan University of Humanities, Science and Technology, Loudi, 417000,P.R.China

[2] The Department of Information science and Engineering,
Hunan University of Humanities, Science and Technology, Loudi, 417000,P.R.China

Abstract. Combining the properties of the gradient projection methods and the system of linear equations(SSLE) algorithm, an improved gradient projection algorithm for general constrained optimization problems is proposed. At each iterate, the projection matrix only needs to be computed one time comparison with Rosen's method, then, a correction direction is yielded by solving systems of linear equations, thus the amount of computation is lower. Under some mild conditions, the algorithm is proved to possess global convergence and superlinear convergence. Finally, some numerical results are reported to show the effectiveness of the proposed algorithm.

Key words. Constrained optimization problems, Gradient projection algorithm, System of linear equations algorithm, Global convergence, Superlinear convergence rate

1. Introduction

In this paper, we consider the following constrained optimization problems,

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_j(x) \leq 0, j \in L_1 = \{1, 2, \cdots, m\}, \\ & g_j(x) = 0, j \in L_2 = \{m+1, \cdots, m+l\}. \end{array} \tag{1.1}$$

Where, $f(x),\ g_j(x)(j \in L = L_1 \bigcup L_2)$ are continuously differentiable functions. Denote the feasible set $X = \{g_j(x) \leq 0,\ j \in L_1;\ g_j(x) = 0, j \in L_2\}$. There are many optimization algorithms have been proposed to solve the problem (1.1), such as feasible direction methods ([1]-[2],etc), trust region techniques([3]-[4], etc), sequential system of linear equations (SSLE,such as [5]-[6], etc) algorithm ,sequential quadratic programming (SQP, such as [7]-[13], etc) methods and gradient projection methods(GPM). We will focus on the GPM method for it has the advantages of simple structure. In 1960's, Rosen first proposed GPM algorithm [14] which is based on projecting the search direction into the subspace tangent to the active constraints. It becomes a early important methods of feasible directions for solving the problem (1.1). Thereafter, a lot of authors further studied and improved the gradient projection methods(such as [15]-[18]). Du and Zhang [17] proved global convergence of Rosen's method with linear constraints while the proof was complex.

On the other side, for optimization with nonlinear constraints, gradient project direction didn't satisfy the requirement of feasibility at active constraints set. Many modified methods (such as [19]-[22]) were presented. In [19]-[20], they proposed generalized gradient projection method for for nonlinear inequality constrained optimization, which the initial point can be chosen arbitrarily.

[1]This work was supported in part by Scientific Research Fund of Hunan Provincial Education Department (NO. 12A077 and 14C0609).Corresponding the author: E-mail: ldlzj123@163.com

AMO -Advanced Modeling and Optimization. ISSN: 1841-4311

In [21]-[22], their methods were proposed by combining the concept of conjugate projection gradient with the idea of variable metric methods([23]) for nonlinear constrained optimization, and the presented algorithms possesses global convergence and superlinear convergence, but they were necessary to compute a more complex modified feasible directions of descent by the gradient project direction, thus computing cost was rather high.

In this paper, combining the properties of the gradient projection methods with the ideas of the sequential system of linear equations algorithm, we present a new gradient projection method for nonlinear programming problems (1.1). Firstly, the original problem (1.1) is transformed into an associated simpler equivalent problem with only inequality constraints, then, a descent gradient project direction is obtained by solving explicit computation formula. By solving systems of linear equations a feasible direction is generated, then, the search direction is obtained by making combination with the descent direction and the feasible direction. The projection matrix only needs to be computed one time comparison with Rosen's method, per single iteration, thus the amount of computation is lower. Under certain conditions, the algorithm is proved to possess global convergence and superlinear convergence.

The paper is organized as follows: In section 2, the algorithm is proposed. The global convergence and superlinear convergence discussed in sections 3 and 4, respectively. In the section 5, some numerical results for some typical examples are listed. And concluding remarks are given in the last section.

2. Description of Algorithm

Firstly, for the problem (1.1), we let $F(x),\ F_c(x): R^n \longrightarrow R^1$ be defined as follows

$$F(x) = -\sum_{j\in L_2} g_j(x), \qquad F_c(x) = f(x) + cF(x).$$

where the penalty parameter $c > 0$. Considering the following related family of simpler problem

$$\min\{F_c(x)|x \in X^+ = \{x \in R^n|g_j(x) \le 0, j \in L\}\}. \tag{2.1}$$

When $L_2 = \phi$, i.e., $F(x) \equiv 0$, (1.1) is the inequality constrained optimization. Obviously, $x \in X$ if and only if $x \in X^+$, and $F(x) = 0$. For $x \in X^+$, denoted

$$L(x) = \{j \in L_1 \mid g_j(x) = 0\} \bigcup L_2.$$

We define some variants as follows:

$$A(x) = \left(\nabla g_j(x), j \in L\right),\ D(x) = \mathrm{diag}(D_j(x), j \in L),\ D_j(x) = \begin{cases} g_j^2(x), & j \in L_1, \\ 0, & j \in L_2. \end{cases}$$
$$\pi(x) = -\left(A(x)^T A(x) + D(x)\right)^{-1} A(x)^T \nabla f(x). \tag{2.2}$$

Throughout this paper, the following assumption holds.

H 2.1. *The feasible sets are nonempty, i.e.,* $X \neq \phi$ *and* $X^+ \neq \phi$.

H 2.2. *The functions* $f, g_j\,(j \in I)$ *are twice continuously differentiable.*

H 2.3. *$\forall x \in X^+$, the vectors $\{\nabla g_j(x), j \in L(x)\}$ are linearly independent.*

According to Lemma 2.2 in [13], we can obtain the following result.

Lemma 2.1. *1) Suppose that H2.1 and H2.2 hold, then $\forall x \in X^+$, the matrix $(A(x)^T A(x) + D(x))$ is positive definite.*

2) If $c > \max\{|\pi_j(x)| : j \in L_2\}$, then x is a KKT point of the problem (1.1) if and only if x is a KKT point of the problem (2.1).

Due to the Lemma 2.1 above, for a current iteration point $x^k \in X^+$, we use multiplier vector $\pi_j(x^k)$ to update the penalty parameter c_k in (2.1)(as to the detail computation of c_k please see the Step 1 of the algorithm below). Furthermore, we can see that solving the original problem (1.1) can be transformed to solve a sequence optimization (2.1) of problems with inequality constraints.

Then, we describe the multiplier function as follows:

Definition 2.1. *A function $\mu(x)$ is said to be multiplier function of the problem (2.1), if*
1) $\mu(x)$is continuous, $\mu(x) \geq 0$ and
2) when x^* is a KKT piont of (2.1), $\mu(x^*)$ is KKT multiplier corresponding to x^*.

Now, the algorithm can be stated as follows.

Algorithm:

Given a starting point $x^0 \in X^+$, and an initial symmetric matrix $H^0 \in R^{n\times n}$. $\mu(x^0) \in R^{m+l}$, Choose parameters $\varepsilon_0, \theta, \upsilon \in (0,1), \alpha \in (0, \frac{1}{2}), \xi \in (2,3), \delta > 2, c_0, \epsilon \in (0, +\infty)$. Set $k = 0$;

Step 1 Update c_k :

$$t_k = \big(\max\{|\pi_j(x^k)| : j \in L_2\}\big) + c_0, c_k = \begin{cases} \max\{t_k, c_{k-1} + \epsilon\}, & \text{if } c_{k-1} < t_k, \\ c_{k-1}, & \text{if } c_{k-1} \geq t_k. \end{cases}$$

Step 2 Computate active constraint set:

Step 2.1 : Let $i = 0$, $\varepsilon_{k,i} = \varepsilon_0$;

Step 2.2 : If $det(A_{k,i}^T) \geq \varepsilon_{k,i}$, let $J_k = J_{k,i} \bigcup L_2$, $i_k = i$, go to Step 3; Otherwise, go to Step 2.3; where

$$\begin{aligned} J_{k,i} &= \{j \in I | -\varepsilon_{k,i}\mu_j(x^k) \leq g_j(x^k) \leq 0\}, \\ A_{k,i}^T &= \{\nabla g_j(x^k), j \in J_{k,i}\}; \end{aligned} \tag{2.3}$$

Step 2.3 : Let $i = i + 1$, $\varepsilon_{k,i} = 0.5\varepsilon_{k,i}$, go to Step 2.2;

Step 3 : Compute

$$\begin{cases} A_k = A(x^k) = (\nabla g_j(x^k), j \in J_k), \\ B_k = B(x^k) = (A_k^T H_k A_k)^{-1} A_k^T H_k, \\ P_k = P(x^k) = H_k(I_n - A_k B_k), \\ \chi^k = \chi(x^k) = -B_k \nabla F_c(x^k) \\ d^k = d(x^k) = -P_k \nabla F_c(x^k) + B_k^T \nu^k, \end{cases} \tag{2.4}$$

where

$$\nu^k = \nu(x^k) = (\nu_j^k,\ j \in J_k),\ \nu_j^k = \begin{cases} -g_j(x^k), & \chi_j^k > 0, \\ \chi_j^k, & \chi_j^k \leq 0. \end{cases}$$

If $d^k = 0$, STOP;

Step 4 :

Reorder the rows of A_k by finding its a maximal linearly independent rows subset, and denote

$$A_k \triangleq \begin{pmatrix} A_k^1 \\ A_k^2 \end{pmatrix},$$

where $A_k^1 \in R^{|J_k|\times|J_k|}$ is invertible matrix whose rows are $|J_k|$ linearly independent rows of A_k, and A_k^2 is the matrix whose rows are the remaining $n - |J_k|$ rows of A_k.

Step 5 :

Compute the following linear system

$$(A_k^1)^T d = -\|d^k\|^\xi e - \widetilde{g}^k,\ \widetilde{g_j}^k = \widetilde{g}_j(x^k + d^k), j \in J_k, \tag{2.5}$$

where $e = (1, \ldots, 1)^T \in R^{|J_k|}$, $\widetilde{g}^k = (\widetilde{g_j}^k, j \in J_k)$. Let $\widetilde{d_1}^k$ be the solution. Correspondingly, define s_1^k to be the vector formed by $\widetilde{d_1}^k$ and $\mathbf{0}$ such that

$$(A_k)^T s_1^k = (A_k^1, A_k^2) \begin{pmatrix} \widetilde{d_1}^k \\ \mathbf{0} \end{pmatrix} = (A_k^1)^T \widetilde{d_1}^k + (A_k^1)^T \mathbf{0} = (A_k^1)^T \widetilde{d_1}^k,$$

where $\mathbf{0} = (0, \ldots, 0)^T \in R^{n-|J_k|}$.

Step 6 *Attempted search*:

If

$$F_{c_k}(x^k + d^k + s_1^k) \le F_{c_k}(x^k) + \alpha \nabla F_{c_k}(x^k)^T d^k, \tag{2.6}$$

$$g_j(x^k + d^k + s_1^k) \le 0, j \in L, \tag{2.7}$$

set $t = 1$, $q^k = d^k + s_1^k$, go to Step 9;

Step 7 :

By solving the following linear system

$$(A_k^1)^T d = -\|d^k\| e, \tag{2.8}$$

Let $\widetilde{d_2}^k$ be an solution of (2.8). According to the combination of s_1^k, we might as well let $s_2^k \triangleq \begin{pmatrix} \widetilde{d_2}^k \\ \mathbf{0} \end{pmatrix}$ such that

$$(A_k)^T s_2^k = (A_k^1)^T \widetilde{d_2}^k + (A_k^1)^T \mathbf{0} = (A_k^1)^T \widetilde{d_2}^k.$$

Establish a convex combination of d^k and s_2^k:

$$\begin{aligned} q^k &= (1 - \tau_k) d^k + \tau_k s_2^k, \\ \tau_k &= \begin{cases} 1, & \nabla F_{c_k}(x^k)^T s_2^k \le \theta \nabla f(x^k)^T d^k \\ \frac{(1-\theta)\nabla F_{c_k}(x^k)^T d^k}{\nabla F_{c_k}(x^k)^T (d^k - s_2^k)}, & \nabla F_{c_k}(x^k)^T s_2^k > \theta \nabla F_{c_k}(x^k)^T d^k \end{cases}; \end{aligned} \tag{2.9}$$

Step 8 *Line search*:

Compute t_k, the first number t in the sequence $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots\}$ satisfying

$$F_{c_k}(x^k + t q^k) \le F_{c_k}(x^k) + \upsilon t \nabla F_{c_k}(x^k)^T q^k, \tag{2.10}$$

$$g_j(x^k + tq^k) \leq 0, \qquad j \in L. \tag{2.11}$$

Step 9 Update:

Let H_{k+1} be a new symmetric approximation of the Hessian matrix. Set $x^{k+1} = x^k + t_k q^k$ and $k = k + 1$; Go back to Step 2.

Remarks 1: In our algorithm, there exist a inner loop in Step 2, and two outer loop between step 1 and step 9, i.e.,Cycle A: 1-2-3-4-5-6-9, and Cycle B: 1-2-3-4-7-8-9.

Remarks 2: If the attempted search in step 6 is successful, Cycle A finishes a complete iteration, and Cycle B is not performed in this iteration. Otherwise, we perform cycle B to ensure the well-definition of the algorithm.

3. Global Convergence of Algorithm

In this section, we firstly prove the global convergence of algorithm. Under mild conditions, we will discuss the strong convergence of the proposed Algorithm.

H 3.1. *$\{x^k\}$ is bounded, and there exist two constants $0 < a \leq b$, such that $a||d||^2 \leq d^T H_k d \leq b||d||^2$, for all k, for all $d \in R^n$.*

Lemma 3.1. *If the sequence $\{x^k\}$ is bounded, then there exists a k_0, such that $c_k \equiv c_{k_0} \triangleq c$ for all $k \geq k_0$.*

The proof is similar to the fashion of Lemma 3.1 in [13]. Due to this lemma, in the rest of this paper, we always assume that $c_k \equiv c$ for all k for k large enough .

Lemma 3.2. *1) For any iteration, there is no infinite cycle in step 2.*

2) If $\{x^k\}_{k\in K} \to x^*$, then there exists a constant $\bar{\varepsilon} > 0$, such that $\varepsilon_{k,i_k} \geq \bar{\varepsilon}$, for $k \in K, k$ large enough.

Proof. 1) Suppose by contradiction that the desired conclusion is false, i.e., $\exists k$, such that there is an infinite cycle in step 2, then it holds that

$$\det(A_{k,i}^T A_{k,i}) = 0, i = 1, 2, \ldots,$$

and by (2.3), we know that $J_{k,i+1} \subseteq J_{k,i}$. Since there are only finitely many choices for $J_{k,i} \subseteq L_1$, it is sure, for i large enough, that $J_{k,i+1} \equiv J_{k,i} \triangleq \overline{J}_k$. So, with $i \to \infty$, we have

$$\overline{J}_k = L_1(x^k) = \{j \in L_1 \mid g_j(x^k) = 0\}, \;\; \det(A^T_{L_1(x^k)} A_{L_1(x^k)}) = 0.$$

This is a contradiction with H 2.3, which shows that the first statement is true.

2) Suppose by contradiction that the desired conclusion is false, too, that is to say, there exists $K' \subseteq K(|K'| = \infty)$, such that

$$\varepsilon_{k,i_k} \to 0, k \in K', k \to \infty.$$

Let $\overline{J}_k = J_{k,i_k-1}$. From the definition of ε_{k,i_k}, it holds, for $k \in K', k$ large enough, that

$$\det\left(A^T_{\overline{J}_k} A_{\overline{J}_k}\right) = 0, \;\; -2\varepsilon_{k,i_k}\mu_j(x^k) \leq g_j(x^k) \leq 0, j \in \overline{J}_k.$$

Since there are only finitely many choices for sets $\overline{J}_k \subseteq L_1$, it is sure that there exists $K'' \subseteq K'(|K''| = \infty)$, such that $\overline{J}_k \equiv \overline{J}$, for $k \in K'', k$ large enough. Denote $\overline{A} = \left\{\nabla g_j(x^*) \mid j \in \overline{J}\right\}$, then, with $k \in K'', k \to \infty$, it holds that

$$\det\left(\overline{A}^T\overline{A}\right) = 0, g_j(x^*) = 0, j \in \overline{J} \subseteq L(x^*).$$

This is a contradiction with H 2.3, too, which shows that the second statement is true. ∎

Lemma 3.3. *1) If x^k is a KKT point of (2.1)$\Longleftrightarrow d^k = 0$.*
2) If x^k is not a KKT point of (2.1), then

$$\nabla F_c(x^k)^T d^k < 0,\ \nabla F_c(x^k)^T q^k < 0,\ \nabla g_j(x^k)^T q^k < 0, j \in L(x^k). \tag{3.1}$$

Proof. 1) Firstly, we have

$$P_k A_k = \mathbf{0},\ P_k H_k P_k = P_k,\ B_k A_k = E_{|J_k|}.$$

If $d^k = 0$, from the definition of d^k, we know that

$$0 = A_k^T d^k = \nu^k, P_k \nabla F_c(x^k) = 0.$$

Then, from (2.4) and H_k is positive definite matrix, we have

$$\chi_j^k \geq 0,\ \chi_j^k g_j(x^k) = 0, j \in J_k, \nabla F_c(x^k) + A_k \chi^k = 0, \tag{3.2}$$

which implies that x^k is a KKT point of (2.1).

On the contrary, if x^k is a KKT point of (2.1), from the definition of J^k, we know that there exist a vector $a = (a_j,\ j \in J_k)$ such that

$$\nabla F_c(x^k) + A_k a = 0,\ a_j \geq 0,\ a_j g_j(x^k) = 0,\ j \in J_k. \tag{3.3}$$

Furthermore,

$$a = -(A_k^T H_k A_k)^{-1} A_k^T H_k \nabla F_c(x^k) = -B_k \nabla F_c(x^k) = \chi^k,$$

then, from (2.4) and (3.3), it is easy to obtain

$$\nu^k = 0,\ H_k \nabla F_c(x^k) - H_k A_k B_k \nabla F_c(x^k) = 0,\ P_k \nabla F_c(x^k) = 0,$$

which shows $d^k = 0$.

2) If x^k is not a KKT point of (2.1), then, $d^k \neq 0$, we have

$$\begin{aligned} &\nabla F_c(x^k)^T d^k \\ &= -\nabla F_c(x^k)^T P_k \nabla F_c(x^k) - (\chi^k)^T \nu^k, \\ &= -(P_k \nabla F_c(x^k))^T H_k^{-1} (P_k \nabla F_c(x^k)) - \sum_{\chi_j^k \leq 0} (\chi_j^k)^2 + \sum_{\chi_j^k > 0} \chi_j^k g_j(x^k)) \\ &< 0. \end{aligned} \tag{3.4}$$

Now we prove that $\nabla g_j(x^k)^T q^k < 0, j \in L(x^k)$. For

$$A_k^T d^k = \nu^k,\ \ \nabla g_j(x^k)^T d^k = \nu_j^k \leq 0, j \in L(x^k) \subseteq J_k,$$

and

$$A_k^T s_2^k = (A_k^1)^T \widetilde{d}_2^{\,k} = -\|d^k\| e,\ \ \nabla g_j(x^k)^T s_2^k = -\|d^k\| < 0,$$

taking into account $\tau_k \in (0,1]$,then we have

$$\nabla g_j(x^k)^T q^k = (1-\tau_k)\nabla g_j(x^k)^T d^k + \tau_k \nabla g_j(x^k)^T s_2^k \leq \tau_k \nabla g_j(x^k)^T s_2^k < 0.$$

For $\nabla F_c(x^k)^T q^k < 0$, two cases of $\tau_k = 1$ and $\tau_k \in (0,1)$ are considered in the remainder discussion. If $\tau_k = 1$, i.e. $\nabla F_c(x^k) s_2^k \leq \theta \nabla F_c(x^k) d^k$, we have

$$\nabla F_c(x^k)^T q^k = \nabla F_c(x^k)^T s_2^k \leq -\theta \nabla F_c(x^k)^T d^k < 0.$$

If $\nabla F_c(x^k) s_2^k > \theta \nabla F_c(x^k) d^k$, it is easy to obtain

$$\nabla F_c(x^k)^T q^k = (1-\tau_k)\nabla F_c(x^k)^T d^k + \tau_k \nabla F_c(x^k)^T s_2^k = -\theta \nabla F_c(x^k)^T d^k < 0.$$

The proof is completed. ∎

According to (3.1), it can be proven that the line search in step 8 is always completed, and thus that the algorithm is well-defined.

In the sequel, we'll prove that the algorithm is globally convergent. Since there are only finitely many choices for sets $L(x^k) \subseteq J_k \subseteq L$, without loss of generality, we might as well assume that there exists a subsequence K, such that $L(x^k) \equiv L_*, J_k \equiv J_*$ and

$$x^k \to x^*, q^k \to q^*, d^k \to d^*, H_k \to H_*, B_k \to B_*, P_k \to P_*, \chi^k \to \chi^*, k \in K,$$

where $L_*,\ J_*$ are constant sets.

Theorem 3.4. *The algorithm either stops at a KKT point x^k of the problem (1.1) in finite iteration, or generates an infinite sequence $\{x^k\}$ any accumulation point x^* of which is a KKT point of the problem (1.1).*

Proof. The first statement is easy to show, since the only stopping point is in step 3. Thus, assume that the algorithm generates an infinite sequence $\{x^k\}$. According to Lemma 2.1, it is only necessary to prove that $d^* = 0$.

From Lemma 3.3, it is obvious that $\{f(x^k)\}$ is monotonous decreasing. So, according to assumption H 2.2, the fact that $\{x^k\}_K \to x^*$ implies that

$$f(x^k) \to f(x^*), k \to \infty. \tag{3.5}$$

If there exists an infinite subsequence $K_1 \subseteq K$, such that for all $k \in K_1, x^{k+1} = x^k + \lambda_k d^k$ are generated by step 6 and step 9, then from (2.6), and (3.5), it is easy to see that

$$0 = \lim_{k \in K_1} (F_c(x^{k+1}) - F_c(x^k)) \leq \lim_{k \in K_1} \alpha \nabla F_c(x^k)^T d^k \leq 0,$$

i.e. $\nabla F_c(x^k)^T d^k \to 0$. So, $d^k \to 0, k \in K_1$. According to $d^k \to d^*, k \in K$, one get $d^k \to 0, k \in K$, i.e. $d^* = 0$.

Without loss of generality, we then suppose that $x^{k+1} = x^k + \lambda_k d^k$ are generated by step 8 and step 9 on K. Suppose by contradiction that $d_0^* \neq 0$. Then, imitating the proof of Lemma 3.3, it is easy to see that d^* is well-defined, and it holds that

$$\nabla F_c(x^*)^T d^* < 0,\ \nabla F_c(x^*)^T q^* < 0,\ \nabla g_j(x^*)^T q^* < 0,\ j \in L(x^*) \subseteq J_*. \tag{3.6}$$

Furthermore, it is obvious, for k large enough, that

$$\nabla F_c(x^*k)^T d^k \leq \frac{1}{2}\nabla F_c(x^*)^T d^* < 0,$$
$$\nabla F_c(x^k)^T q^k \leq \frac{1}{2}\nabla F_c(x^*)^T q^* < 0,$$
$$\nabla g_j(x^k)^T d^k \leq \frac{1}{2}\nabla g_j(x^*)^T q^* < 0.$$

Then, it holds that the step-size λ_k obtained in step 6 are bounded away from zero on K, i.e.,

$$\lambda_k \geq \lambda_* = \inf\{\lambda_k, k \in K\} > 0, k \in K. \tag{3.7}$$

Hence, from (2.10), (3.5),(3.6) and (3.7), it holds that

$$0 = \lim_{k\in K}(F_c(x^{k+1}) - F_c(x^k)) \leq \lim_{k\in K} \upsilon\lambda_k \nabla F_c(x^k)^T q^k \leq \frac{1}{2}\upsilon\lambda_* F_c(x^*)^T q^* < 0, \tag{3.8}$$

It is a contradiction, which implies that $d^k \to 0, k \in K, k \to \infty$. Thus, from Lemma 3.3, it shows that x^* is a KKT point of (2.1). Then, in view of $c > |\pi_j(x^*)|,\ j \in L_2$, we know that x^* is a KKT point of the problem (1.1) from Lemma 2.1. ∎

4. Superlinear Convergence of Algorithm

In this section, we discuss the convergent rate of the proposed algorithm in section 2, and under some mild assumptions prove that the sequence $\{x^k\}$ generated by the algorithm is one-step superlinearly convergent. For this purpose, we add following additional assumptions.

H 4.1. *Suppose that the sequence $\{x^k\}$ of points generated by Algorithm is bounded, and has a limit point x^*. The second-order sufficiency conditions with strict complementary slackness are satisfied at the KKT point x^* and corresponding multipliers u^* of the problem (1.1).*

Firstly, the task is to show that, under these addition assumptions, the proposed algorithm is strongly convergent. According to H4.1 and Proposition 4.1 in article [11], we can obtain the following conclusion.

Lemma 4.1. *Let H2.2∼ H4.1 holds, $\lim_{k\to\infty} ||x^{k+1}-x^k|| = 0$. Thereby, the entire sequence$\{x^k\}$ converges to x^*, i.e. $x^k \to x^*, k \to \infty$.*

Lemma 4.2. *Suppose that H2.2 ∼ H4.1 holds, for k large enough, it holds that*

$$J_k \equiv L(x^*) \triangleq L_*, \quad \lim_{k\to\infty} d^k = 0, \quad \lim_{k\to\infty} \chi^k = (u_j^*,\ j \in L_*)$$

Proof. The proof of this lemma is similar to the Lemma 3 in [22].

Lemma 4.3. *For k large enough,*

1) set $\widetilde{u}^k = \chi^k + (A_k^T H_k A_k)^{-1} g(x^k),\ g(x^k) = (g_j(x^k),\ j \in J_k)$, then there exists constants $b,\ \eta > 0$, such that

$$\sum_{j\in L_*} \widetilde{u}_j^k g_j(x^k) \leq -\eta z_k,\ z_k \triangleq \left(\sum_{j\in L_*} g_j^2(x^k)\right)^{\frac{1}{2}}, \tag{4.1}$$

$$\nabla F_c(x^k)^T d^k \leq -b||d^k||^2. \tag{4.2}$$

2) s_1^k obtained in step 5 satisfies:

$$||s_1^k|| = O(||d^k||^2). \tag{4.3}$$

proof. 1). In view of $J_k \equiv L_*$, $\lim\limits_{k\to\infty} \chi^k = (u_j^*,\ j \in L_*)$ and H4.1, for k large enough we know that

$$\lim_{k\to\infty} \widetilde{u}^k = \lim_{k\to\infty} \chi^k = (u_j^*,\ j \in L_*),\ \widetilde{u}_j^k > 0,\ \chi_j^k > 0, j \in J_k.$$

Then from (2.4) we have $d^k = -P_k \nabla F_c(x^k) + B_k g(x^k)$. Hence, it holds that

$$A_k^T d^k = g(x^k),\ g_j(x^k) + \nabla g_j(x^k)^T d^k = 0,\ j \in J_k,$$

and

$$H_k^{-1} d^k + A_k \widetilde{u}^k + \nabla F_c(x^k) = 0.$$

So, according to $\widetilde{u}_j^k > 0,\ j \in L_*$, there exists a constant $\eta > 0$, such that

$$\sum_{j\in L_*} \widetilde{u}_j^k g_j(x^k) = -\sum_{j\in L_*} \widetilde{u}_j^k \left|g_j(x^k)\right| \le -\sum_{j\in L_*} \eta \left|g_j(x^k)\right| \le -\eta z_k.$$

i.e. (4.1) holds. Furthermore, we have

$$\begin{aligned} \nabla F_c(x^k)^T d^k &= -(d^k)^T H_k d^k + \textstyle\sum_{j\in L_*} \widetilde{u}_j^k g_j(x^k) \\ &\le -b||d_0^k||^2 + o(z_k) - \eta z_k \le -b||d^k||^2. \end{aligned} \tag{4.4}$$

Which shows that (4.2) holds.

2). Firstly, it holds that

$$g_j(x^k + d^k) = g_j(x^k) + \nabla g_j(x^k)^T d^k + O\left(\|d^k\|^2\right) = O\left(||d^k||^2\right), j \in I_*,$$

i.e., $||\widetilde{g}^k|| = O\left(||d^k||^2\right)$. Then, in view of $\tau \in (2,3)$, and A_k^1 is invertible, it is evident that

$$||\widetilde{d_1^k}|| = O\left(||d^k||^2\right),\ ||s_1^k|| = O\left(||d^k||^2\right).$$

The claim holds. ∎

To ensure the step size unit can be accepted, the following assumption about the symmetric matrix satisfied:

H 4.2. *Let*

$$\left\|\widetilde{P}_k \left(H_k - \nabla_{xx}^2 \widetilde{L}(x^k, \widetilde{u}^k)\right) d^k\right\| = o\left(||d^k||\right),$$

where

$$\widetilde{P}_k = I_n - A_k (A_k^T A_k)^{-1} A_k^T,\ \nabla_{xx}^2 \widetilde{L}(x^k, \widetilde{u}^k) = \nabla^2 F_c(x^k) + \sum_{j\in L_*} \widetilde{u}_j^k \nabla^2 g_j(x^k).$$

Lemma 4.4. *For k large enough, the inequalities of Step 6 are satisfied. i.e.*

$$x^{k+1} = x^k + d^k + s_1^k, t_k \equiv 1.$$

Proof. To prove the results, we only necessary to prove that

$$F_c(x^k + d^k + s_1^k) \le F_c(x^k) + \alpha \nabla F_c(x^k)^T d^k, \tag{4.5}$$

$$g_j(x^k + d^k + s_1^k) \le 0, j \in L, \tag{4.6}$$

First, we show that (4.6) are true.

For $j \in L \setminus L_*$, from the facts $g_j(x^*) < 0, x^k \to x^*, d^k \to 0$, and Lemma 4.3 2), it is easy to show that $g_j(x^k + d^k) \le 0$.

For $j \in L_*$, expanding $g_j(x^k + d^k + s_1^k)$ around $x^k + d^k$, we have

$$\begin{aligned} & g_j(x^k + d^k + s_1^k) \\ & = g_j(x^k + d^k) + \nabla g_j(x^k + d^k)^T s_1^k + O\left(||s_1^k||^2\right) \\ & = g_j(x^k + d^k) + \nabla g_j(x^k)^T s_1^k + O\left(||d^k|| \cdot ||s_1^k||\right) \end{aligned}$$

Again, from (2.5), it holds that

$$A_k^T s_1^k = (A_k^1)^T \widetilde{d_1^k} = -||d^k||^\xi e - \widetilde{g}^k,$$

i.e.,

$$g_j(x^k + d^k) + \nabla g_j(x^k)^T s_1^k = -||d^k||^\xi, j \in L_*. \tag{4.7}$$

Considering $\xi \in (2, 3)$, for any $j \in L$, the (4.6) holds.

Then, we prove that (4.5) holds. Denote

$$\begin{aligned} s & \triangleq F_c(x^k + d^k + s_1^k) - F_c(x^k) - \alpha \nabla F_c(x^k)^T d^k \\ & = \nabla F_c(x^k)^T (d^k + s_1^k) + \tfrac{1}{2}(d^k)^T \nabla^2 F_c(x^k) d^k - \alpha \nabla F_c(x^k)^T d^k + o\left(||d^k||^2\right). \end{aligned}$$

While, from (4.4), we have

$$\begin{aligned} & \nabla F_c(x^k)^T d^k = -(d^k)^T H_k d^k - \sum_{L_*} \widetilde{u}_j^k \nabla g_j(x^k)^T d^k, \\ & \nabla F_c(x^k)^T (d^k + s_1^k) = -(d^k)^T H_k d^k - \sum_{L_*} \widetilde{u}_j^k \nabla g_j(x^k)^T (d^k + s_1^k) + o\left(||d^k||^2\right), \\ & g_j(x^k) + \nabla g_j(x^k)^T d^k = o(\|d^k\|^2), j \in L_*. \end{aligned}$$

From (4.7), we have

$$g_j(x^k) + \nabla g_j(x^k)^T d^k + \nabla g_j(x^k)^T s_1^k + \frac{1}{2}(d^k)^T \nabla^2 g_j(x^k) d^k = o\left(||d^k||^2\right), j \in L_*.$$

i.e.,

$$-\sum_{j \in L_*} \widetilde{u}_j^k \nabla g_j(x^k)^T (d^k + s_1^k) = \sum_{j \in L_*} \widetilde{u}_j^k g_j(x^k) + \frac{1}{2}(d^k)^T \left(\sum_{j \in L_*} \widetilde{u}_j^k \nabla g_j^2(x^k) \right) d^k + o\left(||d^k||^2\right).$$

So, it is clear that

$$\begin{aligned} s & = (\alpha - 1)(d^k)^T H_k d^k + \tfrac{1}{2}(d^k)^T \nabla_{xx}^2 \widetilde{L}(x^k, \widetilde{u}_j^k) d^k + \sum_{j \in I_*} (1 - \alpha) \widetilde{u}_j^k g_j(x^k) + o\left(||d^k||^2\right) \\ & \le \left(\alpha - \tfrac{1}{2}\right) a ||d^k||^2 + \tfrac{1}{2}(d^k)^T \left(\nabla_{xx}^2 \widetilde{L}(x^k, \widetilde{u}_j^k) d^k - H_k\right) d^k + (1 - \alpha) \sum_{j \in L_*} \widetilde{u}_j^k g_j(x^k) + o\left(||d^k||^2\right). \end{aligned}$$

Denote $A_* = (\nabla g_j(x^*), j \in L_*), \widetilde{P}_* = I_n - A_*(A_*^T A_*)^{-1} A_*^T$, then $\widetilde{P}_k \to \widetilde{P}_*$. Let

$$d^k = \widetilde{P}_* d^k + y_k, \ y_k = A_* \left(A_*^T A_*\right)^{-1} A_*^T d^k,$$

then, we have

$$y_k = A_* \left(A_*^T A_*\right)^{-1} (A_* - A_k)^T d^k + A_* \left(A_*^T A_*\right)^{-1} A_k^T d^k$$

$$||y_k|| = o\left(||d^k||\right) + O\left((\sum_{L_*} g_j^2(x^k))^{\frac{1}{2}} \right).$$

Thereby, it can be seen that

$$s \le a(\alpha - \tfrac{1}{2})||d^k||^2 + \tfrac{1}{2}\left((d^k)^T P_* + y_k^T\right)\left(\nabla^2_{xx}\widetilde{L}(x^k, \widetilde{u}_j^k) - H_k\right) d^k$$
$$+(1-\alpha)\sum_{L_*} b_j^k g_j(x^k) + o\left(||d^k||^2\right)$$
$$= a(\alpha - \tfrac{1}{2})||d^k||^2 + o(||d^k||^2) + (1-\alpha)\sum_{L_*}\widetilde{u}_j^k g_j(x^k) + o\left((\sum_{L_*} g_j^2(x^k))^{\frac{1}{2}}\right).$$

According to

$$\alpha \in \left(0, \frac{1}{2}\right), \widetilde{u}_j^k \to u_j^* > 0, j \in L_*,$$

it holds, for k large enough, that $s \le 0$, i.e., (4.5) is true.

Moreover, in view of Lemma 4.4 and the way of Theorem 5.2 in [10], we obtain the following theorem:

Theorem 4.5. *Under all above-mentioned assumptions, the algorithm is superlinearly convergent, i.e., the sequence* $\{x^k\}$ *generated by the algorithm satisfied* $||x^{k+1} - x^*|| = o(||x^k - x^*||)$.

5. Numerical Experiments

In this section, we select some problems in Ref.[24] to show the efficiency of our Algorithm in section 2. The code of the proposed algorithm is written by using MATLAB 7.0 and and run on Win 7.

1) The algorithm parameters were set as follows: $\varepsilon_0 = 0.01, \theta = 0.1, \upsilon = 0.1, \alpha = 0.25, \xi = 2.25, \delta = 2.5, c_0 = 2, \epsilon = 10^3$ and $H_0 = I$, the $n \times n$ unit matrix.

2) H_k is updated by the BFGS formula similar to [8].

$$H_{k+1} = H_k - \frac{H_k s_k (H_k s_k)^T}{s_k^T H_k s_k} + \frac{\hat{\eta}_k \hat{\eta}_k^T}{s_k^T \hat{\eta}_k}$$

where

$$s_k = x^{k+1} - x^k, \quad \hat{\eta}_k = \theta_k \gamma_k + (1-\theta_k) H_k s_k,$$

$$\hat{\gamma}_k = \nabla_x L(x^{k+1}, \lambda^k, u^k) - \nabla_x L(x^k, \lambda^k, u^k),$$

$$\nabla_x L(x^k, \lambda^k, u^k) = \sum_{j \in L_1} \lambda_j \nabla g_j(x) + \sum_{j \in L_2} u_j \nabla g_j(x),$$

$$\theta_k = \begin{cases} 1, & s_k^T \hat{\gamma}_k \ge 0.2 s_k^T H_k s_k, \\ \dfrac{0.8 s_k^T H_k s_k}{s_k^T H_k s_k - s_k^T \hat{\gamma}_k}, & s_k^T \hat{\gamma}_k < 0.2 s_k^T H_k s_k. \end{cases}$$

3) In the implementation, the stopping criterion of Step 3 is changed to $If \; \|d^k\| \le 10^{-6} \quad STOP$.

The numerical results for the test problems are listed in Table 1. For each test problem, Pro. is the number of the test problem in Ref.[24], NT the number of iterations, IP the initial point, FV the final value of the objective function. In this paper, we obtain the conclusion that the algorithm stops when $d^k = 0$.

Table 1 The detailed information of the results

Pro.	NT	IP	$\|d^k\|$	FV
HS03	11	$(1,5)^T$	8.064125836437674 e-006	0.040353623273424
HS10	10	$(3,1)^T$	9.504722714043553 e-006	-0.999999447862117
HS22	7	$(0,0)^T$	1.143437043146816 e-006	1.000001836160267
HS29	12	$(1,3)^T$	6.451489138418337 e-006	-22.627416993190010
HS32	9	$(1,7,2)^T$	1.200341952243785 e-006	1.000000000000000
HS41	35	$(0,5,3,2)^T$	1.838268694827953e-006	1.926000098999896
HS42	7	$(4,4,2,8)^T$	4.0016543875696623e-006	1.385798987485438
HS43	10	$(4,2)^T$	7.085351411397026 e-006	-43.999992576356789

6. Conclusion

In this paper, we propose a new projection gradient method for constrained optimization problems. In a single iteration, a feasible direction is generated by solving systems of linear equations, which the cost of of computation is reduced by using the technique of systems of linear equations. Then, by making combination with the descent direction and the feasible direction, the search direction is obtained. In comparison with Rosen's method, our method requires to solve only one projection matrix, in addition, analysis of convergence is simpler than that of Rosen's method. Under mild conditions, the algorithm is proved to possess global convergence and superlinear convergence. The preliminary numerical results also show that the proposed algorithm is effective.

References

[1] Zoutendijk G. Methods of feasible directions: a study in linear and non-linear programming[J]. 1960.

[2] Du D Z, Wu W, Pardalos P M, et al. Feasible Direction Method[J]. Wiley Encyclopedia of Operations Research and Management Science, 2010.

[3] Powell M J D, Yuan Y. A trust region algorithm for equality constrained optimization[J]. Mathematical Programming, 1990, 49(1): 189-211.

[4] Byrd R H, Gilbert J C, Nocedal J. A trust region method based on interior point techniques for nonlinear programming[J]. Mathematical Programming, 2000, 89(1): 149-185.

[5] Gao Z Y, He G P, Wu F. Sequential systems of linear equations algorithm for nonlinear optimization problems with general constraints[J]. Journal of Optimization Theory and Applications, 1997, 95(2): 371-397.

[6] Jian J, Cheng W. A superlinearly convergent strongly sub-feasible SSLE-type algorithm with working set for nonlinearly constrained optimization[J]. Journal of Computational and Applied Mathematics, 2009, 225(1): 172-186.

[7] Han S P. A globally convergent method for nonlinear programming[J]. Journal of optimization theory and applications, 1977, 22(3): 297-309.

[8] Powell M J D. A fast algorithm for nonlinearly constrained optimization calculations[M]. Numerical analysis. Springer Berlin Heidelberg, 1978, 144-157.

[9] Boggs P T, Tolle J W. Sequential quadratic programming[J]. Acta numerica, 1995, 4: 1-51.

[10] Facchinei F, Lucidi S. Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems[J]. Journal of Optimization Theory and Applications, 1995, 85(2): 265-289.

[11] Panier E R, Tits A L, Herskovits J N. A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization[J]. SIAM Journal on Control and Optimization, 1988, 26(4): 788-811.

[12] Zhao W L, Song D J, and Liu B Z, Error Bounds and Finite Termination for Constrained Optimization Problems, Mathematical Problems in Engineering, vol. 2014, Article ID 158780, 10 pages, 2014. doi:10.1155/2014/158780.

[13] Jian J B, Tang C M, Hu Q J, et al. A feasible descent SQP algorithm for general constrained optimization without strict complementarity[J]. Journal of Computational and Applied Mathematics, 2005, 180(2): 391-412.

[14] Rosen J B. The gradient projection method for nonlinear programming. Part I. Linear constraints[J]. Journal of the Society for Industrial & Applied Mathematics, 1960, 8(1): 181-217.

[15] Goldfarb D, Lapidus L. Conjugate gradient method for nonlinear programming problems with linear constraints[J]. Industrial & Engineering Chemistry Fundamentals, 1968, 7(1): 142-151.

[16] Yue Y M, Han J Y. A unified approach to the feasible direction methods for nonlinear programming with linear constraints[J]. Acta Mathematicae Applicatae Sinica, 1984, 1(1): 63-75.

[17] Du D Z, Zhang X S. Global convergence of Rosen's gradient projection method[J]. Mathematical Programming, 1989, 44(1-3): 357-366.

[18] Zeng Q G. An extension of Rosen's gradient projection method and a proof of its global convergence[J]. Acta Mathematicae Applicatae Sinica, 1991, 14(3): 312-322.

[19] Sun Q Y. Generalized super-memory gradient projection method with arbitrary initial point and conjugate gradient scalar for nonlinear programming with nonlinear inequality constraints[J]. Mathematica Numerica Sinica, 2004, 26(4): 401-412.

[20] Jian J B, Zeng Y F, Tang C M. A generalized super-memory gradient projection method of strongly sub-feasible directions with strong convergence for nonlinear inequality constrained optimization[J]. Computers & Mathematics with Applications, 2007, 54(4): 507-524.

[21] Shi Z J. A class of global convergent conjugate projection gradient method and its superlinear convergence rate[J]. Mathematica Numerica Sinica,1996, 4: 411-421.

[22] Zhu Z B, Zhang K C. A new conjugate projection gradient method and superlinear convergence[J]. Acta Mathematicae Applicatae Sinica, 2004, 27(1): 149-161.

[23] Davidon W C. Variable metric method for minimization[J]. SIAM Journal on Optimization, 1991, 1(1): 1-17.

[24] Hock W, Schittkowski K. Test examples for nonlinear programming codes[M]. Lectures Notes in Economics and Mathematical Systems, Vol. 187, Berlin: Springer, 1981.