Volume 31, N. 2, pp. 407–432, 2012

ISSN 0101-8205 / ISSN 1807-0302 (Online)
www.scielo.br/cam

A new double trust regions SQP method without a penalty function or a filter*

XIAOJING ZHU** and DINGGUO PU

Department of Mathematics, Tongji University, Shanghai, 200092, PR China

E-mail: 042563zxj@163.com

Abstract. A new trust-region SQP method for equality constrained optimization is considered. This method avoids using a penalty function or a filter, and yet can be globally convergent to first-order critical points under some reasonable assumptions. Each SQP step is composed of a normal step and a tangential step for which different trust regions are applied in the spirit of Gould and Toint [Math. Program., 122 (2010), pp. 155-196]. Numerical results demonstrate that this new approach is potentially useful.

Mathematical subject classification: 65K05, 90C30, 90C55.

Key words: equality constrained optimization, trust-region, SQP, global convergence.

1 Introduction

We consider nonlinear equality constrained optimization problems of the form

$$\begin{cases} \min & f(x) \\ \text{s.t.} & c(x) = 0, \end{cases} \tag{1.1}$$

where we assume that $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ with $m \le n$ are twice differentiable functions.

#CAM-414/11. Received: 16/IX/11. Accepted: 22/IX/11.
*This research is supported by the National Natural Science Foundation of China (No. 10771162).
**Corresponding author.

A new method for first order critical points of problem (1.1) is proposed in this paper. This method belongs to the class of two-phase trust-region methods, e.g., Byrd, Schnabel, and Shultz [3], Dennis, El-Alem, and Maciel [6], Gomes, Maciel, and Martínez [13], Gould and Toint [15], Lalee, Nocedal, and Plantenga [17], Omojokun [21], and Powell and Yuan [23]. Also, our method, since it deals with two steps, can be classified in the area of inexact restoration methods proposed by Martínez, e.g., [1, 2, 8, 18, 19, 20].

The way we compute trial steps is similar to Gould and Toint's approach [15] that uses different trust regions. Each step is decomposed into a normal step and a tangential step. The normal step is computed by solving a vertical subproblem which aims to minimize the Gauss-Newton approximation of the infeasibility measure within a normal trust region. The tangential step is computed by solving a horizontal subproblem which aims to minimize the quadratic model of the Lagrangian within a tangential trust region on the premise of controlling the linearized infeasibility measure. Similarly, in Martínez's inexact restoration methods, a more feasible intermediate point is computed in the feasibility phase, and then a trial point is computed on the tangent set that passes through the intermediate point to improve the optimality measure.

In most common constrained optimization methods, penalty functions are used to decide whether to accept trial steps. Nevertheless, there exist several difficulties associated with using penalty functions, and in particular the choice of penalty parameters. A too low parameter can result in an infeasible point being obtained, or even an unbounded increase in the penalty. On the other hand, a too large parameter can weaken the effect of the objective function, resulting for example in slow convergence when the iterates follow the boundary of the feasible region. To avoid using a penalty function, Fletcher and Leyffer [10] proposed filter techniques that allow a step to be accepted if it sufficiently reduces either the objective function or the constraint violation. For more theoretical and algorithmic details on filter methods, see, e.g., [4, 9, 11, 14, 24, 25, 26, 27, 28].

The main feature of our method is that a new step acceptance mechanism that avoids using a penalty function or a filter, and yet can promote global convergence. In this sense, our method shares some similarities with Bielschowsky and Gomes' dynamic control of infeasibility (DCI) method [1] and Gould and

Toint's trust funnel method [15]. These methods adopt the idea of progressively reducing the infeasibility measure. Of course, the new step acceptance mechanism in this paper is quite different from the trust funnel and the trust cylinder used in DCI.

The paper is organized as follows. In Section 2, we describe some main details on the new algorithm. Assumptions and global convergence analysis are presented in Section 3. Section 4 is devoted to some numerical results. Conclusions are made in Section 5.

2 The algorithm

2.1 Step computation. At the beginning of this section we define an infeasibility measure as follows

$$h(x) \triangleq \frac{1}{2}||c(x)||^2 \tag{2.1}$$

where $||\cdot||$ denotes the Euclidean norm.

Each SQP step is composed of a normal step and a tangential step for which different trust regions are used in the spirit of [15]. The normal step aims to reduce the infeasibility, and the tangential step which approximately lies on the plane tangent to the constraints aims to reduce the objective function as much as possible.

The normal step n_k is computed by solving the trust-region linear least-squares problem, i.e.,

$$\begin{cases} \min & \frac{1}{2}||c_k + A_k v||^2 \\ \text{s.t.} & ||v|| \le \Delta_k^c. \end{cases} \tag{2.2}$$

Here $c_k = c(x_k)$ and $A_k = A(x_k)$ is the Jacobian of $c(x)$ at x_k. We do not require an exact Gauss-Newton step for (2.2), but a Cauchy condition

$$\delta_k^{c,n} \triangleq \frac{1}{2}||c_k||^2 - \frac{1}{2}||c_k + A_k n_k||^2 \ge \kappa_c ||A_k^T c_k|| \min\left(\frac{||A_k^T c_k||}{1 + ||A_k^T A_k||}, \Delta_k^c\right) \tag{2.3}$$

for some constant $\kappa_c \in (0, 1)$. In addition, we assume the boundedness condition

$$||n_k|| \le \kappa_n ||c_k|| \tag{2.4}$$

for some constant $\kappa_n > 0$. Note that the above two requirements on n_k are very reasonable in both theory and practice. If x_k is a feasible point, we set $n_k = 0$.

After obtaining n_k, we then aims to find a tangential step t_k such that

$$||t|| \leq \Delta_k^f \tag{2.5}$$

to improve the optimality on the premise of controlling the linearized infeasibility measure. Define a quadratic model function

$$m_k(x_k + d) \triangleq f_k + \langle g_k, d\rangle + \frac{1}{2}\langle d, B_k d\rangle,$$

where $f_k = f(x_k), g_k = \nabla f(x_k)$, and B_k is an approximate Hessian of the Lagrangian

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^T c(x).$$

Then we have

$$m_k(x_k + n_k) = f_k + \langle g_k, n_k\rangle + \frac{1}{2}\langle n_k, B_k n_k\rangle,$$

and

$$\begin{aligned} m_k(x_k + n_k + t) &= f_k + \langle g_k, n_k + t\rangle + \frac{1}{2}\langle n_k + t, B_k(n_k + t)\rangle \\ &= m_k(x_k + n_k) + \langle g_k^n, t\rangle + \frac{1}{2}\langle t, B_k t\rangle, \end{aligned}$$

where $g_k^n = g_k + B_k n_k$.

Let Z_k be an orthonormal basis matrix of the null space of A_k if $\text{rank}(A_k) < n$. We assume t_k satisfies the following Cauchy-like condition

$$\delta_k^{f,t} \triangleq m_k(x_k + n_k) - m_k(x_k + n_k + t_k) \geq \kappa_f \chi_k \min\left(\frac{\chi_k}{1 + ||B_k||}, \Delta_k^f\right) \tag{2.6}$$

for some constant $\kappa_f \in (0, 1)$, where

$$\chi_k \triangleq ||Z_k^T g_k^n||. \tag{2.7}$$

Meanwhile we also require t_k does not increase the linearized infeasibility measure too much in the sense that

$$||c_k + A_k(n_k + t_k)||^2 \leq (1 - \kappa_t)||c_k||^2 + \kappa_t ||c_k + A_k n_k||^2 \tag{2.8}$$

for some constant $\kappa_t \in (0, 1)$. This condition on t_k can be satisfied if t_k is enforced to lie (approximately) on the null space of A_k. Achieving both of

(2.6) and (2.8) are quite reasonable since we can compute t_k as a sufficiently approximate solution to

$$\begin{cases} \min & \langle g_k^n, t\rangle + \frac{1}{2}\langle t, B_k t\rangle \\ \text{s.t.} & A_k t = 0, \\ & ||t|| \leq \Delta_k^f, \end{cases}$$

which is equivalent to

$$\begin{cases} \min & \langle Z_k^T g_k^n, v\rangle + \frac{1}{2}\langle v, Z_k^T B_k Z_k v\rangle \\ \text{s.t.} & ||v|| \leq \Delta_k^f. \end{cases}$$

The dogleg method or the CG-Steihaug method can therefore be applied [17]. When $\operatorname{rank}(A_k) = n$, we set $\chi_k = 0$ and $t_k = 0$.

After obtaining t_k, we define the complete step

$$s_k = n_k + t_k.$$

To obtain a relatively concise convergence analysis, we further impose that

$$||s_k|| \leq \Delta_k \triangleq \kappa_s \min(\Delta_k^c, \Delta_k^f) \tag{2.9}$$

for some sufficiently large constant $\kappa_s \geq 1$. In fact, (2.9) can be viewed as an assumption on the relativity of the sizes of Δ_k^c and Δ_k^f. It should be made clear that $\kappa_c, \kappa_f, \kappa_n, \kappa_t, \kappa_s$ are not chosen by users but theoretical constants. It also should be emphasized that the double trust regions approach applied here differs from that of Gould and Toint [15]. They do not compute t_k if n_k lies out of the ball $\{v : ||v|| \leq \kappa_B \min(\Delta_k^c, \Delta_k^f), \kappa_B \in (0, 1)\}$ and require the complete step $s_k = n_k + t_k$ lies within the ball $\{v : ||v|| \leq \min(\Delta_k^c, \Delta_k^f)\}$. In our approach, the sizes of n_k and t_k are more independent of each other, but a stronger assumption (2.9) is made. For more details about the differences see [15].

Now we consider the estimation of the Lagrange multiplier λ_{k+1}. We do not exactly compute

$$\lambda_{k+1} = -[A_k^T]^I g_k,$$

where the superscript ^{I} denotes the Moore-Penrose generalized inverse, but compute λ_{k+1} approximately solving the least-squares problem

$$\min_{\lambda} \frac{1}{2}||g_k + A_k^T \lambda||^2 \tag{2.10}$$

such that

$$||g_k + A_k^T \lambda|| \leq \tau_0 ||g_k|| \tag{2.11}$$

for some tolerance $\tau_0 > 0$.

2.2 Step acceptance. After computing the complete step s_k, we turn to face with the task of accepting or rejecting the trial point $x_k + s_k$.

We do not use a penalty function or a filter, but establish a new acceptance mechanism to promote global convergence. Let us now construct a dynamic finite set called h-**set**,

$$H_k = \{H_{k,1}, H_{k,2}, \cdots, H_{k,l}\},$$

where the l elements are sorted in a decreasing order, i.e., $H_{k,1} \geq H_{k,2} \geq \cdots \geq H_{k,l}$. The h-set is initialized to $H_0 = \{u, \cdots, u\}$ for some sufficiently large constant

$$u \geq \max(1, h(x_0)), \tag{2.12}$$

where x_0 is the starting point. We then consider the following three conditions:

- $h(x_k) = 0, \quad h(x_k + s_k) \leq H_{k,1};$ (2.13)
- $h(x_k) > 0, \quad h(x_k + s_k) \leq \beta h(x_k);$ (2.14)
- $h(x_k) > 0, \quad f(x_k + s_k) \leq f(x_k) - \gamma h(x_k + s_k),$

  $h(x_k + s_k) \leq \beta H_{k,2}.$ (2.15)

Here β, γ are two constants such that $0 < \gamma < \beta < 1$. Note that (2.14) and (2.15) imply

$$f(x_k + s_k) \leq f(x_k) - \gamma h(x_k + s_k) \ \text{ or } \ h(x_k + s_k) \leq \beta h(x_k). \tag{2.16}$$

After $x_{k+1} = x_k + s_k$ has been accepted as the next iterate, we may update the h-set with a new entry

$$h_k^+ \triangleq (1 - \theta) h(x_k) + \theta h(x_{k+1}), \quad \theta \in (0, 1). \tag{2.17}$$

This means we replace $H_{k,1}$ with h_k^+ and then re-sort the elements of h-set in a decreasing order. It is clear to see that the infeasibility measure of the iterates is controlled by the h-set, and that the length of the h-set l affects the strength of infeasibility control although only $H_{k,1}$ and $H_{k,2}$ are involved in conditions (2.13-2.15).

All iterations are classified into the following three types:

- f**-type.** At least one of (2.13-2.15) is satisfied and

$$\chi_k > 0, \quad \delta_k^f \triangleq f(x_k) - m_k(x_k + s_k) \geq \zeta \delta_k^{f,t}, \quad \zeta \in (0,1). \tag{2.18}$$

- h**-type.** At least one of (2.13-2.15) is satisfied but (2.18) fails.

- c**-type.** None of (2.13-2.15) is satisfied.

If k is an f-type iteration, we accept s_k and set $x_{k+1} = x_k + s_k$ if

$$\rho_k^f \triangleq \frac{f(x_k) - f(x_k + s_k)}{\delta_k^f} \geq \eta, \quad \eta \in (0,1). \tag{2.19}$$

Δ_k^f and Δ_k^c are updated according to

$$\Delta_{k+1}^f = \begin{cases} \min\left(\max(\tau_1 \Delta_k^f, \bar{\Delta}), \hat{\Delta}\right) & \text{if } \rho_k^f \geq \eta, \\ \tau_2 \Delta_k^f, & \text{if } \rho_k^f < \eta, \end{cases} \tag{2.20}$$

and

$$\Delta_{k+1}^c = \begin{cases} \max\left(\Delta_k^c, \bar{\Delta}\right) & \text{if } \rho_k^f \geq \eta, \\ \Delta_k^c, & \text{if } \rho_k^f < \eta. \end{cases} \tag{2.21}$$

If k is an h-type iteration, we always accept s_k and set $x_{k+1} = x_k + s_k$. Δ_k^f and Δ_k^c are updated according to

$$\Delta_{k+1}^f = \max(\Delta_k^f, \bar{\Delta}), \tag{2.22}$$

and

$$\Delta_{k+1}^c = \max(\Delta_k^c, \bar{\Delta}). \tag{2.23}$$

If k is a c-type iteration, we accept s_k and set $x_{k+1} = x_k + s_k$ if

$$\delta_k^c > 0, \quad \rho_k^c \triangleq \frac{h(x_k) - h(x_k + s_k)}{\delta_k^c} \geq \eta \tag{2.24}$$

where

$$\delta_k^c \triangleq \frac{1}{2}||c_k||^2 - \frac{1}{2}||c_k + A_k s_k||^2.$$

Δ_k^c and Δ_k^f are updated according to

$$\Delta_{k+1}^c = \begin{cases} \min\left(\max(\tau_1 \Delta_k^c, \bar{\Delta}), \hat{\Delta}\right) & \text{if } \rho_k^c \geq \eta, \\ \tau_2 \Delta_k^c, & \text{if } \rho_k^c < \eta,\ c_k \neq 0, \\ \Delta_k^c, & \text{if } \rho_k^c < \eta,\ c_k = 0. \end{cases} \tag{2.25}$$

and

$$\Delta_{k+1}^f = \begin{cases} \max\left(\Delta_k^f, \bar{\Delta}\right) & \text{if } \rho_k^c \geq \eta, \\ \tau_2 \Delta_k^f, & \text{if } \rho_k^c < \eta,\ c_k = 0, \\ \tau_2 \Delta_k^f, & \text{if } \rho_k^c < \eta,\ c_k \neq 0,\ \Delta_k^f > \bar{\Delta}, \\ \Delta_k^f, & \text{if } \rho_k^c < \eta,\ c_k \neq 0,\ \Delta_k^f \leq \bar{\Delta}. \end{cases} \tag{2.26}$$

The parameters in (2.20-2.23), (2.25) and (2.26), τ_1, τ_2, $\bar{\Delta}$, $\hat{\Delta}$, are some positive constants such that $\tau_2 < 1 \leq \tau_1$, and $\bar{\Delta} < \hat{\Delta}$.

One can easily make some conclusions from the update rules of the trust regions. Firstly, we observe that if k is successful, we have

$$\Delta_{k+1}^f \geq \bar{\Delta} \text{ and } \Delta_{k+1}^c \geq \bar{\Delta}. \tag{2.27}$$

Secondly, Δ_k^f is left unchanged on unsuccessful c-type iterations whenever x_k is infeasible and $\Delta_k^f \leq \bar{\Delta}$, and Δ_k^c is left unchanged on unsuccessful f-type iterations. Thirdly, Δ_k^f is reduced on unsuccessful f-type iterations and maybe reduced on unsuccessful c-type iterations, and Δ_k^c can only be reduced on unsuccessful c-type iterations. These properties are very crucial for our algorithm.

2.3 The algorithm. Now a formal statement of the algorithm is presented as follows.

Algorithm 1. A trust-region SQP algorithm without a penalty function or a filter.

Step 0: Initialize $k = 0$, $x_0 \in \mathbb{R}^n$, $B_0 \in S^{n\times n}$. Choose parameters Δ_0^c, Δ_0^f, $\bar{\Delta}$, $\hat{\Delta} \in (0, +\infty)$ that satisfy $\bar{\Delta} < \Delta_0^c, \Delta_0^f < \hat{\Delta}$, $\beta, \gamma, \theta, \zeta, \eta, \tau_2 \in (0, 1)$, $\tau_1, u \in [1, +\infty)$ and $l \in \{2, 3, \cdots\}$.

Step 1: If $k = 0$ or iteration $k - 1$ is successful, solve (2.10) for λ_{k+1}.

Step 2: Solve (2.2) for n_k that satisfies (2.3) and (2.4) if $c_k \neq 0$. Set $n_k = 0$ if $c_k = 0$.

Step 3: Compute t_k that satisfies (2.5), (2.6), (2.8) and (2.9) if $\mathrm{rank}(A_k) < n$. Set $t_k = 0$ if $\mathrm{rank}(A_k) = n$. Complete the trial step $s_k = n_k + t_k$.

Step 4: (f-type iteration) One of (2.13-2.15) is satisfied and (2.18) holds.

4.1: Accept $x_k + s_k$ if (2.19) holds.

4.2: Update Δ_k^f and Δ_k^c according to (2.20) and (2.21).

Step 5: (h-type iteration) One of (2.13-2.15) is satisfied but (2.18) fails.

5.1: Accept $x_k + s_k$.

5.2: Update Δ_k^f and Δ_k^c according to (2.22) and (2.23).

5.3: Update the h-set with h_k^+.

Step 6: (c-type iteration) None of (2.13-2.15) is satisfied.

6.1: Accept $x_k + s_k$ if (2.24) holds.

6.2: Update Δ_k^c and Δ_k^f according to (2.25) and (2.26).

6.3: Update the h-set with h_k^+ if $x_k + s_k$ is accepted.

Step 7: Accept the trial point. If $x_k + s_k$ has been accepted, set $x_{k+1} = x_k + s_k$, else set $x_{k+1} = x_k$.

Step 8: Update the Hessian. If $x_k + s_k$ has been accepted, choose a symmetric matrix B_{k+1}.

Step 9: Go to the next iteration. Increment k by one and go to Step 1.

Remarks. i) Conditions (2.3-2.6), (2.8) and (2.9) are some basic requirements for step computations. We assume they are satisfied for all iterations. ii) h-type iterations must be successful according to the mechanism of the algorithm. iii) The mechanism of the algorithm implies that the h-set H_k is updated only on h-type and successful c-type iterations. iv) Compared with the trust-cylinder of

DCI [1] and the trust-funnel [15], our h-set mechanism may be more flexible for controlling the infeasibility measure.

3 Global convergence

Before starting our global convergence analysis, we make some assumptions as follows.

Assumptions A

A1. Both f and c are twice differentiable.

A2. There exists a constant $\kappa_B \geq 1$ such that, $\forall\ \xi \in \bigcup_{k\geq 0}[x_k, x_k + s_k]$, $\forall\ k$, and $\forall\ i \in \{1, \cdots, m\}$,

$$1 + \max\left\{||g_k||, ||B_k||, ||A(\xi)||, ||\nabla^2 c_i(\xi)||, ||\nabla^2 f(\xi)||\right\} \leq \kappa_B. \quad (3.1)$$

A3. f is bounded below in the level set,

$$\mathcal{L} \triangleq \left\{x \in \mathbb{R}^n \mid h(x) \leq u\right\}. \quad (3.2)$$

A4. There exist two constants $\kappa_h, \kappa_\sigma > 0$ such that

$$h(x) \leq \kappa_h \implies \sigma_{\min}(A(x)) \geq \kappa_\sigma, \quad (3.3)$$

where $\sigma_{\min}(A)$ represents the smallest singular value of A.

In what follows we denote some useful index sets:

$$S \triangleq \left\{k \mid x_{k+1} = x_k + s_k\right\}$$

the set of successful iterations, $\mathcal{F}$, $\mathcal{H}$, and C, the sets of f-type, h-type, and c-type iterations.

The first two lemmas reveal some useful properties of the h-set. These properties play an important role in the following convergence analysis, particularly in driving the infeasibility measure to zero.

Lemma 1. *If $k \in S$ and x_k is a feasible point which is not a first order critical point, then k must be an f-type iteration and therefore the h-set is left unchanged in iteration k. Furthermore, each component of the h-set is strictly positive.*

Proof. Since x_k is feasible, $\delta_k^c = 0$ and therefore k cannot be a successful c-type iteration according to (2.24). Since x_k is a feasible point which is not a first order critical point, it follows $n_k = 0$, $\delta_k^f = \delta_k^{f,t}$ and (2.18) holds. Thus k must be an f-type iteration. Then, according to the mechanism of the algorithm, each component of H_k must be strictly positive. □

Lemma 2. *For all k, we have*

$$h(x_j) \leq H_{k,1} \leq u, \quad \forall\, j \geq k, \tag{3.4}$$

and $H_{k,1}$ is monotonically decreasing in k.

Proof. Without loss of generality, we can assume that all iterations are successful. We first prove the following inequality

$$h(x_k) \leq H_{k,1} \tag{3.5}$$

by induction. According to (2.12), we immediately have that (3.5) is true for $k = 0$. For $k \geq 1$, we consider the following three cases.

The first case is that $k - 1 \in \mathcal{F}$. Then one of (2.13-2.15) holds and therefore, according to the hypothesis $h(x_{k-1}) \leq H_{k-1,1}$, we have from (2.13-2.15) that

$$h(x_k) \leq \max(H_{k-1,1}, \beta h(x_{k-1}), \beta H_{k-1,2}) = H_{k-1,1}.$$

Since the h-set is not updated on an f-type iteration, we have $H_{k,1} = H_{k-1,1}$. Thus (3.5) follows.

The second case is that $k - 1 \in \mathcal{H}$. Lemma 1 implies that x_{k-1} is an infeasible point. Then one of (2.14) and (2.15) holds and H_{k-1} is updated with h_{k-1}^+. It follows from condition (2.14) or (2.15) that

$$h(x_k) \leq \beta \max(h(x_{k-1}), H_{k-1,2}).$$

Therefore the update rules of the h-set, together with (2.17), implies that (3.5) holds.

The third case is that $k - 1 \in C$. Then, according to (2.17) and (2.24), we have $h(x_k) < h_{k-1}^+$. Since H_{k-1} is updated with h_{k-1}^+, it follows $h_{k-1}^+ \leq H_{k,1}$. Hence we obtain (3.5) from the above two inequalities.

Since $\max(h(x_{k+1}), h(x_k)) \leq H_{k,1}$ we have $h_k^+ \leq H_{k,1}$ from (2.17). Thus the monotonic decrease of $H_{k,1}$ follows. Finally, (3.4) follows immediately from (3.5) and the monotonic decrease of $H_{k,1}$. □

We now verify that our algorithm satisfies a Cauchy-like condition on the predicted reduction in the infeasibility measure.

Lemma 3. *For all k, we have that*

$$\delta_k^c \geq \kappa_t \kappa_c ||A_k^T c_k|| \min\left(\frac{||A_k^T c_k||}{1 + ||A_k^T A_k||}, \Delta_k^c\right). \tag{3.6}$$

Proof. It follows from (2.3) and (2.8) that

$$\begin{aligned}
\delta_k^c &= \frac{1}{2}||c_k||^2 - \frac{1}{2}||c_k + A_k s_k||^2 \\
&\geq \frac{1}{2}||c_k||^2 - \frac{1}{2}(1-\kappa_t)||c_k||^2 - \frac{1}{2}\kappa_t ||c_k + A_k n_k||^2 \\
&= \frac{1}{2}\kappa_t(||c_k||^2 - ||c_k + A_k n_k||^2) \\
&\geq \kappa_t \kappa_c ||A_k^T c_k|| \min\left(\frac{||A_k^T c_k||}{1 + ||A_k^T A_k||}, \Delta_k^c\right).
\end{aligned}$$

□

The following lemma is a direct result of (3.1).

Lemma 4. *For all k, we have that*

$$1 + ||A_k^T A_k|| \leq \kappa_B^2. \tag{3.7}$$

Proof. The proof is identical to that of the first part of Lemma 3.1 of [15]. □

The following lemma is a direct result of Taylor's theorem.

Lemma 5. *For all k, we have that*

$$|f(x_k + s_k) - m_k(x_k + s_k)| \leq \kappa_B ||s_k||^2, \tag{3.8}$$

and

$$|\ ||c(x_k + s_k)||^2 - ||c_k + A_k s_k||^2\ | \leq 2\kappa_C ||s_k||^2, \tag{3.9}$$

where $\kappa_C > \kappa_B^2$ is a constant.

Proof. The proof is similar to that of Lemma 3.4 of [15]. □

The following lemma is very important as for most of trust-region methods.

Lemma 6. *Suppose that $k \in \mathcal{F}$ and that*

$$\Delta_k^f \leq \frac{(1-\eta)\zeta\kappa_f\chi_k}{\kappa_B\kappa_s^2}. \tag{3.10}$$

Then $\rho_k^f > \eta$. Similarly, suppose that $k \in C$, $c_k \neq 0$, and

$$\Delta_k^c \leq \frac{(1-\eta)\kappa_t\kappa_c||A_k^T c_k||}{\kappa_C\kappa_s^2}. \tag{3.11}$$

Then $\rho_k^c > \eta$.

Proof. The proof of both statements is similar to that of Theorem 6.4.2 of [5]. In fact, using (2.6), (2.18) and (3.1), we have

$$\delta_k^f \geq \zeta\kappa_f\chi_k \min\left(\frac{\chi_k}{1+||B_k||}, \Delta_k^f\right) \geq \zeta\kappa_f\chi_k \min\left(\frac{\chi_k}{\kappa_B}, \Delta_k^f\right).$$

Then it follows from (2.9) and (3.8) that if (3.10) holds then

$$\begin{aligned} |1-\rho_k^f| &= \left|\frac{f(x_k+s_k)-m_k(x_k+s_k)}{\delta_k^f}\right| \\ &\leq \frac{\kappa_B||s_k||^2}{\zeta\kappa_f\chi_k\min\left(\frac{\chi_k}{\kappa_B}, \Delta_k^f\right)} \\ &\leq \frac{\kappa_B\kappa_s^2(\Delta_k^f)^2}{\zeta\kappa_f\chi_k\Delta_k^f} \\ &\leq 1-\eta. \end{aligned}$$

Hence, the first conclusion follows. Similarly, we use (2.9), (3.6), (3.7) and (3.9) to obtain the second conclusion. □

We now verify below that our algorithm can eventually take a real iteration at any point which is not an infeasible stationary point of $h(x)$. We recall beforehand the definition of an infeasibility stationary point of $h(x)$.

Definiton 1. *We call $\hat{x}$ an infeasible stationary point of $h(x)$ if $\hat{x}$ satisfies*

$$A(\hat{x})^T c(\hat{x}) = 0 \quad and \quad c(\hat{x}) \neq 0.$$

The algorithm will fail to progress towards the feasible region if started from an infeasible stationary point since no suitable normal step can be found in this situation. If such a point is detected, restarting the whole algorithm from a different point might be the best strategy.

Lemma 7. *Suppose that first order critical points and infeasible stationary points never occur. Then we have that $|S| = +\infty$.*

Proof. Since $x_k + s_k$ must be accepted if k is an h-type iteration, we only consider $k \in \mathcal{F} \cup C$. First consider the case that x_k is infeasible. Since the assumption that x_k is not an infeasible stationary point implies $||A_k^T c_k|| > 0$, it follows from (3.6) that $\delta_k^c > 0$ and from Lemma 6 that $\rho_k^c \geq \eta$ for sufficiently small Δ_k^c. It also follows from Lemma 6 that $\rho_k^f \geq \eta$ for sufficiently small Δ_k^f if $\chi_k > 0$. Note that $k \in \mathcal{F} \backslash S$ implies $\chi_k > 0$, $\chi_{k+1} = \chi_k$, and $\Delta_{k+1}^f = \tau_2 \Delta_k^f$, and $k \in C \backslash S$ implies $\Delta_{k+1}^c = \tau_2 \Delta_k^c$. Therefore, a successful iteration must be finished at x_k in the end.

Next we consider the case that x_k is feasible. Since x_k is not a first order critical point we have $\chi_k > 0$. Then it follows from Lemma 6 that $\rho_k^f \geq \eta$ for sufficiently small Δ_k^f. Furthermore, (2.13) must be satisfied if $\Delta_k^f \leq \sqrt{\frac{H_{k,1}}{\kappa_C}}$. Because, according to (3.9) and the fact that $c_k + A_k s_k = 0$ when $c_k = 0$ implied by (2.8), we have $h(x_k + s_k) \leq \kappa_C ||s_k||^2 \leq H_{k,1}$. Hence a successful iteration must be finished at x_k in the end. □

The following lemma is a crucial result of the mechanism of the algorithm.

Lemma 8. *Suppose that, for some $\epsilon_f > 0$,*

$$\chi_k \geq \epsilon_f. \tag{3.12}$$

Then

$$\Delta_k^f \geq \min\left(\mu_f, \tau_2 \sqrt{\frac{H_{k,1}}{\kappa_C}}\right), \tag{3.13}$$

where

$$\mu_f \triangleq \tau_2 \min\left(\frac{(1-\eta)\zeta\kappa_f\epsilon_f}{\kappa_B\kappa_s^2}, \bar{\Delta}\right).$$

Moreover, (3.13) *can be reduced to* $\Delta_k^f \geq \mu_f$ *if* x_k *is infeasible. Similarly, suppose that, for some* $\epsilon_c > 0$,

$$||A_k^T c_k|| \geq \epsilon_c. \tag{3.14}$$

Then,

$$\Delta_k^c \geq \mu_c \triangleq \min\left(\frac{\tau_2(1-\eta)\kappa_t\kappa_c\epsilon_c}{\kappa_C\kappa_s^2}, \bar{\Delta}\right). \tag{3.15}$$

Proof. The two statements are proved in the same manner, and immediately result from (2.27), Lemma 6, the proof of Lemma 7 and the update rules of the the trust-region radii. □

Now we consider the global convergence property of our algorithm in the case that successful c-type and h-type iterations are finitely many.

Lemma 9. *Suppose that* $|S| = +\infty$ *and that* $|(\mathcal{H} \cup C) \cap S| < +\infty$. *Then*

$$\lim_{k\to\infty, k\in S} \chi_k = 0, \tag{3.16}$$

and there exists an infinite subsequence $\mathcal{K} \subset S$ *such that*

$$\lim_{k\to\infty, k\in\mathcal{K}} h(x_k) = 0. \tag{3.17}$$

Proof. Since all successful iterations must be f-type for sufficiently large k, we can deduce from (2.18) and (2.19) that $f(x_k)$ is monotonically decreasing in k for all sufficiently large k. For the purpose of deriving a contradiction, we assume that (3.12) holds for an infinite subsequence $\mathcal{K} \subset S$. Then (2.6), (2.18), (3.1) and (3.13) together yield that, for sufficiently large $k \in \mathcal{K}$,

$$\begin{aligned}\delta_k^f &\geq \zeta\kappa_f\chi_k \min\left(\frac{\chi_k}{1+||B_k||}, \Delta_k^f\right)\\ &\geq \zeta\kappa_f\epsilon_f \min\left(\frac{\epsilon_f}{\kappa_B}, \mu_f, \tau_2\sqrt{\frac{H_{k,1}}{\kappa_C}}\right).\end{aligned}$$

Then we have from (2.19) and the above inequality that, for sufficiently large $k \in \mathcal{K}$,

$$f(x_k) - f(x_{k+1}) \geq \eta \delta_k^f \geq \eta \zeta \kappa_f \epsilon_f \min\left(\frac{\epsilon_f}{\kappa_B}, \mu_f, \tau_2 \sqrt{\frac{H_{k,1}}{\kappa_C}}\right).$$

Since the assumption of the lemma implies that the h-set is updated for finitely many times, we have that $H_{k,1}$ is a constant for all sufficiently large k. This, together with the monotonic decrease of $f(x_k)$, implies that $\lim_{k\to\infty} f(x_k) = -\infty$. Since Lemma 2 implies $\{x_k\}$ is contained in the level set $\mathcal{L}$ defined by (3.2), the below unboundedness of $f(x_k)$ contradicts the assumption A3. Hence (3.12) is impossible and (3.16) follows.

Now consider (3.17). Assume that x_k is infeasible for all sufficiently large k; otherwise (3.17) follows immediately for some infinite subsequence $\mathcal{K} \subset \mathcal{S}$. Then it follows from the monotonic decrease of $f(x_k)$, (2.16), and Lemma 1 of [11] that $\lim_{k\to\infty} h(x_k) = 0$, which also derives (3.17). □

Next we verify that the constraint function must converge to zero in the case that h-type iterations are infinitely many.

Lemma 10. *Suppose that* $|\mathcal{H}| = +\infty$. *Then* $\lim_{k\to\infty} h(x_k) = 0$.

Proof. Denote $\mathcal{H} = \{k_i\}$. Recalling that at least one of (2.13-2.15) holds on h-type iterations and that x_{k_i} is infeasible by Lemma 1, we deduce from (2.14), (2.15), (2.17) and (3.4) that

$$\begin{aligned} h_{k_i}^+ &= (1-\theta)h(x_{k_i}) + \theta h(x_{k_i+1}) \\ &\leq (1-\theta)H_{k_i,1} + \theta\beta \max(H_{k_i,2}, h(x_{k_i})) \\ &\leq (1-\theta+\theta\beta)H_{k_i,1}. \end{aligned}$$

It then follows from the mechanism of the h-set that

$$H_{k_{i+l},1} \leq (1-\theta+\beta\theta)H_{k_i,1}.$$

Hence, from the above inequality and the monotonic decrease of $H_{k,1}$, we have that

$$\lim_{k\to\infty} H_{k,1} = 0. \tag{3.18}$$

Thus, from (3.4) and (3.18), the result follows. □

In what follows, to obtain global convergence, we will exclude a scenario for successful c-type iterations which is less unlikely than being trapped into a local infeasible stationary point. This scenario is

$$\lim_{k\to\infty, k\in C\cap S} ||A_k^T c_k|| = 0 \text{ with } \liminf_{k\to\infty, k\in C\cap S} ||c_k|| > 0. \tag{3.19}$$

We now verify below that the constraints also converges to zeros in the case that successful c-type iterations are infinitely many provided that the above undesirable situation is avoided.

Lemma 11. *Suppose that* $|C \cap S| = +\infty$ *and that* (3.19) *is avoided. Then* $\lim_{k\to\infty} h(x_k) = 0$.

Proof. We first prove that

$$\lim_{k\to\infty, k\in C\cap S} ||A_k^T c_k|| = 0. \tag{3.20}$$

Assume, for the purpose of deriving a contradiction, that (3.14) holds for some infinite subsequence indexed by $\mathcal{K} \subset C \cap S$. Recall that the h-set is updated on successful c-type iterations and denote $\{k_i\} = \mathcal{K}$. It then follows from (2.17), (2.24), (3.4), (3.6), (3.7), (3.14) and (3.15) that

$$\begin{aligned}
H_{k_i,1} - h_{k_i}^+ &\geq h(x_{k_i}) - h_{k_i}^+ \\
&= \theta(h(x_{k_i}) - h(x_{k_i+1})) \\
&\geq \theta\eta\delta_{k_i}^c \\
&\geq \theta\eta\kappa_t\kappa_c ||A_{k_i}^T c_{k_i}|| \min\left(\frac{||A_{k_i}^T c_{k_i}||}{1 + ||A_{k_i}^T A_{k_i}||}, \Delta_{k_i}^c\right) \\
&\geq \epsilon_h \triangleq \theta\eta\kappa_t\kappa_c\epsilon_c \min\left(\frac{\epsilon_c}{\kappa_B^2}, \mu_c\right).
\end{aligned}$$

It then follows from the above inequality, the monotonic decrease of $H_{k,1}$ and the mechanism of the h-set that

$$H_{k_i,1} - H_{k_{i+l,1}} \geq \epsilon_h.$$

This, together with $|\mathcal{K}| = +\infty$, yields that $H_{k,1}$ is unbounded below, which is impossible. Hence (3.20) holds.

Since (3.19) does not hold, it follows from (3.20) that

$$\liminf_{k\to\infty, k\in C\cap S} ||c_k|| = 0.$$

Thus, there exists an infinite subsequence indexed by $\mathcal{J} \subseteq C \cap S$ such that

$$\lim_{k\to\infty, k\in\mathcal{J}} ||c_k|| = 0.$$

Since $h(x_{k+1}) \leq h(x_k)$ for all $k \in C \cap S$, the above limit implies

$$\lim_{k\to\infty, k\in\mathcal{J}} h_k^+ = 0$$

by (2.17). Then (3.18) follows from the facts that the h-set is updated on successful c-type iterations and that $H_{k,1}$ is monotonically decreasing. Therefore the result follows immediately from (3.4) and (3.18). □

In what follows, we give the global convergence property of our algorithm in the case that successful c-type and h-type iterations are infinitely many.

Lemma 12. *Suppose that* $|(\mathcal{H} \cup C) \cap S| = +\infty$ *and that* (3.19) *is avoided. Then*

$$\lim_{k\to\infty} h(x_k) = 0 \tag{3.21}$$

and if β *is sufficiently close to 1, we have*

$$\liminf_{k\to\infty} \chi_k = 0. \tag{3.22}$$

Proof. Limit (3.21) follows immediately from Lemmas 10 and 11. Then we consider (3.22). It follows from (2.4) and (3.21) that

$$\lim_{k\to\infty} n_k = 0. \tag{3.23}$$

Therefore, from (3.1) and (3.23), we have

$$\lim_{k\to\infty} \delta_k^{f,n} = 0, \tag{3.24}$$

where

$$\delta_k^{f,n} \triangleq f(x_k) - m_k(x_k + n_k) = -g_k^T n_k - \frac{1}{2} n_k^T B_k n_k.$$

Assume now again, for the purpose of deriving a contradiction, that (3.12) holds for all k sufficiently large. Then, if x_k is infeasible, we have from (2.6), (3.1), and Lemma 8 that

$$\begin{aligned}\delta_k^{f,t} &\geq \kappa_f \chi_k \min\left(\frac{\chi_k}{1+||B_k||}, \Delta_k^f\right)\\ &\geq \epsilon_t \triangleq \kappa_f \epsilon_f \min\left(\frac{\epsilon_f}{\kappa_B}, \mu_f\right)\end{aligned}$$

for all k sufficiently large. It then follows from (3.24) that, for all sufficiently large k,

$$|\delta_k^{f,n}| \leq (1-\zeta)\epsilon_t. \tag{3.25}$$

It is easy to see that, for all sufficiently large k such that (3.25) holds, we have

$$\frac{\delta_k^f}{\delta_k^{f,t}} = 1 + \frac{\delta_k^{f,n}}{\delta_k^{f,t}} \geq 1-(1-\zeta) = \zeta \tag{3.26}$$

and therefore (2.18) holds. If x_k is feasible, then $n_k = 0$ and therefore (2.18) must hold. Thus k cannot be an h-type iteration for all sufficiently large k.

Now consider any sufficiently large $k \in (\mathcal{H} \cup C) \cap S$ so that (3.25) holds and

$$h(x_k) \leq \kappa_h. \tag{3.27}$$

Since (3.25) holds, we have $k \in C \cap S$ by the above analysis. Note that Lemma 1 implies that x_k is infeasible. It follows from (3.3), (3.6), (3.7) and (3.27) that

$$\begin{aligned}\delta_k^c &\geq \kappa_t\kappa_c||A_k^T c_k|| \min\left(\frac{||A_k^T c_k||}{1+||A_k^T A_k||}, \Delta_k^c\right)\\ &\geq \kappa_t\kappa_c\kappa_\sigma||c_k|| \min\left(\frac{\kappa_\sigma||c_k||}{\kappa_B^2}, \Delta_k^c\right).\end{aligned} \tag{3.28}$$

Reasoning as in the proof of (3.15) in Lemma 8, one can conclude that

$$\Delta_k^c \geq \min\left(\frac{\tau_2(1-\eta)\kappa_t\kappa_c\kappa_\sigma||c_k||}{\kappa_C\kappa_s^2}, \bar{\Delta}\right).$$

This, together with $\lim_{k\to\infty} c_k = 0$, implies that

$$\Delta_k^c \geq \frac{\tau_2(1-\eta)\kappa_t\kappa_c\kappa_\sigma||c_k||}{\kappa_C\kappa_s^2} \tag{3.29}$$

for all sufficiently large k. Then (3.28) and (3.29) yield

$$\begin{aligned}
\delta_k^c &\geq \kappa_t \kappa_c \kappa_\sigma ||c_k|| \min\left(\frac{\kappa_\sigma ||c_k||}{\kappa_B^2}, \frac{\tau_2(1-\eta)\kappa_t\kappa_c\kappa_\sigma ||c_k||}{\kappa_C \kappa_s^2}\right) \\
&\geq \kappa_t \kappa_c \kappa_\sigma \min\left(\frac{\kappa_\sigma}{\kappa_B^2}, \frac{\tau_2(1-\eta)\kappa_t\kappa_c\kappa_\sigma}{\kappa_C \kappa_s^2}\right) ||c_k||^2 \\
&\geq \frac{\tau_2(1-\eta)\kappa_t^2\kappa_c^2\kappa_\sigma^2}{\kappa_C \kappa_s^2} ||c_k||^2 \\
&= \kappa_\beta h(x_k),
\end{aligned} \tag{3.30}$$

where

$$\kappa_\beta \triangleq \frac{2\tau_2(1-\eta)\kappa_t^2\kappa_c^2\kappa_\sigma^2}{\kappa_C \kappa_s^2}.$$

Since $k \in C \cap S$, we have from (2.24) and (3.30) that

$$h(x_{k+1}) \leq h(x_k) - \eta \delta_k^c \leq (1 - \eta\kappa_\beta) h(x_k).$$

Then the above inequality implies that if $\beta \in (0, 1)$ is sufficiently close to 1, more specifically, if

$$1 - \eta\kappa_\beta \leq \beta < 1,$$

it follows

$$h(x_{k+1}) \leq \beta h(x_k).$$

Then x_{k+1} satisfies condition (2.14) and therefore k cannot be a c-type iteration, which contradicts $k \in C$. Hence (3.22) holds and the proof is now completed. □

We now present the main theorem on the basis of all the results obtained above.

Theorem 1. *Suppose that first order critical points and infeasible stationary points never occur and that* (3.19) *is avoided. Then there exists a subsequence indexed by $\mathcal{K}$ such that*

$$\lim_{k\to\infty, k\in\mathcal{K}} c_k = 0,$$

and if β is sufficiently close to 1,

$$\lim_{k\to\infty, k\in\mathcal{K}} Z_k^T g_k = 0.$$

As a consequence, if β is sufficiently close to 1, any accumulation point of the sequence $\{x_k\}_{k\in\mathcal{K}}$ is a first order critical point.

Proof. It is easy to see that if

$$\lim_{k\to\infty, k\in\mathcal{K}} c_k = 0,$$

then we have from (2.4), (2.7) and (3.1) that

$$\lim_{k\to\infty, k\in\mathcal{K}} \chi_k = \lim_{k\to\infty, k\in\mathcal{K}} Z_k^T(g_k + B_k n_k) = \lim_{k\to\infty, k\in\mathcal{K}} Z_k^T g_k.$$

This means χ_k defined by (2.7) is actually an optimality measure for first-order critical points. Then the desired conclusions immediately follow from Lemmas 7, 9 and 12. □

4 Numerical results

In this section, we present some numerical results for some small size examples to demonstrate our new method may be promising. All the experiments were run in MATLAB R2009b. Details about the implementation are described as follows.

We initialized the approximate Hessian to the identity matrix $B_0 = I$ and updated B_k by Powell's damped BFGS formula [22]. The dogleg method was applied to compute both normal steps and tangential steps. Moreover, each tangential step was found in the null space of the Jacobian. We computed the Lagrangian multiplier by using MATLAB's lsqlin function. The parameters for Algorithm 1 were chosen as:

$$\beta = 0.9999,\ \gamma = \theta = \zeta = \eta = 10^{-4},$$

$$\tau_1 = 1.1,\ \tau_2 = 0.5,\ l = 3,\ u = \max(500, 1.5h(x_0)),$$

$$\Delta_0^c = 0.5\max(||x_0||_2, \sqrt{n}),\ \Delta_0^f = 1.2\Delta_0^c,\ \hat{\Delta} = 10\Delta_0^c,\ \bar{\Delta} = 10^{-4}.$$

Now we compare the performance of Algorithm 1 with that of SNOPT Version 5.3 [12] based on the numbers of function and gradient evaluations required to achieve convergence. A standard stopping criterion is used for Algorithm 1, i.e.,

$$||c_k||_\infty \le 10^{-6}(1 + ||x_k||_2),$$

and

$$||g_k + A_k^T \lambda_{k+1}||_\infty \le 10^{-6}(1 + ||\lambda_{k+1}||_2).$$

The test problems here are all the equality constrained problems from [16]. We ran SNOPT under default options on the NEOS Server (http://www.neos-server.org/neos/solvers/nco:SNOPT/AMPL.html). The corresponding results are shown in Table 1, where Nit, Nf, and Ng represent the numbers of successful iterations, of function evaluations and of gradient evaluations, respectively. It can be observed from Table 1 that Algorithm 1 is generally superior to SNOPT for these problems.

Problem			Algorithm 1			SNOPT		
Name	n	m	Nit	Nf	Ng	Nit	Nf	Ng
hs06	2	1	7	8	8	5	9	8
hs07	2	1	10	11	11	18	31	30
hs08	2	2	6	7	7	2	7	6
hs09	2	1	8	9	9	9	9	8
hs26	3	1	19	20	20	25	25	24
hs27	3	1	18	21	19	22	24	23
hs28	3	1	6	7	7	4	4	4
hs39	4	2	18	22	19	20	31	30
hs40	4	3	6	7	7	7	8	7
hs42	4	2	8	9	9	8	9	8
hs46	5	2	29	32	30	28	27	26
hs47	5	3	21	24	22	24	32	31
hs48	5	2	8	12	9	6	6	6
hs49	5	2	22	24	23	37	33	32
hs50	5	3	13	15	14	31	22	21
hs51	5	3	6	8	7	6	6	6
hs52	5	3	8	9	9	5	5	5
hs56	7	4	11	12	12	13	15	14
hs61	3	2	9	10	10	69	169	168
hs77	5	2	11	15	12	15	15	14
hs78	5	3	7	8	8	9	8	7
hs79	5	3	8	11	9	14	15	14

Table 1 – Numerical results.

We also plot the logarithmic performance profiles proposed by Dolan and

Moré [7] in Figure 1. In the plots, the performance profile is defined by

$$\pi_s(\tau) \triangleq \frac{\text{no. of problems where } \log_2(r_{p,s}) \leq \tau}{\text{total no. of problems}}, \quad \tau \geq 0,$$

where $r_{p,s}$ is the ratio of Nf or Ng required to solve problem p by solver s and the lowest value of Nf or Ng required by any solver on this problem. The ratio $r_{p,s}$ is set to infinity whenever solver s fails to solve problem p. It can be observed from Figure 1 that Algorithm 1 outperforms SNOPT for these problems.

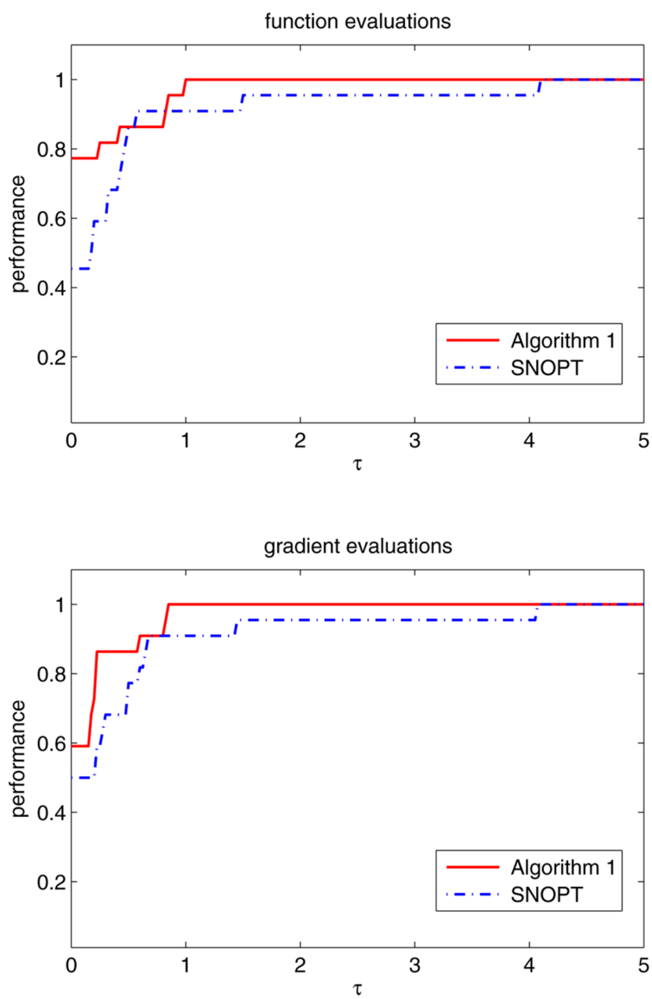

Figure 1 – Performance profiles.

5 Conclusions

In this paper, a new double trust regions sequential quadratic programming method for solving equality constrained optimization is presented. Each trial step is computed using a double trust regions strategy in two phases, the first of which aims feasibility and the second, optimality. Thus, the approach is similar to inexact restoration methods for nonlinear programming. The most important feature of this paper is to prove global convergence without using a penalty function or a filter. We propose a new step acceptance technique, the h-set mechanism, which is quite different from Gould and Toint's trust-funnel and Bielschowsky and Gomes' trust cylinder. Numerical results demonstrate the efficiency of this new approach.

REFERENCES

[1] R.H. Bielschowsky and F.A.M. Gomes, *Dynamic control of infeasibility in equalitly constrained optimization.* SIAM J. Optim., **19** (2008), 1299–1325.

[2] E.G. Birgin and J.M. Martínez, *Local convergence of an inexact restoration method and numerical experiments.* J. Optim. Theory Appl., **127** (2005), 229–247.

[3] R.H. Byrd, R.B. Schnabel and G.A. Shultz, *A trust region algorithm for nonlinearly constrained optimization.* SIAM J. Numer. Anal., **24** (1987), 1152–1170.

[4] C.M. Chin and R. Fletcher, *On the global convergence of an SLP-filter algorithm that takes EQP steps.* Math. Program., **96** (2003), 161–177.

[5] A.R. Conn, N.I.M. Gould and P.L. Toint, *Trust Region Methods, No. 01 in MPS-SIAM Series on Optimization.* SIAM, Philadelphia, USA (2000).

[6] J.E. Dennis, M. El-Alem and M.C. Maciel, *A global convergence theory for general trust-region-based algorithm for equality constrained optimization.* SIAM J. Optim., **7** (1997), 177–207.

[7] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles.* Math. Program., **91** (2002), 201–213.

[8] A. Fischer and A. Friedlander, *A new line search inexact restoration approach for nonlinear programming.* Comput. Optim. Appl., **46** (2010), 333–346.

[9] R. Fletcher, N. Gould, S. Leyffer, P.L. Toint and A. Wächter, *Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming.* SIAM J. Optim. **13** (2002), 635–659.

[10] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function.* Math. Program., **91** (2002), 239–269.

[11] R. Fletcher, S. Leyffer and P.L. Toint, *On the global convergence of a filter-SQP algorithm.* SIAM J. Optim., **13** (2002), 44–59.

[12] P.E. Gill, W. Murray and M.A. Saunders, *SNOPT: an SQP algorithm for large-scale constrained optimization.* SIAM Rev., **47** (2005), 99–131.

[13] F.M. Gomes, M.C. Maciel and J.M. Martínez, *Nonlinear programming algorithms using trust regions and augmented lagrangians with nonmonotone penalty parameters.* Math. Program., **84** (1999), 161–200.

[14] C.C. Gonzaga, E.W. Karas and M. Vanti, *A globally convergent filter method for nonlinear programming.* SIAM J. Optim., **14** (2003), 646–669.

[15] N.I.M. Gould and P.L. Toint, *Nolinear programming without a penalty function or a filter.* Math. Program., **122** (2010), 155–196.

[16] W. Hock and K. Schittkowski, Test examples for nonlinear programming codes, Springer-Verlag (1981).

[17] M. Lalee, J. Nocedal and T.D. Plantenga, *On the implementation of an algorithm for large-scale equality constrained optimization.* SIAM J. Optim., **8** (1998), 682–706.

[18] J.M. Martínez, *Two-phase model algorithm with global convergence for nonlinear programming.* J. Optim. Theory Appl., **96** (1998), 397–436.

[19] J.M. Martínez, *Inexact restoration method with lagriangian tangent decrease and new merit function for nonlinear programming.* J. Optim. Theory Appl., **111** (2001), 39–58.

[20] J.M. Martínez and E.A. Pilotta, *Inexact restoration algorithms for constrained optimization.* J. Optim. Theory Appl., **104** (2000), 135–163.

[21] E.O. Omojokun, *Trust region algorithms for optimization with nonlinear equality and inequality constraints.* Ph.D. thesis, Dept. of Computer Science, University of Colorado, Boulder (1989).

[22] M.J.D. Powell, *A fast algorithm for nonlinearly constrained optimization calculations.* In Numerical Analysis Dundee 1977, G.A. Watson, (ed.), Springer-Verlag, Berlin, (1978), 144–157.

[23] M.J.D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization.* Math. Program., **49** (1991), 189–211.

[24] A.A. Ribeiro, E.W. Karas and C.C. Gonzaga, *Global convergence of filter methods for nonlinear programming.* SIAM J. Optim., **19** (2008), 1231–1249.

[25] C. Shen, W. Xue and D. Pu, *A filter SQP algorithm without a feasibility restoration phase.* Comput. Appl. Math., **28** (2009), 167–194.

[26] S. Ulbrich, *On the superlinear local convergence of a filter-SQP method.* Math. Program., **100** (2004), 217–245.

[27] A. Wächter and L.T. Biegler, *Line search filter methods for nonlinear programming: local convergence.* SIAM J. Optim., **16** (2005), 32–48.

[28] A. Wächter and L.T. Biegler, *Line search filter methods for nonlinear programming: motivation and global convergence.* SIAM J. Optim., **16** (2005), 1–31.