A New Convergence Theory for Trust-Region Algorithm for Solving Constrained Optimization Problems

Bothina EL-Sobky

Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt

Copyright © 2013 Bothina EL-Sobky. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In this paper, we propose a new trust-region algorithm for solving a constrained optimization problem with equality and inequality constraints. In this algorithm, an active-set technique is used to convert the constrained optimization problem with equality and inequality constraints to equality constrained optimization problem. A projected Hessian technique is used together with a conjugate gradient method to compute the trial step. Global convergence results are established under important assumptions and it is shown that a subsequence of the iteration sequence is not bounded away from KKT points. Preliminary numerical experiment on the algorithm is presented. The performance of the algorithm is reported. The numerical results show that our approach is of value and merit further investigation.

Keywords: Constrained optimization, active set, trust region, projected Hessian, conjugate gradient, global convergence

1 Introduction

Consider the following constrained optimization problem

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad c_i(x) = 0 \quad i \in E, \\
& \quad c_i(x) \leq 0 \quad i \in I,
\end{align*}
\]

(1.1)

where \( f : \mathbb{R}^n \to \mathbb{R}, c_i : \mathbb{R}^n \to \mathbb{R}, E \cup I = \{1, \ldots, m\} \) and \( E \cap I = \emptyset \). Let \( C_E(x) : \mathbb{R}^n \to \mathbb{R}^{|E|} \) be the vector function whose components are \( c_i(x) \) for \( i \in E \), and \( C_I(x) : \mathbb{R}^n \to \mathbb{R}^{|I|} \) to be the
vector function whose components are \( c_i(x) \) for \( i \in I \). Then Problem (1.1) can be written as

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad C_E(x) = 0, \\
& \quad C_I(x) \leq 0.
\end{align*}
\]  

(1.2)

The Lagrangian function \( l : \mathbb{R}^n \times \mathbb{R}^{|E|} \times \mathbb{R}^{|I|} \rightarrow \mathbb{R} \) associated with Problem (1.2) is the function

\[
l(x, \mu, \nu) = f(x) + \mu^T C_E(x) + \nu^T C_I(x),
\]

(1.3)

where \( \mu = (\mu_1, ..., \mu_{|E|})^T \) and \( \nu = (\nu_1, ..., \nu_{|I|})^T \) are the Lagrange multiplier vectors associated with the equality and inequality constraints respectively. Let \( J(x) \) be the set of indices of violated or binding inequality constraints at a point \( x \). i.e., \( J(x) = \{ i : c_i(x) \leq 0, i \in I \} \). A point \( x_* \) is a regular point for Problem (1.2) if the vectors in the set \( \{ \nabla c_i(x_*), i = 1, 2, ..., |E| \} \cup \{ \nabla c_i(x_*), i \in J(x_*) \} \) are linearly independent. Let \( x_* \) be a regular point. The first-order necessary conditions (or the KKT conditions) for the point \( x_* \) to be a stationary point of Problem (1.2) are the existence of Lagrange multipliers \( \mu_* \in \mathbb{R}^{|E|} \) and \( \nu_* \in \mathbb{R}^{|I|} \) such that

\[
\begin{align*}
\nabla f(x_*) + \nabla C_E(x_*)\mu_* + \nabla C_I(x_*)\nu_* &= 0, \\
C_E(x_*) &= 0, \\
C_I(x_*) &\leq 0, \\
(\nu_*)_i c_i(x_*) &= 0, \quad i = 1, ..., |I| \\
(\nu_*)_i &\geq 0, \quad i = 1, ..., |I|.
\end{align*}
\]

(1.4) - (1.8)

Here we used the notations \( \nabla C_E(x) \) and \( \nabla C_I(x) \) for the matrices whose columns are \( \nabla c_i(x) \), \( i = 1, ..., |E| \) and \( \nabla c_i(x) \), \( i = 1, ..., |I| \), respectively. For a detailed discussion of optimality conditions, see Fiacco and McCormick [12]. Over the last four decades, trust-region algorithms have proven to be robust techniques for solving optimization problems. Their high regard is due to the strong global convergence properties that they possess Powell[18] and due to the existence of reliable, well developed, and efficient software More[15]. Since mid eighties, many authors have considered trust-region algorithms for solving the equality constrained optimization problem

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad C_E(x) = 0.
\end{align*}
\]

Most trust-region algorithms for solving this problem try to combine the trust-region idea with the successive quadratic programming (SQP) method. See Wilson[20]. The SQP method iteratively solves a quadratic programming subproblem that consists of minimizing a quadratic model of the Lagrangian function \( \ell(x, \mu) = f(x) + \mu^T C_E(x) \) subject to satisfying a linear approximation of the constraints. If a trust-region constraint is simply added to the quadratic programming subproblem the resulting trust-region subproblem may be infeasible because there may be no intersecting points between the trust-region constraint and the hyperplane of the linearized constraints. See [3]. Even if they intersect, there is no guarantee that this will remain true if the trust-region radius is decreased. The reduced Hessian is a successful approach to overcoming the difficulty of having an infeasible trust-region subproblem. The approach was suggested in [2] and [16]. In this approach, the trial step is decomposed into two orthogonal components;
Solving constrained optimization problems

the tangential component and the normal component. Each component is computed by solving a trust-region subproblem. The two subproblems are similar to the trust-region subproblem for the unconstrained case. In this paper, we propose a new trust-region algorithm for solving Problem (1.2). The proposed algorithm uses an active-set strategy to convert Problem (1.2) to equality constrained optimization problem. The chief feature of the proposed active set is that the active set is identified and updated naturally by the trial step. Many authors have considered active set techniques for extending the SQP method to handle Problem (1.2), for example see [13], [19], [16], [21], [9], [10], and [11].

If the trust-region constraint is added to the quadratic programming subproblem of our algorithm, trial steps are computed using the projected Hessian technique in the tradition of numerous works on equality constrained optimization (see, for example, [1], [4], [8], [16], [17], [22], [9], and [10]).

Some authors have proposed a trust-region active-set algorithm for solving Problem (1.2), (see, for example, [16], [21], [9], [10], and [11]).

Following Dennis, El-Alem, and Williamson [5], we define the indicator matrix

\[ W(x) \in \mathbb{R}^{m \times m}, \text{ whose diagonal entries are} \]

\[ w_i(x) = \begin{cases} 
1, & \text{if } i \in E, \\
1, & \text{if } i \in I \text{ and } c_i(x) \geq 0, \\
0, & \text{if } i \in I \text{ and } c_i(x) < 0. 
\] (1.9)

Using the above matrix, the constrained optimization Problem (1.2) can be transformed to the following equality constrained optimization problem

\[
\begin{align*}
\text{minimize} & \quad f(x), \\
\text{subject to} & \quad 1/2C(x)^T W(x) C(x) = 0, 
\end{align*}
\] (1.10)

where \( f(x) \) is twice continuously differentiable function and \( C(x) = (c_1(x),...,c_m(x))^T \) is continuously differentiable. Thus \( \nabla C(x) W(x) C(x) \) is well-defined and continuous. The matrix \( W(x) \) is discontinuous; however, the function \( W(x) C(x) \) is Lipschitz continuous, see [5]. The Lagrangian function associated with Problem (1.10) is given by

\[ L(x, \lambda) = f(x) + \lambda \| W(x) C(x) \|^2, \] (1.11)

where \( \lambda \) is the Lagrange multiplier vector associated with the equality constraint \( C(x)^T W(x) C(x) \).

Let the quadratic model of the Lagrangian function (1.10) be

\[ q_k(d) = L_k + \nabla_x L_k^T d + \frac{1}{2}d^T H_k d, \] (1.12)

where \( H_k = \nabla^2 f_k + \nabla C_k W_k \nabla C_k^T \). Consider Problem (1.10), the first-order necessary condition for the point \( x_* \) to be a stationary point is the existence of the Lagrange multiplier vector \( \lambda_* \in \mathbb{R}^m \) such that

\[
\begin{align*}
\nabla f(x_*) + \nabla C(x_*) W(x_*) C(x_*) \lambda_* &= 0, \\
C(x_*)^T W(x_*) C(x_*) &= 0.
\end{align*}
\] (1.13) (1.14)

It is easy to see that equations (1.4)-(1.8) imply equations (1.13)-(1.14) but the converse is not true in general. We design our trust-region algorithm such that, if a point \( (x_*) \) satisfies
(1.13)-(1.14) and if all accumulation points are regular, then \( x^* \) also satisfies (1.4)-(1.8). The rest of this section introduces some notations. In Section 2, we present a detailed description of our trust-region algorithm. In Section 3, we present assumptions under which we prove global convergence theory. Sections 4-6 are devoted to presenting our global convergence theory. Section 7 contains a Matlab implementation of the proposed algorithm and our numerical results. Section 8 contains concluding remarks. Subscripted functions denote function values at particular points; for example, \( f_k = f(x_k) \), \( L_{k+1} = L(x_{k+1}, \lambda_{k+1}) \), \( \nabla_x L_k = \nabla_x L(x_k, \lambda_k) \), and so on. However, the arguments of the functions are not abbreviated when emphasizing the dependence of the functions on their arguments. We use the same symbol 0 to denote the real number zero, the zero vector, and the zero matrix. The matrix \( H_k \) denotes the Hessian of the Lagrangian function (1.11) at the point \( (x_k, \lambda_k) \) or an approximation to it. Finally, all norms are \( l_2 \)-norms.

# 2 Algorithm Outline

This section is devoted to presenting the detailed description of our trust-region algorithm for solving Problem (1.10). The algorithm combines ideas from [2], [5], [7], [16], and [4]. In the following section we discuss how the trial step \( d_k \) can be computed.

## 2.1 Computing the Trial Step \( d_k \)

In this approach, the trial step \( d_k \) is decomposed into two orthogonal components; the normal component \( d_k^n \) and the tangential component \( d_k^t \). The trial step \( d_k \) has the form \( d_k = d_k^n + d_k^t \), where \( d_k^n = Z_k d_k^t \) and \( Z_k \) is a matrix whose columns form an orthogonal basis for the null space of \( (\nabla C_k W_k C_k)^T \). See ([2], [4], and [16]).

In our algorithm, the normal component \( d_k^n \) is obtained by solving the following trust-region subproblem

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| W_k (C_k + \nabla C_k^T d_k^n) \|^2 \\
\text{subject to} & \quad \| d^n \| \leq \theta \delta_k, \quad (2.1)
\end{align*}
\]

for some \( \theta \in (0, 1) \), where \( \delta_k \) is the trust-region radius. To solve the above problem, any method that approximates the solution of the above problem can be used as long as a fraction of Cauchy decrease condition is satisfied. That is, for some \( \theta \in (0, 1) \),

\[
\| W_k C_k \|^2 - \| W_k (C_k + \nabla C_k^T d_k^n) \|^2 \geq \theta \{ \| W_k C_k \|^2 - \| W_k (C_k + \nabla C_k^T d_k^{ncp}) \|^2 \}, \quad (2.2)
\]

where \( d_k^{ncp} = -s_k^{ncp} \nabla C_k W_k C_k \) is the normal Cauchy step and the parameter \( s_k^{ncp} \) is defined by

\[
s_k^{ncp} = \begin{cases} \\
\frac{\| \nabla C_k W_k C_k \|^2}{\| W_k (\nabla C_k^T \nabla C_k W_k C_k) \|^2} & \text{if } \frac{\| \nabla C_k W_k C_k \|^3}{\| W_k (\nabla C_k^T \nabla C_k W_k C_k) \|^2} \leq \delta_k \\
\frac{\delta_k}{\| \nabla C_k W_k C_k \|} & \text{and } \| W_k (\nabla C_k^T \nabla C_k W_k C_k) \| > 0,
\end{cases} \quad (2.3)
\]

Once the normal component \( d_k^n \) is obtained, the tangential component \( d_k^t = Z_k d_k^t \) is computed by solving the following trust-region subproblem

\[
\begin{align*}
\text{minimize} & \quad [Z_k^T (\nabla_x L_k + H_k d_k^t)]^T d^t + \frac{1}{2} d^T H_k d^t \\
\text{subject to} & \quad \| Z_k d^t \| \leq \Delta_k, \quad (2.4)
\end{align*}
\]
where $\Delta_k = \sqrt{s_k^2 - \|d_k^p\|^2}$ and $\bar{H}_k = Z_k^T H_k Z_k$. To solve the above problem, any method that approximates the solution of the above problem can be used as long as a fraction of Cauchy decrease condition is satisfied. That is, for some $\vartheta \in (0, 1]$,

$$q_k(d_k^p) - q_k(d_k^p + Z_k d_k^{tcp}) \geq \vartheta[q_k(d_k^p) - q_k(d_k^p + Z_k d_k^{tcp})], \quad (2.5)$$

where $d_k^{tcp} = -s_k^{tcp} Z_k^T \nabla q_k(d_k^p)$ is the tangential Cauchy step and the parameter $s_k^{tcp}$ is defined by

$$s_k^{tcp} = \begin{cases} \frac{\|Z_k^T \nabla q_k(d_k^p)\|^2}{(Z_k^T \nabla q_k(d_k^p))^T H_k Z_k^T \nabla q_k(d_k^p)} & \text{if } \frac{\|Z_k^T \nabla q_k(d_k^p)\|^3}{(Z_k^T \nabla q_k(d_k^p))^T H_k Z_k^T \nabla q_k(d_k^p)} \leq \Delta_k \\
\frac{\Delta_k}{\|Z_k^T \nabla q_k(d_k^p)\|} & \text{otherwise.} \end{cases} \quad (2.6)$$

Since our convergence theory is based on the fraction of Cauchy decrease condition, any method that computes the two components of the trial step in such a way that they produce the double fraction of the Cauchy decrease on the tangential and the normal predicted decrease can be used. Therefore, the conjugate gradient method is used to compute the two components of the trial step. This method is known to be suitable for large problems for which effective preconditions are known. For more details see Section 9 of [4]. For theoretical purpose, the sequence $\{\lambda_k\}$ is required to be bounded. Therefore, any approximation to the Lagrange multiplier vectors $\lambda_{k+1}$ satisfied the above conditions can be used. Our way for updating $\lambda_k$ is presented in Step 4 of algorithm (2.1). Similarly, the sequence $\{H_k\}$ of approximate Hessian is required to be bounded. Thus, any approximation to the Hessian matrices that produces a bounded sequence of Hessian can be used.

### 2.2 Testing the Step and Updating $\delta_k$

Once the trial step is computed, it needs to be tested to determine whether it will be accepted. To do that, a merit function is needed. In our algorithm, the following augmented Lagrangian function is used as the merit function

$$\Phi(x, \lambda; r) = L(x, \lambda) + r\|W(x)C(x)\|^2, \quad (2.7)$$

where $r > 0$ is a parameter usually called the penalty parameter. To test the step, we compare the actual reduction in the merit function in moving from $x_k$ to $x_k + d_k$ versus the predicted reduction. The actual reduction in the merit function is defined as

$$\text{Ared}_k = L(x_k, \lambda_k) - L(x_{k+1}, \lambda_k) - \Delta \lambda_k \|W_{k+1} C_{k+1}\|^2 + r_k[\|W_{k} C_{k}\|^2 - \|W_{k+1} C_{k+1}\|^2], \quad (2.8)$$

where $\Delta \lambda_k = (\lambda_{k+1} - \lambda_k)$. The predicted reduction in the merit function is defined as

$$\text{Pred}_k = -\nabla_x L(x_k, \lambda_k)^T d_k - \frac{1}{2} d_k^T H_k d_k - \Delta \lambda_k \|W_k (C_k + \nabla C_k^T d_k)\|^2 + r_k[\|W_k C_k\|^2 - \|W_k (C_k + \nabla C_k^T d_k)\|^2]. \quad (2.9)$$
The predicted reduction can be written as
\[ \text{Pred}_k = q_k(0) - q_k(d_k) - \Delta \lambda_k \| W_k (C_k + \nabla C_k^T d_k) \|^2 + r_k [ \| W_k C_k \|^2 - \| W_k (C_k + \nabla C_k^T d_k) \|^2 ] \].  \( 2.10 \)

After computing a trial step \( d_k \) and updating the Lagrange multipliers \( \lambda_k \), the penalty parameter is updated to ensure that \( \text{Pred}_k \geq 0 \). Our way of updating \( r_k \) is presented in Step 5 of algorithm (2.1). After that, the step is tested to know whether it is accepted. This is done by comparing \( \text{Pred}_k \) against \( A_{\text{red}} \).

If \( A_{\text{red}} < \eta_1 \) where \( \eta_1 \in (0, 1) \) is a small fixed constant, then the step is rejected. In this case, the radius of the trust region \( \delta_k \) is decreased by setting \( \delta_k = \alpha_1 \| d_k \| \), where \( \alpha_1 \in (0, 1) \), and another trial step is computed using the new trust-region radius. If \( A_{\text{red}} \geq \eta_1 \), then the step is accepted and the trust-region radius is updated. Our way of updating \( \delta_k \) is presented in Step 6 of algorithm (2.1). Finally, the algorithm is terminated when \( \| Z_k^T \nabla_x L_k \| + \| W_k C_k \| \leq \epsilon \), for some \( \epsilon > 0 \).

### 2.3 Main Algorithm

A formal description of our trust-region algorithm is presented in the following algorithm.

**Algorithm 2.1 (The Main Algorithm)**

**Step 0.** (Initialization)

Given \( x_0 \in \mathbb{R}^n \). Compute \( W_0 \) and evaluate \( \lambda_0 \). Set \( r_0 = 1 \) and \( \beta > 0 \)

Choose \( \varepsilon, \alpha_1, \alpha_2, \eta_1, \eta_2, \delta_{\text{min}}, \delta_{\text{max}}, \) and \( \delta_0 \) such that \( \varepsilon > 0, 0 < \alpha_1 < 1 < \alpha_2 \) and \( 0 < \eta_1 < \eta_2 < 1 \) and \( \delta_{\text{min}} \leq \delta_0 \leq \delta_{\text{max}} \).

Set \( k = 0 \).

**Step 1.** (Test for convergence)

If \( \| Z_k^T \nabla_x L_k \| + \| W_k C_k \| \leq \epsilon \), then terminate the algorithm.

**Step 2.** (Compute a trial step)

If \( \| W_k C_k \| = 0 \), then

a) Set \( d_k^n = 0 \).

b) Compute the step \( d_k^n \) by solving the trust-region subproblem (2.4).

c) Set \( d_k = Z_k d_k^n \).

Else

a) Compute the normal component \( d_k^n \) by solving the trust-region subproblem (2.1).

b) If \( \| Z_k^T (\nabla_x L_k + H_k d_k^n) \| = 0 \), then set \( d_k = 0 \).

Else, compute \( d_k^n \) by solving the trust-region subproblem (2.4).

End if

c) Set \( d_k = d_k^n + Z_k d_k^n \) and \( x_{k+1} = x_k + d_k \).

End if.
Solving constrained optimization problems

Step 3. (Update the active set)

Compute $W_{k+1}$.

Step 4. (Compute the Lagrange multiplier $\lambda_{k+1}$)

Compute $\lambda_{k+1}$ by solving

$$\text{minimize } \|\nabla f_{k+1} + \nabla C_{k+1} W_{k+1} C_{k+1} \lambda\|^2.$$ 

Step 5. (Update the penalty parameter $r_k$)

If $\text{Pred}_k \geq \frac{1}{2} \left[ \|W_k C_k\|^2 - \|W_k (C_k + \nabla C_k^T d_k)\|^2 \right]$ then set $r_{k+1} = r_k$

else set

$$r_k = \frac{2 [q_k(d_k) - q_k(0) + \Delta \lambda_k \|W_k (C_k + \nabla C_k^T d_k)\|^2]}{\|W_k C_k\|^2 - \|W_k (C_k + \nabla C_k^T d_k)\|^2} + \beta, \quad (2.11)$$

End if

Step 6. (Test the step and update the trust-region radius)

While $\frac{\text{Ared}_k}{\text{Pred}_k} < \eta_1$

Reduce the trust-region radius by setting $\delta_k = \alpha_1 \|d_k\|$. and compute a new trial step $d_k$

End while

If $\eta_1 \leq \frac{\text{Ared}_k}{\text{Pred}_k} < \eta_2$, then

Accept the step: $x_{k+1} = x_k + d_k$.
Set the trust-region radius: $\delta_{k+1} = \max(\delta_k, \delta_{\text{min}})$.

End if

If $\frac{\text{Ared}_k}{\text{Pred}_k} \geq \eta_2$, then

Accept the step: $x_{k+1} = x_k + d_k$.
Set the trust-region radius: $\delta_{k+1} = \min\{\delta_{\text{max}}, \max\{\delta_{\text{min}}, \alpha_2 \delta_k\}\}$.

End if

Step 7. Set $k = k + 1$ and go to Step 1.
3 Problem Assumptions

We state the assumptions under which our global convergence theory is proved. Let \( \{(x_k, \lambda_k)\} \) be the sequence of points generated by the algorithm and let \( \Omega \) be a convex subset of \( \mathbb{R}^n \) that contains all iterates \( x_k \) and \( x_k + d_k \), for all trial steps \( d_k \) examined in the course of the algorithm. On the set \( \Omega \), the following assumptions are imposed.

**A1.** The functions \( f, C_E \) and \( C_I \) are twice continuously differentiable for all \( x \in \Omega \).

**A2.** The matrix \( \nabla C(x) \) has full column rank.

**A3.** All of \( f(x), \nabla f(x), \nabla^2 f(x), \|W_k C_k\|, \nabla C_k W_k C_k^T, \nabla C_k W_k \nabla C_k^T \), and \( ((\nabla C_k W_k C_k)^T (\nabla C_k W_k C_k))^{-1} \), are uniformly bounded in \( \Omega \).

**A4.** The sequence \( \{\lambda_k\} \) is bounded.

**A5.** If an approximation to the Hessian of the Lagrangian is used, then the sequence of approximated Hessian matrices \( \{H_k\} \) is bounded.

An immediate consequence of assumptions \( A_4 \) and \( A_5 \) is the existence of a constant \( b_1 > 0 \) that does not depend on \( k \) such that \( \|H_k\| \leq b_1, \|Z_k^T H_k\| \leq b_1 \), and \( \|Z_k^T H_k Z_k\| \leq b_1 \).

4 Technical Lemmas

We present some important lemmas needed in the subsequent proofs.

**Lemma 4.1** Assume \( A_1 - A_3 \). Then at any iteration \( k \), there exist a constant \( K_1 > 0 \) such that

\[
\|d_n^k\| \leq K_1 \|W_k C_k\|. \tag{4.1}
\]

**Proof.** The proof is similar to the proof of Lemma 7.1 of (1997)[4]. \( \square \)

**Lemma 4.2** Assume \( A_1 \) and \( A_3 \). Then \( W(x)C(x) \) is Lipschitz continuous in \( \Omega \).

**Proof.** See Lemma 4.1 of [5]. \( \square \)

From the above lemma, we conclude that \( C(x)^T W(x)C(x) \) is differentiable and \( \nabla C(x) W(x) C(x) \) and \( \nabla C(x)^T W(x) \nabla C(x)^T \) is Lipschitz continuous in \( \Omega \).

The following lemma shows that, at any iteration \( k \), the predicted decrease obtained by the normal component of the trial step in the 2-norm of the linearized constraints is at least equal to the decrease obtained by the Cauchy step.

**Lemma 4.3** Assume \( A_1 - A_5 \). Then there exists a constant \( K_2 > 0 \) such that the predicted decrease obtained by the normal component of the trial step \( d_k \) satisfies

\[
\|W_k C_k\|^2 - \|W_k (C_k + \nabla C_k^T d_k)\|^2 \geq K_2 \|W_k C_k\| \min\{\delta_k, \|W_k C_k\|\}. \tag{4.2}
\]
Solving constrained optimization problems

Proof. See El-Alem [6].

From the way of updating the penalty parameter $r_k$ (See step 5 of algorithm 2.1) and the above lemma, we have, for all $k$,

$$
Pred_k \geq \frac{r_k}{2}K_2 \|W_k C_k\| \min\{\delta_k, \|W_k C_k\|\}. \quad (4.3)
$$

The following lemma gives a lower bound to the predicted decrease obtained by the tangential component of the trial step. In this lemma, we use the fact that the decrease obtained by the step $d^*_k$ is greater than the decrease obtained by the Cauchy step.

**Lemma 4.4** Assume $A_1$-$A_5$. Then there exist a positive constants $K_3$ and $K_4$ such that the predicted decrease obtained by the tangential component $d^*_k = Z_k d^p_k$ of the trial step satisfies

$$
q_k(d^*_k) - q_k(d_k) \geq K_3 \|Z_k^T \nabla q_k(d^p_k)\| \min\{\Delta_k, K_4 \|Z_k^T \nabla q_k(d^p_k)\|\}, \quad (4.4)
$$

where $\Delta_k = \sqrt{\delta_k^2 - \|d^p_k\|^2}$.

Proof. See El-Sobky [10] ∎

**Lemma 4.5** At any iteration $k$, let $U(x_k) \in \mathbb{R}^{m \times m}$ be a diagonal matrix whose diagonal entries are

$$(u_k)_j = \begin{cases} 1 & \text{if } (C(x_k))_j < 0 \text{ and } (C(x_{k+1}))_j \geq 0, \\ -1 & \text{if } (C(x_k))_j \geq 0 \text{ and } (C(x_{k+1}))_j < 0, \\ 0 & \text{otherwise}, \end{cases} \quad (4.5)
$$

where $j = 1, 2, \ldots, m$. Then

$$
W_{k+1} = W_k + U_k. \quad (4.6)
$$

Proof. The proof is similar to the proof of Lemma 5.1 of [10]. ∎

**Lemma 4.6** Assume $A_1$ and $A_3$. At any iteration $k$, there exists a positive constant $K_4$ independent of $k$, such that

$$
\|U(x_k)C(x_k)\| \leq K_5 \|d_k\|, \quad (4.7)
$$

where $U_k \in \mathbb{R}^{m \times m}$ is the diagonal matrix whose diagonal entries are defined in (4.5).

Proof. The proof is similar to the proof of Lemma 5.2 of [10]. ∎

The following two lemmas give upper bounds on the difference between the actual reduction and the predicted reduction. It shows how accurate our definition of $Pred_k$ is as an approximation to $Ared_k$.

**Lemma 4.7** Assume $A_1$-$A_5$. Then there exists a positive constant $K_6$, $K_7$, and $K_8$ such that

$$
|Ared_k - Pred_k| \leq K_6 \|d_k\|^2 + K_7 r_k \|d_k\|^2 \|W_k C_k\| + K_8 r_k \|d_k\|^3 + K_9 r_k \|d_k\|^2. \quad (4.8)
$$
Proof. From equations (2.8), (2.9), (4.6), and using Cauchy-Schwarz inequality, we have

\[
|\text{Ared}_k - \text{Pred}_k| \leq \frac{1}{2} \|\nabla^2 f(x_k) - \nabla^2 f(x_k + \xi_1 d_k)\| d_k \|^2 + |\lambda_{k+1} | \| W_k \nabla C^T(x_k + \xi_2 d_k)\|^2 d_k \|^2 \\
+ |\lambda_{k+1} | \| U_k C_k\|^2 + |\lambda_{k+1} | \| \nabla C(x_k + \xi_2 d_k)\| U_k C_k\| d_k \| \\
+ |\lambda_{k+1} | \| U_k \nabla C^T(x_k + \xi_2 d_k)\|^2 d_k \|^2 + |\lambda_{k+1} | \| W_k \nabla C_k\|^2 d_k \|^2 \\
+ \Delta \lambda_k \| \nabla C_k - \nabla C(x_k + \xi_2 d_k)\| W_k C_k\| d_k \| + 2r_k \| \nabla C_k - \nabla C(x_k + \xi_2 d_k)\| W_k C_k\| d_k \| \\
+ r_k \| \nabla C_k W_k \nabla C_k^T - \nabla C(x_k + \xi_2 d_k) W_k \nabla C^T(x_k + \xi_2 d_k)\| d_k \|^2 \\
+ r_k \| U_k C_k\|^2 + 2r_k \| \nabla C(x_k + \xi_2 d_k)\| U_k C_k\| d_k \| + r_k \| U_k \nabla C^T(x_k + \xi_2 d_k)\| d_k \|^2 ,
\]

for some $\xi_1$ and $\xi_2 \in (0, 1)$. Then under problem assumption $A_1 - A_5$, there exist positive constant $K_6, K_7, K_8$, and $K_9$ such that inequality (4.8) hold.

If the penalty parameter uniformly bounded, the next lemma shows that the predicted reduction provides an approximation to the actual reduction, that is accurate to within the square of the step length.

Lemma 4.8 Assume $A_1 - A_5$. Then there exists a constant $K_{10} > 0$ such that

\[
|\text{Ared}_k - \text{Pred}_k| \leq K_{10} r_k \| d_k \|^2 .
\]

Proof. The proof follows directly from the above lemma and the fact that $\| d_k \|$ is bounded. □

The following inequality is needed in many forthcoming lemmas. In what follows, we use implicity that $\nabla C_k^T d_k = \nabla C_k^T d_k$.

Lemma 4.9 Assume $A_1 - A_5$. Then there exists a constant $K_{11} > 0$ such that

\[
q_k(0) - q_k(d_k) - \Delta \lambda_k \| W_k (C_k + \nabla C_k^T d_k)\|^2 \geq -K_{11} \| W_k C_k \|.
\]

Proof. From (1.12), we have

\[
q_k(0) - q_k(d_k) = \Delta \lambda_k \| W_k (C_k + \nabla C_k^T d_k)\|^2 = -\nabla_x L_k^T d_k - \frac{1}{2} (d_k^T H_k d_k) \\
- \Delta \lambda_k \| W_k C_k\|^2 + 2\Delta \lambda_k \| \nabla C_k W_k C_k\| d_k - |\Delta \lambda_k|^2 \| W_k \nabla C_k^T d_k\|^2 \\
\geq -\| \nabla_x L_k\| \| d_k \| - \frac{1}{2} \| H_k\| \| d_k \|^2 - |\Delta \lambda_k|^2 \| W_k C_k\|^2 \\
- 2|\Delta \lambda_k| \| \nabla C_k W_k C_k\| \| d_k \| - |\Delta \lambda_k|^2 \| W_k \nabla C_k^T\| \| d_k \|^2 \\
= -\| \nabla_x L_k\| + \frac{1}{2} \| H_k\| \| d_k \| + 2|\Delta \lambda_k| \| \nabla C_k W_k C_k\| + |\Delta \lambda_k|^2 \| W_k \nabla C_k^T\| \| d_k \|^2 \\
- |\Delta \lambda_k|^2 \| W_k C_k\|^2 .
\]

Using (4.1), the fact that $\| d_k \| < \delta_{\text{max}}$, $\Delta \lambda_k$ is bounded, and the problem assumptions, then inequality (4.10) is satisfied. □
5 Decreasing in the Model

This section deals with the predicted decrease in the merit function produced by the trial step.

Lemma 5.1 Assume $A_1$-$A_5$. Then the predicted decrease in the merit function satisfies

\[
\text{Pred}_k \geq K_3 \|Z_k^T \nabla q_k(d_k^r)\| \min \{\Delta_k, K_4 \|Z_k^T \nabla q_k(d_k^r)\|\} \\
- K_{11} \|W_k C_k\| + r_k [\|W_k C_k\|^2 - \|W_k (C_k + \nabla C_k^T d_k)\|^2].
\]

(5.1)

Proof. From (2.10), we have

\[
\text{Pred}_k = q_k(0) - q_k(d_k) - \Delta \lambda_k \|W_k (C_k + \nabla C_k^T d_k)\|^2 + r_k [\|W_k C_k\|^2 - \|W_k (C_k + \nabla C_k^T d_k)\|^2] \\
= [q_k(d_k^r) - q_k(d_k)] + [q_k(0) - q_k(d_k^r) - \Delta \lambda_k \|W_k (C_k + \nabla C_k^T d_k)\|^2] \\
+ r_k [\|W_k C_k\|^2 - \|W_k (C_k + \nabla C_k^T d_k)\|^2].
\]

From inequalities (4.4) and (4.10), we have

\[
\text{Pred}_k \geq K_3 \|Z_k^T \nabla q_k(d_k^r)\| \min \{\Delta_k, K_4 \|Z_k^T \nabla q_k(d_k^r)\|\} \\
- K_{11} \|W_k C_k\| + r_k [\|W_k C_k\|^2 - \|W_k (C_k + \nabla C_k^T d_k)\|^2].
\]

Hence the result is established. \( \square \)

If $x_k$ is feasible, then the predicted reduction does not depend on $r_k$, so we take $r_k$ as the penalty parameter from the previous iteration. The question now is how near feasibility must an iterate be in order that the penalty parameter need not be increased. The answer is given by the following lemma.

Lemma 5.2 Assume $A_1$-$A_5$. At any iteration $k$ at which the algorithm does not terminate and $\|W_k C_k\| \leq \zeta \delta_k$ where $\zeta$ is a positive constant given by

\[
\zeta \leq \min \left\{ \frac{\varepsilon}{3 \delta_{\max}}, \frac{\sqrt{3}}{2 K_1}, \frac{\varepsilon}{3 b_1 K_1 \delta_{\max}}, \frac{K_3 \varepsilon}{12 K_{11}} \min \{1, \frac{2 K_4 \varepsilon}{3 \delta_{\max}}\} \right\},
\]

(5.2)

then for any $r_k > 0$ there exist $K_{12} > 0$ independent of $k$ such that

\[
\text{Pred}_k \geq K_{12} \delta_k.
\]

(5.3)

Proof. If the algorithm does not terminate at $x_k$, then $\|Z_k^T \nabla L_k\| + \|W_k C_k\| > \varepsilon$ and since $\|W_k C_k\| \leq \zeta \delta_k$ with $\zeta \leq \frac{\varepsilon}{3 \delta_{\max}}$, therefore $\|W_k C_k\| \leq \frac{\varepsilon}{3}$ and then $\|Z_k^T \nabla L_k\| > \frac{2 \varepsilon}{3}$. By using inequality (4.1) and assumption $A_5$, we have

\[
\|Z_k^T \nabla q_k(d_k^r)\| = \|Z_k^T (\nabla L_k + H_k d_k^r)\| \\
\geq \|Z_k^T \nabla L_k\| - \|Z_k^T H_k d_k^r\| \\
\geq \frac{2 \varepsilon}{3} - b_1 K_1 \|W_k C_k\| \\
\geq \frac{2 \varepsilon}{3} - b_1 K_1 \zeta \delta_k \\
\geq \frac{\varepsilon}{3},
\]

(5.4)
where $\zeta \leq \frac{3\epsilon}{2K_{1}\delta_{\text{max}}}$. Since $\Delta_k = \sqrt{\delta_k^2 - \|d^*\|^2}$ and $\|d^*_k\| \leq K_1\|W_kC_k\| \leq K_1\zeta\delta_k \leq K_1\frac{\sqrt{3}}{2}\delta_k$, then we obtain $\Delta_k^2 = \delta_k^2 - \|d^*_k\|^2 \geq \frac{2}{3}\delta_k^2 - \frac{2}{3}\delta_k^2 = \frac{1}{3}\delta_k^2$. Hence,

$$\Delta_k \geq \frac{1}{2}\delta_k. \quad (5.5)$$

From inequalities (5.1), (5.4), and (5.5), we have

$$\text{Pred}_k \geq \frac{K_3^3}{2}\|Z_k^T\nabla g_k(d_k^*)\| \min\left\{ \frac{1}{2}\delta_k, K_4\|Z_k^T\nabla g_k(d_k^*)\| \right\}
+ \frac{K_3^3\epsilon}{12}\min\{1, \frac{2K_4\epsilon}{3\delta_{\text{max}}}\}\delta_k - K_{11}\zeta\delta_k
+ r_k[\|W_kC_k\|^2 - \|W_k(C_k + \nabla C_k^T d_k)\|^2]
\geq \frac{K_3^3}{2}\|Z_k^T\nabla g_k(d_k^*)\| \min\left\{ \frac{1}{2}\delta_k, K_4\|Z_k^T\nabla g_k(d_k^*)\| \right\} + r_k[\|W_kC_k\|^2 - \|W_k(C_k + \nabla C_k^T d_k)\|^2].$$

where $\zeta \leq \frac{K_3^3\epsilon}{12K_{11}}\min\{1, \frac{2K_4\epsilon}{3\delta_{\text{max}}}\}$. Hence

$$\text{Pred}_k \geq \frac{K_3^3}{2}\|Z_k^T\nabla g_k(d_k^*)\| \min\left\{ \frac{1}{2}\delta_k, K_4\|Z_k^T\nabla g_k(d_k^*)\| \right\}
\geq \frac{K_3^3\epsilon}{12}\min\{1, \frac{2K_4\epsilon}{3\delta_{\text{max}}}\}\delta_k
\geq K_{12}\delta_k,$$

where $K_{12} = \frac{K_3^3\epsilon}{12}\min\{1, \frac{2K_4\epsilon}{3\delta_{\text{max}}}\}$. Hence the result is established. \hfill \Box

The above lemma guarantees that if the algorithm does not terminate and if $\|W_kC_k\| \leq \zeta\delta_k$, then the penalty parameter at the current trial step does not need to be increased. This equivalent to saying that any increases in the penalty parameter occur only when $\|W_kC_k\| > \zeta\delta_k$.

**Lemma 5.3** Assume $A_1 - A_5$. If the algorithm does not terminate at $x_k$ and let $k \geq \tilde{k}$ be the index of an iteration at which $r_k$ is increased. Then there exists a constant $K_{14} > 0$ such that

$$r_k\delta_k \leq K_{14}. \quad (5.6)$$

**Proof.** Since $r_k$ is increased at the $k^{th}$ iteration, then from (2.11), we have

$$r_k = \frac{2[q_k(d_k) - q_k(0) + \Delta\lambda_k\|W_k(C_k + \nabla C_k^T d_k)\|^2]}{\|W_kC_k\|^2 - \|W_k(C_k + \nabla C_k^T d_k)\|^2} + \beta.$$

Hence,

$$\frac{r_k}{2}\|W_kC_k\|^2 - \|W_k(C_k + \nabla C_k^T d_k)\|^2 = [q_k(d_k) - q_k(d_k^*)] + [q_k(d_k^*) - q_k(0) + \Delta\lambda_k\|W_k(C_k + \nabla C_k^T d_k)\|^2]
+ \beta\left[\|W_kC_k\|^2 - \|W_k(C_k + \nabla C_k^T d_k)\|^2\right].$$

Applying inequality (4.2) to the left hand side and inequalities (4.4) and (4.10) on the right hand side we can obtain for all $k \geq \tilde{k}$ that

$$\frac{r_k}{2}K_2\|W_kC_k\| \min\{\delta_k, \|W_kC_k\|\} \leq \frac{K_3^3}{2}\|Z_k^T\nabla g_k(d_k^*)\| \min\{\Delta_k, K_4\|Z_k^T\nabla g_k(d_k^*)\|\}$$
Under assumption $A_3$, we have for all $k \geq \bar{k}$

$$\frac{r_k}{2} K_2 \min\{\delta_k, \|W_k C_k\|\} \leq K_{13}.$$ 

Since at the current trial step the penalty parameter increases, from Lemma (5.2), we have

$$\|W_k C_k\| \geq \zeta \delta_k.$$ 

Then

$$\frac{r_k}{2} K_2 \min\{1, \zeta\} \delta_k \leq K_{13}.$$ 

Hence $r_k \delta_k \leq K_{14}$ such that $K_{14} = \frac{2 K_{13}}{K_2 \min\{1, \zeta\}}$. This completes the proof. □

**Lemma 5.4** Assume $A_1$-$A_5$. If the algorithm does not terminate at $x_k$ and let $k \geq \bar{k}$ be the index of an iteration at which the penalty parameter is increased at the $h$th trial step of $k$th iteration. Then there exists $\sigma$, such that

$$\delta_{kh} \geq \sigma. \quad (5.7)$$

**Proof.** To begin, we note that if $h = 0$, i.e., we are at the first trial step of iteration $k$, then by algorithm (2.1), $\delta_k$ cannot have become smaller than $\delta_{\text{min}}$ during the course of the iteration. Thus, we can restrict our attention to the case where $h \geq 1$. Our proof will consist of showing the existence of $\sigma$ such that $\delta_{kh} \geq \sigma$ whether or not $d_{kh}$ is acceptable. Remember that for all the rejected trial steps we have $\delta_{kj} = \alpha_1 \|d_{kj-1}\|$. We consider two cases:

(i) If $\|W_k C_k\| > \zeta \delta_k$ for all $j = 0, ..., h$, then from inequality (4.3) we have

$$\text{Pred}_{kj} \geq \frac{r_{kj}}{2} K_2 \|W_k C_k\| \min\{\delta_{kj}, \|W_k C_k\|\} \geq \frac{r_{kj}}{2} K_2 \|W_k C_k\| \min\{1, \zeta\} \delta_{kj}. \quad (5.8)$$

From inequalities (4.9) and (5.8), we have

$$\frac{|A_{red,kj} - \text{Pred}_{kj}|}{\text{Pred}_{kj}} \leq \frac{2 K_{10} \|d_{kj}\|}{K_2 \|W_k C_k\| \min\{1, \zeta\}}. \quad (5.9)$$

Since all the steps $d_{kj}$ are rejected for all $j = 0, ..., h - 1$, it must be the case that

$$(1 - \eta_1) < \frac{|A_{red,kj}|}{\text{Pred}_{kj}} - 1 \quad . \quad (5.10)$$

So from inequalities (5.9) and (5.10), we have

$$(1 - \eta_1) \leq \frac{2 K_{10} \|d_{kj}\|}{K_2 \|W_k C_k\| \min\{1, \zeta\}}.$$
Hence,

$$\|d_{k'}\| \geq \frac{(1-\eta_1)K_2 \min\{1,\zeta\}}{2K_{10}} \|W_kC_k\|,$$

for all \(j = 0, \ldots, h - 1\). Since \(\delta_{kh} = \alpha_1\|d_{k-1}\|, \|W_kC_k\| > \zeta \delta_{ko}\), and \(\delta_{k0} \geq \delta_{\min}\), it follows that

$$\delta_{kh} = \alpha_1\|d_{k-1}\| \geq \frac{\alpha_1(1-\eta_1)K_2 \min\{1,\zeta\}}{2K_{10}} \zeta \delta_{\min} = K_{15}. \quad (5.12)$$

(ii) If \(\|W_kC_k\| > \zeta \delta_{k'}\) does not hold for all \(j = 0, \ldots, h\), then there exists a largest index \(l\), \(0 \leq l < h\) such that \(\|W_kC_k\| \leq \zeta \delta_{k'}\) holds. If \(h = l + 1\), then from the way of updating the trust-region radius, \(\delta_{kh} = \alpha_1\|d_{k'}\|\). On the other hand, if \(h \neq l + 1\), since \(\|W_kC_k\| > \zeta \delta_{k'}\) for all \(j = l + 1, \ldots, h\), then from inequality (5.11), we have

$$\|d_{k'}\| \geq \frac{(1-\eta_1)K_2 \min\{1,\zeta\}}{2K_{10}} \|W_kC_k\|$$

for all \(j = l+1, \ldots, h-1\). Since \(d_{kh-1}\) and \(d_{k'+1}\) are rejected trial steps and using \(\|W_kC_k\| > \zeta \delta_{k'+1}\), then

\[
\begin{align*}
\delta_{kh} &= \alpha_1\|d_{k-1}\| \\
&\geq \frac{\alpha_1(1-\eta_1)K_2 \min\{1,\zeta\}}{2K_{10}} \|W_kC_k\| \\
&\geq \frac{\alpha_1(1-\eta_1)K_2 \min\{1,\zeta\}}{2K_{10}} \zeta \delta_{k'+1} \\
&\geq \frac{\alpha_2\zeta(1-\eta_1)K_2 \min\{1,\zeta\}}{2K_{10}} \|d_{k'}\| \\
&\geq K_{16}\|d_{k'}\|, \quad (5.13)
\end{align*}
\]

where \(K_{16} = \min\{\alpha_1, \frac{\alpha_2\zeta(1-\eta_1)K_2 \min\{1,\zeta\}}{2K_{10}}\}\). From Lemma (5.3) and inequality (5.13), we have

$$r_{k'}\|d_{k'}\| \leq r_{kh} \frac{\delta_{kh}}{K_{16}} \leq \frac{K_{14}}{K_{16}} = K_{17}. \quad (5.14)$$

Since \(\|W_kC_k\| \leq \zeta \delta_{k}\) and from inequalities (4.8) and (5.14), we have

\[
\begin{align*}
|\text{Ared}_{k'} - \text{Pred}_{k'}| &\leq K_6\|d_{k'}\|^2 + K_7r_{k'}\|d_{k'}\|^2\|W_kC_k\| + K_8r_{k'}\|d_{k'}\|^3 + K_9r_{k'}\|d_{k'}\|^2 \\
&\leq \left[K_6 + K_7K_{17}\zeta + K_8K_{17}\right]\|d_{k'}\| + K_9K_{17}\delta_{k}.
\end{align*}
\]

Also, since \(\|W_kC_k\| \leq \zeta \delta_{k}\), then from Lemma (5.2)

$$\text{Pred}_{k'} \geq K_{12}\delta_{k}\quad (5.15)$$

Using the above inequality and inequality (5.15) and the fact that \(d_{k'}\) is rejected, we obtain

\[
(1-\eta_1) \frac{|\text{Ared}_{k'} - \text{Pred}_{k'}|}{\text{Pred}_{k'}} - 1 \leq \frac{\left[K_6 + K_7K_{17}\zeta + K_8K_{17}\right]\|d_{k'}\| + K_9K_{17}\delta_{k}}{K_{12}\delta_{k}}.
\]

Hence,

$$\|d_{k'}\| \geq \frac{(1-\eta_1)K_{12} - K_9K_{17}}{K_6 + K_7K_{17}\zeta + K_8K_{17}}. \quad (5.16)$$
From inequalities (5.13) and (5.16), we have

\[
\delta_k h \geq K_{16} \|d_k\| \\
\geq \frac{(1 - \eta_1)K_{12} - K_9 K_{17} |K_{16}|}{K_6 + K_7 K_{17} \zeta + K_8 K_{17}} = K_{18}.
\]

Defining

\[\sigma = \min\{\delta_{\min}, K_{15}, K_{18}\}.\]

This complete the proof.

In the following lemma we show that the nondecreasing sequence of penalty parameters generated by our algorithm is bounded.

**Lemma 5.5** Assume $A_1$-$A_5$. If the algorithm does not terminate, then there is some $\hat{r}$, for which

\[
\lim_{k \to \infty} r_k = \hat{r}.
\]

Furthermore, there exists some index $\bar{k}$ such that $r_k = \hat{r}$ for all $k \geq \bar{k}$.

**Proof.** From Lemmas (5.3) and (5.4), we have

\[
r_k \leq \frac{K_{14}}{\delta_k h} \leq \frac{K_{14}}{\sigma}.
\]

Therefore $\{r_k\}$ is bounded sequence and since it is nondecreasing, there exists $\hat{r} < \infty$ such that

\[
\lim_{k \to \infty} r_k = \hat{r}.
\]

From the way of updating the penalty parameter, we know that every increase in the penalty parameter is by at least $\beta$. Then there must be at most finitely many increases and the proof is complete.

\[\square\]

## 6 Global Convergence Theory

In this section, we prove our main global convergence results for our trust-region algorithm for solving Problem (1.2). The following lemma shows that our algorithm is well defined in the sense that at each iteration we can find an acceptable step after a finite number of trial step computations or, equivalently, trust-region reductions. This allows us to drop the consideration of trial steps and only consider ”successful trial steps,” $\{s_k\}$.

**Lemma 6.1** Assume $A_1$-$A_5$. Unless some iterate $x_k$ satisfies the termination condition of algorithm (2.1), an acceptable step from $x_k$ is found after finitely many trial steps.

**Proof.** The proof follows from Theorem 5.1 of [6].

\[\square\]

**Lemma 6.2** Assume $A_1$-$A_5$. If the algorithm does not terminate, then there exists $\hat{\sigma} > 0$, such that

\[
\delta_k h \geq \hat{\sigma} \quad \text{for all } k, h.
\]

\[\square\]
\textbf{Proof.} If the first trial step is acceptable, then by algorithm (2.1), $\delta_k$ cannot have become smaller than $\delta_{\min}$ during the course of the iteration. Thus we can consider to the case where there is at least one unsuccessful trial step. Let us assume then that we have $j$ unsuccessful steps. Our proof consists of showing $\delta_{k_j} \geq \hat{\sigma}$ whether or not $d_{k_j}$ is acceptable, i.e., is $d_k$. Remember that for all the rejected trial steps we have $\delta_{k_j+1} = \alpha_1 \|d_{k_j}\| < \delta_{k_j}$. We consider two cases:

(i) If $\|W_k C_k\| > \zeta \delta_{k_j}$ for all $h = 0, \ldots, j$. (ii) If $\|W_k C_k\| > \zeta \delta_{k_j}$ does not hold for some $h$ such that $0 < h \leq j$. The proof of (i) is exactly the same as in the proof of Lemma (5.4). (ii)If $\|W_k C_k\| > \zeta \delta_{k_j}$ for all $h = 0, \ldots, j$, as in Lemma (5.4), we let $l$ be the largest index such that $\|W_k C_k\| \leq \zeta \delta_{k_l}$ holds. Now, since $\|W_k C_k\| \leq \zeta \delta_{k_j}$ for all $h \leq l$, it follows from Lemma (5.2) that for all such $h$, $\text{Pred}_{k_j} \geq K_1 \delta_{k_j}$. From Lemma (4.8), $|\text{Ared}_{k_j} - \text{Pred}_{k_j}| \leq K_{10} r_{k_j} \|d_{k_j}\|^{2}$ and because the step $d_{k_j}$ is an unacceptable step, we have

$$1 - \eta_1 < \frac{|\text{Ared}_{k_j} - \text{Pred}_{k_j}|}{\text{Pred}_{k_j}} \leq \frac{K_{10} r_{k_j} \|d_{k_j}\|^{2}}{K_{12} \delta_{k_j}} \leq \frac{K_{10} \hat{\tau} \|d_{k_j}\|}{K_{12}}.$$  

The above inequality implies that for all $h \leq l$,

$$\delta_{k_j} \geq \|d_{k_j}\| \geq \frac{(1 - \eta_1) K_{12}}{K_{10} \hat{\tau}}.$$  

For all $h > l$ we have from the above inequality and inequality (5.13),

$$\delta_{k_j} \geq K_{16} \|d_{k_j}\| \geq K_{16} \frac{K_{12} (1 - \eta_1)}{K_{10} \hat{\tau}} = K_{19}.  \tag{6.3}$$  

Hence,

$$\delta_{k_j} \geq \hat{\sigma}$$

where, $\hat{\sigma} = \min\{\delta_{\min}, K_{15}, K_{19}\}$.

\begin{flushright} \Box \end{flushright}

\textbf{Theorem 6.1} Assume $A_1$-$A_5$. Then the sequence of iterates generated by the algorithm satisfies

$$\lim_{k \to \infty} \|W_k C_k\| = 0.$$  

\textbf{Proof.} We prove the theorem by contradiction. We begin by assuming that there exists an infinite sequence of indices $\{k_j\}$ such that $\|W_k C_k\|$ is bounded a way from zero for all $k \in \{k_j\}$ . This implies that there exists $\tau > 0$ such that for all $k \in \{k_j\}$, $\|W_k C_k\| \geq \tau$. Now for each $k_j \geq k$ where is as in Lemma (5.3), we have from inequality (4.3)

$$\text{Pred}_k \geq \frac{r_{k_j}}{2} K_2 \|W_{k_j} C_{k_j}\| \min\{\delta_{k_j}, \|W_{k_j} C_{k_j}\|\} \geq \frac{\hat{\tau}}{2} K_2 \tau \min\{1, \frac{\tau}{\delta_{\max}}\} \hat{\sigma} \geq K_{20} > 0.$$  

Remember that are only looking at successful steps at this point in the analysis, so

$$\Phi_{k_j} - \Phi_{k_j+1} = \text{Ared}_{k_j} \geq \eta_1 \text{Pred}_{k_j} \geq \eta_1 K_{20} > 0.$$  

Since $\{\Phi_k\}$ is bounded below, a contradiction arise if we let $k_j$ go to infinity. \hfill \Box
Theorem 6.2 Assume $A_1$–$A_5$. The algorithm terminated because for all $\epsilon > 0$
\[ \|Z_k^T \nabla_x L_k\| + \|W_k C_k\| \leq \epsilon. \]

Proof. Suppose that the algorithm does not terminate and that some subsequence of $\{\|Z_k^T \nabla_x L_k\|\}$ converges to zero, then nontermination is immediately contradicted by Theorem (6.1). Suppose that $\|Z_k^T \nabla_x L_k\| \geq \epsilon_1$, for some $\epsilon_1 > 0$. Since $\|W_k C_k\|$ goes to zero by Theorem (6.1) and the sequence of trust-region radii is bounded below by $\hat{\sigma}$, there exists an index $\hat{k} > \bar{k}$ such that for all $k \geq \hat{k}$, $\|W_k C_k\| \leq \zeta \hat{\sigma} \leq \zeta \delta_k$, with $\zeta$ as in (5.2). Therefore, by Lemmas (5.2) and (6.2), we have
\[ \Phi_k - \Phi_{k+1} = A_{red} k \geq \eta_1 P_{red} k \geq \eta_1 K_{12} \delta_k \geq \eta_1 K_{12} \hat{\sigma}. \]
This contradicts the bounden of $\Phi_k$ and completes the proof. \[ \square \]

From the above two theorems, we conclude that, given any $\epsilon > 0$, the algorithm terminates because $\|Z_k^T \nabla_x L_k\| + \|W_k C_k\| < \epsilon$, for some finite $k$.

7 Numerical Results

In this section, we report our preliminary numerical experience with the new trust-region algorithm for solving Problem (1.2). Our program was written in MATLAB and run under MATLAB Version 7 with machine epsilon about $10^{-16}$. Given a starting point $x_0$, we choose the initial trust-region radius to be $\delta_0 = \text{max}(\|d_0^{ncp}\|, \|d_0^{tcp}\|, \|\delta_{min}\|)$, where $\delta_{min}$ was taken to be $10^{-3}$. We choose the maximum trust-region radius to be $\delta_{\text{max}} = 10^5 \delta_0$. The values of the constants that are needed in Step 0 of Algorithm (2.1) were set to be $\eta_1 = 10^{-4}$, $\eta_2 = 0.5$, $\alpha_1 = 0.05$, $\alpha_2 = 2$, $\epsilon = 10^{-8}$, and $\beta = 0.1$.

For computing the two components of the trial steps, we used the conjugate gradient method. Successful termination with respect to the proposed trust-region algorithm means that the termination condition of the algorithm is met with $\epsilon = 10^{-8}$. On the other hand, unsuccessful termination means that the number of iterations is greater than 300, the number of function evaluations is greater than 500, or the length of the trial step is less than $\epsilon = 10^{-8}$. We report the numerical results of the proposed algorithm. The numerical results are summarized in Tables (8.1) and (8.2). The problems which are tested in these Tables are the Hock and Schittkowski’s subset of the Constrained and Unconstrained Testing Environment (the CUTE collection). See Hock and Schittkowski [14]. In Tables (8.1) and (8.2), columns 1-4 give the data of the problem. The following abbreviations are used:

- HS-number stands for the number of problem from Hock and Schittkowski [14].
- n: number of variables.
- m: number of equality constraints.
- p: number of inequality constraints.

In the fifth column, of Table (8.1), we list the number of iterations and the number of function evaluation of LANCELOT and in the fifth column, of Table (8.2), we list the number of iterations and the number of function evaluation of Elsobky [11]. In sixth column of Tables (8.1) and (8.2), we list the number of iterations and the number of function evaluations of the proposed algorithm. In many of the test problems reported in Tables (8.1) and (8.2), the number of
iterations and the number of function evaluations of the proposed trust-region algorithm are better than those obtained by LANCELOT and by Elsobky [11]. This indicates the viability of the proposed approach. However, we believe that the proposed algorithm needs to be refined with efficiency in mined to be suitable for large-scale problems.

8 Concluding Remarks

We introduced a new trust-region algorithm for solving the Constrained optimization problem with equality and inequality constraints. This algorithm can be viewed as an extension of Byrd and Omojokun’s trust-region algorithm for solving the equality constrained optimization problem. The algorithm handles inequality constraints in a fashion similar to the approach of Dennis, El-Alem, and Williamson for treating the active constraints. At every iteration, the step is computed by solving two simple trust-region subproblems similar to those for unconstrained optimization. We proved that the algorithm is globally convergent in the sense that, in the limit, a subsequence of the iteration sequence generated by the algorithm satisfies the KKT conditions. For future work, there are many questions that should be answered. Although we have implemented the algorithm and tested it, we believe that the implementation of the algorithm should be refined with efficiency in mined. In particular, a better way of solving the trust-region subproblems that can handle large-scale problems should be used.

A related important question that has to be looked at is how to use a secant approximation of the Hessian of the Lagrangian matrix in order to produce a more efficient algorithm.

Acknowledgments I would like to thank my professor Dr. El-Alem for his helpful comments.

References


Solving constrained optimization problems


### Table 8.1: Numerical results of LANCELOT and new trust region algorithm

<table>
<thead>
<tr>
<th>Problem Name</th>
<th>n</th>
<th>m</th>
<th>p</th>
<th>LANCELOT iter(nfunc)</th>
<th>New iter(nfunc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS6</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>49(56)</td>
<td>3(4)</td>
</tr>
<tr>
<td>HS7</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>18(19)</td>
<td>10(14)</td>
</tr>
<tr>
<td>HS9</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4(5)</td>
<td>5(6)</td>
</tr>
<tr>
<td>HS11</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>15(16)</td>
<td>7(18)</td>
</tr>
<tr>
<td>HS12</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>22(23)</td>
<td>19(22)</td>
</tr>
<tr>
<td>HS14</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>12(13)</td>
<td>27(52)</td>
</tr>
<tr>
<td>HS21</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>1(2)</td>
<td>3(11)</td>
</tr>
<tr>
<td>HS22</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>9(10)</td>
<td>4(11)</td>
</tr>
<tr>
<td>HS24</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>7(8)</td>
<td>1(2)</td>
</tr>
<tr>
<td>HS30</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>7(8)</td>
<td>2(10)</td>
</tr>
<tr>
<td>HS34</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>9(19)</td>
<td>5(16)</td>
</tr>
<tr>
<td>HS36</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>11(12)</td>
<td>3(12)</td>
</tr>
<tr>
<td>HS41</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>12(13)</td>
<td>7(10)</td>
</tr>
<tr>
<td>HS60</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>15(15)</td>
<td>7(9)</td>
</tr>
<tr>
<td>HS78</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>11(11)</td>
<td>9(11)</td>
</tr>
<tr>
<td>HS79</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>9(10)</td>
<td>10(28)</td>
</tr>
<tr>
<td>HS80</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>14(15)</td>
<td>8(10)</td>
</tr>
<tr>
<td>HS81</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>16(17)</td>
<td>11(19)</td>
</tr>
</tbody>
</table>

### Table 8.2: Numerical results of Elsobky[11] and new trust region algorithm

<table>
<thead>
<tr>
<th>Problem Name</th>
<th>n</th>
<th>m</th>
<th>p</th>
<th>Elsobky iter(nfunc)</th>
<th>New iter(nfunc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS6</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4(5)</td>
<td>3(4)</td>
</tr>
<tr>
<td>HS7</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>9(10)</td>
<td>10(14)</td>
</tr>
<tr>
<td>HS9</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>7(8)</td>
<td>5(6)</td>
</tr>
<tr>
<td>HS11</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>9(10)</td>
<td>7(18)</td>
</tr>
<tr>
<td>HS21</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>5(6)</td>
<td>3(11)</td>
</tr>
<tr>
<td>HS22</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>11(23)</td>
<td>4(11)</td>
</tr>
<tr>
<td>HS30</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>1(2)</td>
<td>2(10)</td>
</tr>
<tr>
<td>HS40</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>15(37)</td>
<td>17(29)</td>
</tr>
<tr>
<td>HS60</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>10(12)</td>
<td>7(9)</td>
</tr>
<tr>
<td>HS79</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>10(11)</td>
<td>10(28)</td>
</tr>
</tbody>
</table>