Journal of Computational Mathematics
Vol.36, No.6, 2018, 776–791.
http://www.global-sci.org/jcm
doi:10.4208/jcm.1705-m2016-0735

A TRUST-REGION ALGORITHM FOR SOLVING MINI-MAX PROBLEM*

Bothina El-Sobky and Abdallah Abotahoun
Department of Mathematics, Alexandria University, Alexandria, Egypt
Email: bothinaelsobky@yahoo.com, tahoun44@yahoo.com

Abstract

In this paper, we propose an algorithm for solving inequality constrained mini-max optimization problem. In this algorithm, an active set strategy is used together with multiplier method to convert the inequality constrained mini-max optimization problem into unconstrained optimization problem. A trust-region method is a well-accepted technique in constrained optimization to assure global convergence and is more robust when they deal with rounding errors. One of the advantages of trust-region method is that it does not require the objective function of the model to be convex.

A global convergence analysis for the proposed algorithm is presented under some conditions. To show the efficiency of the algorithm numerical results for a number of test problems are reported.

Mathematics subject classification: 90C30, 90B50, 65K05, 62C20.
Key words: Mini-max problem, Active-set, Multiplier method, Trust-region, Global convergence.

1. Introduction

Many real world applications can be modeled as a mini-max optimization problem. This problem arises in engineering design, computer-aided design, circuit design, chemical design, systems of nonlinear equations, problems of finding feasible points of systems of inequalities, nonlinear programming problems, multi objective problems, optimal control and others. Theoretical study for the mini-max optimization problem can be found in [1,2].

In this paper, we introduce an active-set trust-region algorithm to solve the following mini-max problem

$$\begin{array}{ll} \min_{x\in\Re^n} & \Psi(x), \\ \text{subject } to & h(x) \leq 0, \end{array} \tag{1.1}$$

where $\Psi(x) = \max_{1\leq i\leq m} f_i(x)$. The functions $f_i : \Re^n \to \Re$, $i = 1, ..., m$, and $h(x) : \Re^n \to \Re^p$, are twice continuously differentiable. The objective function $\Psi(x)$ is not necessarily differentiable even though the functions $f_i(x)$, $i = 1, ..., m$, are all differentiable. So, the classical algorithms which are using for solving smooth nonlinear programming problems can not be applied directly on Problem (1.1). There are several types of algorithms suggested to solve min-max problems, see [3–13]. The first type of algorithms shows the Problem (1.1) as a constrained non-smooth optimization problem. Therefore, general methods is used to solve it, see [14,15]. The second type of algorithms solves the Problem (1.1) by considering the special structure of its non-differentiability so as to make use of certain smooth optimization methods, see [4,16]. The third type of algorithms solves the Problem (1.1) by converting it into

* Received August 29, 2016 / Revised version received March 2, 2017 / Accepted May 8, 2017 /
Published online August 7, 2018 /

an equivalent smooth inequality constrained optimization problem by inserting a new variable $z \in \Re$.

$$\begin{array}{ll} \min_{(x^T,z)} & z \\ \textit{subject to} & h(x) \leq 0, \\ & f_i(x) - z \leq 0, \quad i = 1, ..., m. \end{array}$$

It is obviously implies that solving the finite min-max inequality constrained Problem (1.1) is equivalent to solve the above problem, see [1,2]. In this paper, the proposed approach belongs to the third type.

The above problem can be summarized as follows

$$\begin{array}{ll} \min_{\tilde{x}} & F(\tilde{x}) \\ \textit{subject to} & G(\tilde{x}) \leq 0, \end{array} \tag{1.2}$$

where $\tilde{x}$ represent the vector $(x^T, z) \in \Re^{n+1}$, $F(\tilde{x}) = z$, and $G(\tilde{x}) \in \Re^{m+p}$ is a vector whose elements are $(h(x), f_i(x) - z)^T$, $i = 1, ..., m$.

The Lagrangian function associated with Problem (1.2) is the function

$$\ell(\tilde{x}, \lambda) = F(\tilde{x}) + \lambda^T G(\tilde{x}), \tag{1.3}$$

where $\lambda \in \Re^{m+p}$ is the Lagrange multiplier vector associated with inequality constraints $G(\tilde{x})$. Let $J(\tilde{x})$ be the set of indices of violated or binding inequality constraints at a point x. i.e., $J(\tilde{x}) = \{j : G_j(\tilde{x}) \geq 0\}$. If the vectors in the set $\{\nabla G_j(\tilde{x}), j \in J(\tilde{x}_*)\}$ are linearly independent, then the point $\tilde{x}_*$ is called a regular point for Problem (1.2).

The first-order necessary conditions for the regular point $\tilde{x}_*$ to be a local minimizer of Problem (1.2) are the existence of the multiplier vector $\lambda_* \in \Re^{m+p}$ such that $(\tilde{x}_*, \lambda_*)$ satisfies

$$\nabla_{\tilde{x}} F(\tilde{x}_*) + \nabla_{\tilde{x}} G(\tilde{x}_*) \lambda_* = 0, \tag{1.4}$$

$$G(\tilde{x}_*) \leq 0, \tag{1.5}$$

$$(\lambda_*)_i G_i(\tilde{x_*}) = 0, \quad i = 1, ..., m+p, \tag{1.6}$$

$$(\lambda_*)_i \geq 0, \quad i = 1, ..., m+p. \tag{1.7}$$

Conditions (1.4)-(1.7) are also known as the Karush-Kuhn-Tucker conditions or the KKT conditions. A point $(\tilde{x}_*, \lambda_*)$ that satisfies the KKT conditions is called a KKT point. For more details, see [17].

In this paper an active set strategy is used together with a multiplier method to convert Problem (1.2) into unconstrained optimization problem. The general idea behind the active-set strategy is to identify at every iteration, the active inequality constraints and treat them as equalities. This allows the use of the well-developed techniques for solving the equality constrained optimization problems. Many authors have proposed active-set algorithms for solving general nonlinear programming problems, see, e.g., [18–21].

Following the active set strategy in [18], we define a 0-1 diagonal indicator matrix $D(x) \in \Re^{m+p \times m+p}$, whose diagonal entries are

$$d_i(\tilde{x}) = \begin{cases} 1 & \text{if } G_i(\tilde{x}) \geq 0, \\ 0 & \text{if } G_i(\tilde{x}) < 0. \end{cases} \tag{1.8}$$

Using the above matrix, Problem (1.2) is converted to the following problem

$$\begin{array}{ll} \min & F(\tilde{x}), \\ \textit{subject to} & G(\tilde{x})^T D(\tilde{x}) G(\tilde{x}) = 0. \end{array}$$

In this algorithm, the multiplier method is used to replace the above equality constrained optimization problem to the following unconstrained optimization problem and at the same time the penalty parameter needs not to go to infinity,

$$\begin{array}{ll} \min & \ell(\tilde{x},\lambda) + \frac{\rho}{2}\|D(\tilde{x})G(\tilde{x})\|_2^2, \\ subject\ to & \tilde{x} \in \Re^{n+1}, \end{array} \tag{1.9}$$

where ρ is positive parameter. For more details about the multiplier methods see [22].

The first-order necessary condition for the point $\tilde{x}_*$ to be a local minimizer of Problem (1.9) is the existence of the multiplier vector $\lambda_* \in \Re^{m+p}$ such that $(\tilde{x}_*, \lambda_*)$ satisfies

$$\nabla_{\tilde{x}}\ell(\tilde{x}_*,\lambda_*) + \rho\nabla G(\tilde{x}_*)D(\tilde{x}_*)G(\tilde{x}_*) = 0, \tag{1.10}$$

where $\nabla_{\tilde{x}}\ell(\tilde{x}_*,\lambda_*) = \nabla F(\tilde{x}_*) + \nabla G(\tilde{x}_*)\lambda_*$.

We note that if the point $(\tilde{x}_*, \lambda_*)$ satisfies the KKT conditions of Problem (1.1), then it also satisfies the first-order necessary optimal conditions of Problem (1.9) but the converse is not necessarily true. We design our algorithm in such a way that, if the point $(\tilde{x}_*, \lambda_*)$ satisfies the first-order necessary optimal condition of Problem (1.9), then it also satisfies the first-order necessary optimal conditions of Problem (1.1).

As we know a trust-region method is a well-accepted technique in nonlinear optimization to assure global convergence and is more robust when they deal with rounding errors, so we used it in this paper. One of the advantages of trust-region method is that it does not require the objective function of the model to be convex. However, in traditional trust-region method, after solving a trust-region subproblem, we need to use some criterion to check if the trial step is acceptable. If not, the subproblem must be resolved with a reduced trust-region radius. For more details see [20, 23–28].

In this paper, a global convergence theory for the proposed algorithm is introduced under some assumptions.

Subscripted functions denote function values at particular points; for example, $G_k = G(\tilde{x}_k)$, $\nabla G_k = \nabla G(\tilde{x}_k)$, $D_k = D(\tilde{x}_k)$, $\ell_k = \ell(\tilde{x}_k, \lambda_k)$, $\nabla_{\tilde{x}}\ell_k = \nabla_{\tilde{x}}\ell(\tilde{x}_k, \lambda_k)$, and so on. Finally, all norms are l_2-norms.

The rest of this section introduces some notations. In Section 2, we present an outline of the proposed trust-region algorithm. Section 3 is devoted to analysis of the global convergence of the proposed algorithm. Section 4 contains implementation of the proposed algorithm and the results of test problems. Section 5 contains concluding remarks.

2. Algorithm Outline

This section is devoted to presenting the detailed description of the proposed trust-region algorithm for solving Problem (1.1).

2.1. Compute a step s_k

In this section, a trial step s_k is evaluated by solving the following trust-region subproblem (2.1).

$$\begin{array}{ll} \min & q_k(s_k) = \ell_k + \nabla_{\tilde{x}}\ell_k^T s + \frac{1}{2}s^T H_k s + \frac{\rho_k}{2}\|D_k(G_k + \nabla G_k^T s)\|^2 \\ subject\ to & \|s\| \leq \delta_k, \end{array} \tag{2.1}$$

where H_k is the Hessian of the Lagrangian function (1.3) or an approximation to it and $\delta_k > 0$ is a trust-region radius. We represent the quadratic form of the objective function of Problem (1.9) by $q_k(s_k)$. For complete survey see [29, 30].

It is not necessary to obtain a very accurate approximation to the solution of the subproblem (2.1). Instead any approximation to the solution of the subproblem (2.1) can be used as long as the predicted decrease obtained by the step s_k is greater than or equal to a fraction of the predicted decrease obtained by the Cauchy step s_k^{cp}. This means that the following condition must be achieved

$$q_k(0) - q_k(s_k) \geq \varphi[q_k(0) - q_k(s_k^{cp})], \tag{2.2}$$

for some $\varphi \in (0, 1]$. The Cauchy step s_k^{cp} is defined as

$$s_k^{cp} = -\alpha_k^{cp}(\nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k), \tag{2.3}$$

where α_k^{cp} is given by

$$\alpha_k^{cp} = \begin{cases} \dfrac{\| \nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k \|^2}{(\nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k)^T B_k (\nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k)} \\ \quad \text{if } \dfrac{\| \nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k \|^3}{(\nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k)^T B_k (\nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k)} \leq \delta_k \\ \quad \text{and } (\nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k)^T B_k (\nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k) > 0, \\ \dfrac{\delta_k}{\| \nabla_{\tilde{x}}\ell_k + \rho_k \nabla G_k D_k G_k \|} \qquad \text{Otherwise,} \end{cases}$$

and $B_k = H_k + \nabla G_k D_k \nabla G_k^T$.

Therefore, we use a generalized dogleg algorithm introduced by [31] to compute s_k.

2.2. Testing s_k and Updating δ_k

Once s_k is evaluated, it needs to be tested to determine whether it will be accepted. To do that, a merit function is needed. We use the following augmented Lagrangian function as a merit function

$$\Phi(\tilde{x}, \lambda; \rho) = \ell(\tilde{x}, \lambda) + \frac{\rho}{2}\|D(\tilde{x})G(\tilde{x})\|^2. \tag{2.4}$$

To test the step, we need to estimate the Lagrange multiplier λ_{k+1}. Our way of estimating λ_{k+1} is presented in Step 5 of Algorithm (2.1) below. To test whether the point $(\tilde{x}_k + s_k,\ \lambda_{k+1})$ will be taken as a next iterate, an actual reduction and predicted reduction in the merit function must be defined.

The actual reduction in the merit function in moving from $(\tilde{x}_k, \lambda_k)$ to $(\tilde{x}_k + s_k, \lambda_{k+1})$ is defined as

$$Ared_k = \Phi(\tilde{x}_k, \lambda_k; \rho_k) - \Phi(\tilde{x}_k + s_k, \lambda_{k+1}; \rho_k).$$

Note that $Ared_k$ can be written as

$$Ared_k = \ell(\tilde{x}_k, \lambda_k) - \ell(\tilde{x}_{k+1}, \lambda_k) - \Delta\lambda_k^T G_{k+1} + \frac{\rho_k}{2}[G_k^T D_k G_k - G_{k+1}^T D_{k+1} G_{k+1}], \tag{2.5}$$

where $\Delta\lambda_k = (\lambda_{k+1} - \lambda_k)$.

The predicted reduction in the merit function is defined as

$$\begin{aligned} Pred_k = &-\nabla_{\tilde{x}}\ell(\tilde{x}_k, \lambda_k)^T s_k - \frac{1}{2}s_k^T H_k s_k - \Delta\lambda_k^T (G_k + \nabla G_k^T s_k) \\ &+\frac{\rho_k}{2}[\|D_k G_k\|^2 - \|D_k(G_k + \nabla G_k^T s_k)\|^2]. \end{aligned} \tag{2.6}$$

After evaluating s_k and estimating λ_{k+1}, the step is tested to know whether it is accepted by comparing $Pred_k$ against $Ared_k$. It is presented in Step 6 of Algorithm (2.1) below.

After accepting the step, we update the parameter ρ_k by using a scheme suggested by [32]. Our way of updating ρ_k is presented in Step 7 of Algorithm (2.1) below.

Finally, the algorithm is terminated when either $\|\nabla_{\tilde{x}}\ell_k\| + \|\nabla G_k D_k G_k\| \leq \varepsilon_1$, or $\|s_k\| \leq \varepsilon_2$ for some $\varepsilon_1, \varepsilon_2 > 0$.

2.3. The main algorithm

Master steps of our method is presented in the following algorithm.

Algorithm 2.1. *(The trust-region algorithm)*
Step 0. *(Initialization)*

Given $\tilde{x}_0 \in \Re^{n+1}$. *Compute* D_0. *Evaluate* λ_0. *Set* $\rho_0 = 1$. *Choose* ε_1, ε_2, α_1, α_2, η_1, *and* η_2 *such that* $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, $0 < \alpha_1 < 1 < \alpha_2$, *and* $0 < \eta_1 < \eta_2 < 1$. *Choose* $\delta_{\min}$, $\delta_{\max}$, *and* δ_0 *such that* $\delta_{\min} \leq \delta_0 \leq \delta_{\max}$. *Set* $k = 0$.

Step 1. *If* $\|\nabla_{\tilde{x}}\ell_k\| + \|\nabla G_k D_k G_k\| \leq \varepsilon_1$, *then stop.*
Step 2. *a)Compute the step* s_k *by solving subproblem (2.1).*

b) Set $\tilde{x}_{k+1} = \tilde{x}_k + s_k$.

Step 3. *If* $\|s_k\| \leq \varepsilon_2$, *then stop.*
Step 4. *Compute* D_{k+1} *given by (1.8).*
Step 5. *Compute* λ_{k+1} *by solving*

$$\begin{array}{cc} \min & \|\nabla F_{k+1} + \nabla G_{k+1}\lambda\|^2 \\ \textit{subject to} & \lambda \geq 0, \end{array} \tag{2.7}$$

Step 6. *If* $Ared_k < \eta_1 Pred_k$.

Set $\delta_k = \alpha_1\|s_k\|$ *and go to step 2.*
Else, if $\eta_1 Pred_k \leq Ared_k < \eta_2 Pred_k$.
Then accept the step: $\tilde{x}_{k+1} = \tilde{x}_k + s_k$.
Set $\delta_{k+1} = \max(\delta_k, \delta_{\min})$.
Else, accept the step: $\tilde{x}_{k+1} = \tilde{x}_k + s_k$.
Set $\delta_{k+1} = \min\{\delta_{\max}, \max\{\delta_{\min}, \alpha_2\delta_k\}\}$.
End if.

Step 7. *Set* $\rho_{k+1} = \rho_k$.

If

$$\frac{1}{2}(q_k(0) - q_k(s_k)) - \Delta\lambda_k^T(G_k + \nabla G_k^T s_k) < \sigma\|\nabla G_k D_k G_k\| \min\{\|\nabla G_k D_k G_k\|, \delta_k\}, \tag{2.8}$$

then set $\rho_{k+1} = 2\rho_k$.
End if.

Step 8. *Set* $k = k + 1$ *and go to Step 1.*

In the following section, we present a global convergence theory for the proposed trust-region algorithm.

3. Global Convergence Analysis

Let $\{(\tilde{x}_k, \lambda_k)\}$ be the sequence of points generated by Algorithm (2.1) and let Ω be a convex subset of $\Re^{n+1}$ that contains all iterates $\tilde{x}_k$ and $\tilde{x}_k + s_k$. On the set Ω, the following assumptions under which our global convergence theory is proved are imposed.

Assumptions:

- A_1. The functions $f_i(x)$, $i = 1, 2, ..., m$ and $h(x)$ are twice continuously differentiable for all $x \in \Omega$.
- A_2. All of $f_i(x)$, $\nabla f_i(x)$, $\nabla^2 f_i(x)$, $h(x)$, $\nabla h(x)$ for $i = 1, 2, ..., m$, are uniformly bounded in Ω.
- A_3. The sequence $\{\lambda_k\}$ is bounded.
- A_4. The sequence of Hessian matrices $\{H_k\}$ is bounded.

In the above assumptions, we do not presume $\nabla G_i(\tilde{x})$, $i = \{1, ..., m+p\}$ has inverse for all $\tilde{x} \in \Omega$. So, we may have other kinds of stationary points. They are presented in the following three definitions.

Definition 3.1 (Fritz John Point). *A point $\tilde{x}_*$ is called a Fritz John point if there exist γ_* and λ_* not all zeros, such that*

$$\begin{aligned}
&\gamma_* \nabla F(\tilde{x}_*) + \nabla G(\tilde{x}_*)\lambda_* = 0, \\
&D_* G(\tilde{x}_*) = 0, \\
&(\lambda_*)_i G_i(\tilde{x}_*) = 0, \qquad i = 1, ..., m+p, \\
&\gamma_*,\ (\lambda_*)_i \geq 0, \qquad i = 1, ..., m+p.
\end{aligned}$$

The above conditions are called Fritz John conditions, see [33].

If $\gamma_* \neq 0$, then the Fritz John conditions correspond with the KKT conditions (1.4)-(1.7) and the point $(\tilde{x}_*, \frac{\lambda_*}{\gamma_*})$ is called a KKT point.

Definition 3.2 (Infeasible Fritz John Point). *A point $\tilde{x}_*$ is called an infeasible Fritz John point if there exist γ_* and λ_*, not all zeros, such that*

$$\begin{aligned}
&\gamma_* \nabla F(\tilde{x}_*) + \nabla G(\tilde{x}_*)\lambda_* = 0, \\
&\nabla G(\tilde{x}_*) D(\tilde{x}_*) G(\tilde{x}_*) = 0 \quad \textit{but} \ \ \|D(\tilde{x}_*) G(\tilde{x}_*)\| > 0, \\
&(\lambda_*)_i G_i(\tilde{x}_*) \geq 0, \qquad i = 1, ..., m+p, \\
&\gamma_*,\ (\lambda_*)_i \geq 0, \qquad i = 1, ..., m+p.
\end{aligned}$$

The above conditions are called the infeasible Fritz John conditions, see [33].

If $\gamma_* \neq 0$, then the point $(\tilde{x}_*, \frac{\lambda_*}{\gamma_*})$ is called an infeasible KKT point and the infeasible Fritz John conditions are called the infeasible KKT conditions.

Definition 3.3 (Infeasible Mayer-Bliss Point). *A point $\tilde{x}_*$ is called an infeasible Mayer-Bliss if*

$$\begin{aligned}
&\nabla G(\tilde{x}_*) D(\tilde{x}_*) G(\tilde{x}_*) = 0, \\
&\|D(\tilde{x}_*) G(\tilde{x}_*)\| > 0.
\end{aligned}$$

The above conditions are called the infeasible Mayer-Bliss conditions, see [34].

The conditions stated in Definitions (3.1)-(3.3) are called stationary conditions of problem (1.1) and the point that satisfies any of these stationary conditions is called a stationary point.

The following three lemmas provide conditions equivalent to the conditions given in Definitions (3.1)-(3.3).

Lemma 3.1. *Suppose that assumptions A_1-A_4 hold. A subsequence $\{\tilde{x}_{k_i}\}$ of the iteration sequence asymptotically satisfies the infeasible Fritz John conditions if it satisfies:*

$$1)\ \lim_{k_i \to \infty} \|D_{k_i} G(\tilde{x}_{k_i})\| > 0;$$
$$2)\ \lim_{k_i \to \infty} (\nabla G(\tilde{x}_{k_i}) D_{k_i} G(\tilde{x}_{k_i}) = 0.$$

Proof. See Lemma 4.1 of [20]. □

Lemma 3.2. *Suppose that assumptions A_1-A_4 hold. A subsequence $\{\tilde{x}_{k_i}\}$ of the iteration sequence asymptotically satisfies the feasible Fritz John's conditions if it satisfies:*

$$1)\ \textit{For all } k_i,\ \|D_{k_i} G_{k_i}\| > 0 \textit{ and } \lim_{k_i \to \infty} D_{k_i} G_{k_i} = 0;$$
$$2)\ \textit{For } k_i \to \infty,\ \lim_{k_i \to \infty} \left\{ \min_{s \in \Re^{n+1}} \frac{\|D_{k_i}(G_{k_i} + \nabla G_{k_i}^T s)\|^2}{\|D_{k_i} G_{k_i}\|^2} \right\} = 1.$$

Proof. See Lemma 4.2 of [20]. □

Lemma 3.3. *Suppose that assumptions A_1-A_4 hold. A subsequence $\{\tilde{x}_{k_i}\}$ of the iteration sequence asymptotically satisfies the infeasible Mayer-Bliss conditions if it satisfies:*

$$1)\ \lim_{k_i \to \infty} \|D_{k_i} G_{k_i}\| > 0;$$
$$2)\ \lim_{k_i \to \infty} \left\{ \min_{s \in \Re^{n+1}} \|D_{k_i}(G_{k_i} + \nabla G_{k_i}^T s)\|^2 \right\} = \lim_{k_i \to \infty} \|D_{k_i} G_{k_i}\|^2.$$

Proof. See Lemma 4.3 of [20]. □

Lemma 3.4. *Assume A_1 and A_2. Then $D(\tilde{x})G(\tilde{x})$ is Lipschitz continuous in Ω.*

Proof. See Lemma 4.1 of [18]. □

From the above lemma, we conclude that $G(\tilde{x})^T D(\tilde{x}) G(\tilde{x})$ is differentiable and $\nabla G(\tilde{x}) D(\tilde{x})$ $G(\tilde{x})$ is Lipschitz continuous in Ω.

Lemma 3.5. *At any iteration k, let $V(x_k) \in \Re^{m+p \times m+p}$ be a diagonal matrix whose diagonal entries are*

$$(v_k)_i = \begin{cases} 1 & \textit{if } (G_k)_i < 0 \textit{ and } (G_{k+1})_i \geq 0, \\ -1 & \textit{if } (G_k)_i \geq 0 \textit{ and } (G_{k+1})_i < 0, \\ 0 & \textit{otherwise}, \end{cases} \tag{3.1}$$

where $i = 1, ..., m+p$. Then

$$D_{k+1} = D_k + V_k. \tag{3.2}$$

Proof. See Lemma 5.1 of [19]. □

Lemma 3.6. *Assume A_1 and A_2. At any iteration k, there exists a positive constant K_1 independent of k, such that*

$$\|V_k G_k\| \leq K_1 \|s_k\|, \tag{3.3}$$

where $V_k \in \Re^{m+p \times m+p}$ is the diagonal matrix whose diagonal entries are defined in (3.1).

Proof. See Lemma 5.2 of [19]. □

The following lemma gives upper bound on the difference between the actual reduction and the predicted reduction.

Lemma 3.7. *Suppose that assumptions A_1-A_4 hold, then there exists a constant $K_2 > 0$ that does not depend on k, such that*

$$| \, Ared_k - Pred_k \, | \leq K_2 \rho_k \|s_k\|^2. \tag{3.4}$$

Proof. From (2.5) and (3.2), we have

$$Ared_k = \ell(\tilde{x}_k, \lambda_k) - \ell(\tilde{x}_{k+1}, \lambda_k) - \Delta\lambda_k^T G_{k+1} + \frac{\rho_k}{2}[G_k^T D_k G_k - G_{k+1}^T (D_k + V_k) G_{k+1}].$$

From the above equation, (2.6), and using Cauchy-Schwarz inequality, we have

$$\begin{aligned} &| \, Ared_k - Pred_k \, | \\ \leq & \, | \, \ell(\tilde{x}_k, \lambda_k) + \nabla\ell(\tilde{x}_k, \lambda_k)^T s_k + \frac{1}{2} s_k^T H_k s_k - \ell(\tilde{x}_{k+1}, \lambda_k) \, | + | \, \Delta\lambda_k^T [(G_k + \nabla G_k^T s_k) - G_{k+1}] \, | \\ & + \frac{\rho_k}{2} \, | \, (G_k + \nabla G_k^T s_k)^T D_k (G_k + \nabla G_k^T s_k) - G_{k+1}^T (D_k + V_k) G_{k+1} \, | \, . \end{aligned}$$

Hence,

$$\begin{aligned} &|Ared_k - Pred_k| \\ \leq & \, | \, \frac{1}{2} s_k^T (H_k - \nabla^2 \ell(\tilde{x}_k + \xi_1 s_k, \lambda_k)) s_k \, | + | \, \Delta\lambda_k^T (\nabla G_k - \nabla G(\tilde{x}_k + \xi_2 s_k))^T s_k \, | \\ & + \rho_k \, | \, [(\nabla G_k - \nabla G(\tilde{x}_k + \xi_2 s_k)) D_k G_k]^T s_k \, | + \frac{\rho_k}{2} \, | \, s_k^T \nabla G(\tilde{x}_k + \xi_2 s_k) D_k \nabla G(\tilde{x}_k + \xi_2 s_k)^T s_k \, | \\ & + \frac{\rho_k}{2} \, | \, s_k^T \nabla^2 G(\tilde{x}_k + \xi_2 s_k) V_k G(\tilde{x}_k + \xi_2 s_k) s_k \, | + \rho_k \, | \, (\nabla G(\tilde{x}_k + \xi_2 s_k) V_k G_k)^T s_k \, | + \frac{\rho_k}{2} \, | \, G_k^T V_k G_k \, |, \end{aligned}$$

for some ξ_1 and $\xi_2 \in (0, 1)$. Using the assumptions $A_1 - A_4$, and Inequality (3.3), the proof follows. □

Lemma 3.8. *Suppose that assumptions A_1-A_4 hold. Then for all $k > \bar{k}$, there exists a positive constant K_3 independent of the iterates such that,*

$$\begin{aligned} & q_k(0) - q_k(s_k) \\ \geq & K_3 \, \| \, \nabla_{\tilde{x}} \ell(\tilde{x}_k, \lambda_k) + \rho_k \nabla G_k D_k G_k \, \| \min \left\{ \delta_k, \frac{\| \, \nabla\ell(\tilde{x}_k, \lambda_k) + \rho_k \nabla G_k D_k G_k \, \|}{\| \, B_k \, \|} \right\}. \end{aligned} \tag{3.5}$$

Proof. Using (2.2) and (2.3), the proof is similar to the proof of Lemma (6.2) of [19] for another algorithm. □

From the way of updating the positive parameter ρ_k, we have

$$\frac{1}{2}(q_k(0) - q_k(s_k)) - \Delta\lambda_k^T (G_k + \nabla G_k^T s_k) < \sigma \|\nabla G_k D_k G_k\| \min\{\|\nabla G_k D_k G_k\|, \delta_k\}, \tag{3.6}$$

only when there exists an infinite subsequence of indices $\{k_i\}$ indexing iterates of acceptable steps that satisfy, for all $k \in \{k_i\}$ the sequence $\{\rho_k\}$ is unbounded.

The following two lemmas show that if $\rho_k \to \infty$, as $k \to \infty$, then a subsequence of the iteration sequence generated by Algorithm (2.1) satisfies Fritz John conditions or infeasible Mayer-Bliss conditions in the limit.

Lemma 3.9. *Suppose that assumptions A_1-A_4 hold. If $\rho_k \to \infty$, as $k \to \infty$ and there exists a subsequence $\{k_j\}$ of indices indexing iterates that satisfy $\|D_k G_k\| \geq \epsilon_1 > 0$ for all $k \in \{k_j\}$, then a subsequence of the iteration sequence indexed $\{k_j\}$ satisfies the infeasible Mayer-Bliss conditions in the limit.*

Proof. The proof is by contradiction. Suppose there exists no subsequence of the sequence of iterates that satisfies the infeasible Mayer-Bliss conditions in the limit. Using Lemma (3.3), then for all k we have, $|\ \|D_k G_k\|^2 - \|D_k(G_k + \nabla G_k^T s_k)\|^2\ | \geq \varepsilon_1$ and from Definition (3.3), we have $\|\nabla G_k D_k G_k\| \geq \varepsilon_2$ for some $\varepsilon_2 > 0$. Since $\rho_k \to \infty$, then there exists infinite number of acceptable iterates at which Inequality (3.6) holds. We consider two cases:

i) If $\|D_k G_k\|^2 - \|D_k(G_k + \nabla G_k^T s_k)\|^2 \geq \varepsilon_1$, we have

$$\rho_k\{\|D_k G_k\|^2 - \|D_k(G_k + \nabla G_k^T s_k)\|^2\} \geq \rho_k \varepsilon_1 \to \infty. \tag{3.7}$$

Since

$$\begin{aligned}
&\frac{1}{2}(q_k(0) - q_k(s_k)) - \Delta\lambda_k^T(G_k + \nabla G_k^T s_k)\\
&= -\frac{1}{2}\nabla_{\tilde{x}_k}\ell(\tilde{x}_k, \lambda_k)^T s_k - \frac{1}{4}s_k^T H_k s_k - \frac{1}{2}\Delta\lambda_k^T(G_k + \nabla G_k^T s_k)\\
&\quad + \frac{\rho_k}{4}\{\|D_k G_k\|^2 - \|D_k(G_k + \nabla G_k^T s_k)\|^2\}.
\end{aligned}$$

Using assumptions A_2 -A_4, and Inequality (3.7), we have $[\frac{1}{2}(q_k(0) - q_k(s_k)) - \Delta\lambda_k^T(G_k + \nabla G_k^T s_k)] \to \infty$. Hence, the left hand side of Inequality (3.6) tends to infinity as $k \to \infty$, while the right hand side goes to zero. This gives a contradiction in this case.

ii) If $\|D_k G_k\|^2 - \|D_k(G_k + \nabla G_k^T s_k)\|^2 \leq -\varepsilon_1$. Because $\rho_k \to \infty$ as $k \to \infty$, we have

$$\rho_k\{\|D_k G_k\|^2 - \|D_k(G_k + \nabla G_k^T s_k)\|^2\} \leq -\rho_k \varepsilon_1 \to -\infty. \tag{3.8}$$

Similar to the case (i), we have

$$[\frac{1}{2}(q_k(0) - q_k(s_k)) - \Delta\lambda_k^T(G_k + \nabla G_k^T s_k)] \to -\infty.$$

Since $Pred_k = (q_k(0) - q_k(s_k)) - \Delta\lambda_k^T(G_k + \nabla G_k^T s_k)$, we have $Pred_k \to -\infty$. This gives a contradiction with $Pred_k > 0$. These two contradictions prove the lemma. □

The following lemma studies the case when $\lim inf_{k\to\infty}\|D_k G_k\| = 0$ and $\rho_k \to \infty$ as $k \to \infty$.

Lemma 3.10. *Suppose that assumptions A_1-A_4 hold. If $\rho_k \to \infty$, as $k \to \infty$, and there exists a subsequence $\{k_j\}$ of iterates that satisfies $\|D_k G_k\| > 0$ for all $k \in \{k_j\}$ and $\lim_{k_j\to\infty}\|D_{k_j} G_{k_j}\| = 0$, then a subsequence of the sequence of iterates indexed $\{k_j\}$ satisfies Fritz John's conditions in the limit.*

Proof. Let the subsequence $\{k_j\}$ be renamed to $\{k\}$ to simplify the notations avoiding double indexes. The proof is by contradiction. Assume there exists no subsequence that satisfies Fritz John's conditions in the limit. Hence, using Lemma (3.2), there exists a constant ε_3 such that for all k sufficiently large,

$$\frac{|\ \|D_k G_k\|^2 - \|D_k(G_k + \nabla G_k^T s_k)\|^2\ |}{\|D_k G_k\|^2} \geq \varepsilon_3. \tag{3.9}$$

We consider three cases:

i) If $\lim inf_{k\to\infty} \frac{s_k}{\|D_kG_k\|} = 0$, Inequality (3.9) gives contradiction.

ii)If $\lim sup_{k\to\infty} \frac{s_k}{\|D_kG_k\|} = \infty$. From the way of computing the trial steps, we have

$$\nabla_{\tilde{x}_k}\ell(\tilde{x}_k,\lambda_k) + \rho_k\nabla G_k D_k G_k = -(B_k + \mu_k I)s_k, \tag{3.10}$$

where $\mu_k \geq 0$ is the Lagrange multiplier of the trust region constraint. Since $B_k = H_k + \rho_k\nabla G_k D_k \nabla G_k^T$ and using (3.10), then Inequality (3.5) can be written as follows

$$\begin{aligned} &q_k(0) - q_k(s_k)\\ \geq &K_3 \parallel \nabla_{\tilde{x}_k}\ell_k + \rho_k\nabla G_k D_k G_k \parallel \min\left\{\delta_k, \frac{\parallel [\frac{1}{\rho_k}H_k + (\nabla G_k D_k \nabla G_k^T + \frac{\mu_k}{\rho_k}I)]s_k \parallel}{\parallel \frac{1}{\rho_k}H_k + \nabla G_k D_k \nabla G_k^T \parallel}\right\}. \end{aligned} \tag{3.11}$$

Because $\rho_k \to \infty$, as $k \to \infty$, there exists an infinite number of acceptable steps such that Inequality (3.6) holds. But Inequality (3.6) can be written as

$$\frac{1}{2}(q_k(0) - q_k(s_k)) - \Delta\lambda_k^T(G_k + \nabla G_k^T s_k) < \sigma\|\nabla G_k\|^2\|D_kG_k\|^2. \tag{3.12}$$

From Inequalities (3.11) and (3.12), we have

$$\begin{aligned} &\frac{K_3}{2} \parallel \nabla_{\tilde{x}_k}\ell_k + \rho_k\nabla G_k D_k G_k \parallel \min\left\{\delta_k, \frac{\parallel [\frac{1}{\rho_k}H_k + (\nabla G_k D_k \nabla G_k^T + \frac{\mu_k}{\rho_k}I)]s_k \parallel}{\parallel \frac{1}{\rho_k}H_k + \nabla G_k D_k \nabla G_k^T \parallel}\right\}\\ &\qquad -\Delta\lambda_k^T(G_k + \nabla G_k^T s_k) < \sigma b_1^2\|D_kG_k\|^2 \end{aligned} \tag{3.13}$$

where $b_1 = sup_{x\in\Omega}\|\nabla G_k\|$. Since

$$\begin{aligned} &\Delta\lambda_k^T(G_k + \nabla G_k^T s_k) = \Delta\lambda_k^T G_k + \Delta\lambda_k^T\nabla G_k^T s_k\\ &= (\lambda_{k+1}D_{k+1} - \lambda_k D_k)^T G_k + \Delta\lambda_k^T\nabla G_k^T s_k\\ &= (\lambda_{k+1}(D_k + V_k) - \lambda_k D_k)^T G_k + \Delta\lambda_k^T\nabla G_k^T s_k\\ &\leq \|\Delta\lambda_k\|\|D_kG_k\| + \|\lambda_{k+1}\|\|V_kG_k\| + \|\Delta\lambda_k^T\nabla G_k^T\|\|s_k\|\\ &\leq \|\Delta\lambda_k\|\|D_kG_k\| + K_1\|\lambda_{k+1}\|\|s_k\| + \|\Delta\lambda_k^T\nabla G_k^T\|\|s_k\|\\ &\leq \|\Delta\lambda_k\|\|D_kG_k\| + [K_1\|\lambda_{k+1}\| + \|\Delta\lambda_k^T\nabla G_k^T\|]\|s_k\|\\ &\leq \|\Delta\lambda_k\|\|D_kG_k\| + [K_1\|\lambda_{k+1}\| + \|\Delta\lambda_k^T\nabla G_k^T\|]\delta_k. \end{aligned}$$

Then, from Inequality (3.13) and the above inequality we have

$$\begin{aligned} &\frac{K_3}{2} \parallel \nabla_{\tilde{x}_k}\ell_k + \rho_k\nabla G_k D_k G_k \parallel \min\left\{\delta_k, \frac{\parallel [\frac{1}{\rho_k}H_k + (\nabla G_k D_k \nabla G_k^T + \frac{\mu_k}{\rho_k}I)]s_k \parallel}{\parallel \frac{1}{\rho_k}H_k + \nabla G_k D_k \nabla G_k^T \parallel}\right\}\\ &\qquad -\|\Delta\lambda_k\|\|D_kG_k\| - [K_1\|\lambda_{k+1}\| + \|\Delta\lambda_k^T\nabla G_k^T\|]\delta_k < \sigma b_1^2\|D_kG_k\|^2. \end{aligned}$$

Hence, if we divided the above inequality by $\|D_kG_k\|$, we obtain

$$\begin{aligned} &\frac{K_3}{2} \parallel \nabla_{\tilde{x}_k}\ell_k + \rho_k\nabla G_k D_k G_k \parallel \min\left\{\frac{\delta_k}{\|D_kG_k\|}, \frac{\parallel [\frac{1}{\rho_k}H_k + (\nabla G_k D_k \nabla G_k^T + \frac{\mu_k}{\rho_k}I)]s_k \parallel}{\parallel \frac{1}{\rho_k}H_k + \nabla G_k D_k \nabla G_k^T \parallel \|D_kG_k\|}\right\}\\ &\qquad -\|\Delta\lambda_k\| - [K_1\|\lambda_{k+1}\| + \|\Delta\lambda_k^T\nabla G_k^T\|]\frac{\delta_k}{\|D_kG_k\|} < \sigma b_1^2\|D_kG_k\|. \end{aligned} \tag{3.14}$$

The right hand side of the above inequality goes to zero as $k \to \infty$ and $\|\Delta\lambda_k\|$ is bounded. This implies that along the subsequence $\{k_i\}$ where $\lim_{k_i\to\infty} \frac{s_{k_i}}{\|D_{k_i}G_{k_i}\|} = \infty$,

$$\left\|\nabla_{\tilde{x}_{k_i}}\ell_{k_i} + \rho_{k_i}\nabla G_{k_i}D_{k_i}G_{k_i}\right\| \frac{\left\|[\frac{1}{\rho_{k_i}}H_{k_i} + (\nabla G_{k_i}D_{k_i}\nabla G_{k_i}^T + \frac{\mu_{k_i}}{\rho_{k_i}}I)]s_{k_i}\right\|}{\left\|\frac{1}{\rho_{k_i}}H_{k_i} + \nabla G_{k_i}D_{k_i}\nabla G_{k_i}^T\|\|D_{k_i}G_{k_i}\right\|},$$

is bounded. Therefore, asymptotically, either $\frac{s_{k_i}}{\|D_{k_i}G_{k_i}\|}$ lies in the null space of $\nabla G_{k_i}D_{k_i}\nabla G_{k_i}^T + \frac{\mu_{k_i}}{\rho_{k_i}}I$ or $\| \nabla_{\tilde{x}_{k_i}}\ell_{k_i} + \rho_{k_i}\nabla G_{k_i}D_{k_i}G_{k_i} \| \to 0$. The first possibility occurs only when $\frac{\mu_{k_i}}{\rho_{k_i}} \to 0$ as $k_i \to \infty$ and $s_{k_i}/\|D_{k_i}G_{k_i}\|$ lies in the null space of the matrix $\nabla G_{k_i}D_{k_i}\nabla G_{k_i}^T$ which contradicts assumption (3.9) and implies that a subsequence of the iteration sequence satisfies the Fritz John conditions in the limit. The second possibility implies as $k_i \to \infty$

$$\| \nabla_{\tilde{x}_{k_i}}\ell_{k_i} + \rho_{k_i}\nabla G_{k_i}D_{k_i}G_{k_i} \| \to 0.$$

Hence as $k_i \to \infty$, $\rho_{k_i}\|\nabla G_{k_i}D_{k_i}G_{k_i}\|$ must be bounded. Hence, we have $\nabla_{\tilde{x}_{k_i}}\ell_{k_i} = 0$. This implies that a subsequence of the iteration sequence satisfies the Fritz John conditions in the limit.

iii) If $\lim sup_{k\to\infty} \frac{s_k}{\|D_kG_k\|} < \infty$ and $\lim inf_{k\to\infty} \frac{s_k}{\|D_kG_k\|} > 0$. Therefore $\|s_k\| \to 0$. Hence, as in the second case, the right hand side of (3.14) goes to zero as $k \to \infty$. This implies that

$$\| \nabla_{\tilde{x}_k}\ell_k + \rho_k\nabla G_kD_kG_k \| \frac{\| (\nabla G_kD_k\nabla G_k^T + \frac{\mu_k}{\rho_k}I)s_k \|}{\| \nabla G_kD_k\nabla G_k^T \| \|D_kG_k\|} \to 0.$$

But this implies that asymptotically, either

$$\| \nabla_{\tilde{x}_k}\ell_k + \rho_k\nabla G_kD_kG_k \| \to 0 \quad \text{or} \quad \frac{\| (\nabla G_kD_k\nabla G_k^T + \frac{\mu_k}{\rho_k}I)s_k \|}{\| \nabla G_kD_k\nabla G_k^T \| \|D_kG_k\|} \to 0.$$

As the second case, the two possibilities imply that a subsequence of the iteration sequence satisfies the Fritz John conditions in the limit. This completes the proof. □

In the rest of this paper, we continue our analysis assuming that the positive parameter ρ_k is bounded. That is, we assume the existence of an integer $\bar{k}$ such that for all $k \geq \bar{k}$, $\rho_k = \bar{\rho} < \infty$ and

$$\frac{1}{2}(q_k(0) - q_k(s_k)) - \Delta\lambda_k^T(G_k + \nabla G_k^T s_k) \geq \sigma\|\nabla G_kD_kG_k\| \min\{\|\nabla G_kD_kG_k\|, \delta_k\}. \tag{3.15}$$

Lemma 3.11. *Suppose that assumptions A_1-A_4 hold. At any given iteration indexed k at which $\|\nabla_{\tilde{x}_k}\ell_k + \bar{\rho}\nabla G_kD_kG_k\| + \|\nabla G_kD_kG_k\| > \epsilon_1$, there exists a positive constant K_4 that depends on ϵ_1 but not depend on k, such that*

$$Pred_k \geq K_4\delta_k. \tag{3.16}$$

Proof. From (2.6), (3.15) and using Lemma (3.8), we have

$$\begin{aligned} Pred_k &= \frac{1}{2}(q_k(0) - q_k(s_k)) + [\frac{1}{2}(q_k(0) - q_k(s_k)) - \Delta\lambda_k^T(G_k + \nabla G_k^T s_k)] \\ &\geq \frac{K_3}{2} \| \nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_kD_kG_k \| \min\left\{\delta_k, \frac{\| \nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_kD_kG_k \|}{\|B_k\|}\right\} \\ &\quad +\sigma\|\nabla G_kD_kG_k\| \min\{\|\nabla G_kD_kG_k\|, \delta_k\}. \end{aligned} \tag{3.17}$$

We consider two cases:

i) If $\|\nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_k D_k G_k\| > \frac{\epsilon_1}{2}$ and using Inequality (3.17), then

$$\begin{aligned} Pred_k &\geq \frac{K_3}{2} \parallel \nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_k D_k G_k \parallel \min\left\{\delta_k, \frac{\parallel \nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_k D_k G_k \parallel}{\|B_k\|}\right\}, \\ &\geq \frac{K_3\epsilon_1}{4} \min\left\{1, \frac{\epsilon_1}{2b_2\delta_{\max}}\right\}\delta_k, \end{aligned}$$

where $\|B_k\| \leq b_2$ under assumptions $A_1 - A_4$.

ii) If $\|\nabla G_k D_k G_k\| > \frac{\epsilon_1}{2}$ and using Inequality (3.17), then we have

$$Pred_k \geq \frac{\sigma\epsilon_1}{2} \min\left\{\frac{\epsilon_1}{2\delta_{\max}}, 1\right\}\delta_k.$$

From the above two cases, the result follows by takeing $K_4 = \min\{\frac{K_3\epsilon_1}{4} \min\{1, \frac{\epsilon_1}{2b_2\delta_{\max}}\}, \frac{\sigma\epsilon_1}{2} \min\{\frac{\epsilon_1}{2\delta_{\max}}, 1\}\}$. □

Lemma 3.12. *Suppose that assumptions A_1-A_4 hold. If*

$$\|\nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_k D_k G_k\| + \|\nabla G_k D_k G_k\| > \epsilon_1,$$

then the condition $Ared_{k_j} \geq \tau_1 Pred_{k_j}$ will be satisfied for some finite j i.e., an acceptable step is found after finitely many trials.

Proof. Since $\|\nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_k D_k G_k\| + \|\nabla G_k D_k G_k\| > \epsilon_1$. From Inequalities (3.4) and (3.16), we have

$$\left| \frac{Ared_k}{Pred_k} - 1 \right| = \frac{| Ared_k - Pred_k |}{Pred_k} \leq \frac{K_2\bar{\rho}\delta_k^2}{K_4\delta_k} = \frac{K_2\bar{\rho}\delta_k}{K_4}.$$

Now as the trial step s_{k_j} gets rejected, δ_{k_j} becomes small and eventually after finite number of trials, (i.e., for j finite), the acceptance rule will be met. This completes the proof. □

Lemma 3.13. *Suppose that assumptions A_1-A_4 hold. If $\|\nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_k D_k G_k\| + \|\nabla G_k D_k G_k\| > \epsilon_1$, at a given iteration k, the j^{th} trial step satisfies*

$$\|s_{k^j}\| \leq \frac{(1-\eta_1)K_4}{2\bar{\rho}K_2}, \tag{3.18}$$

then it must be accepted.

Proof. We prove this lemma by contradiction. Assume that the step s_{k^j} is rejected and Inequality (3.18) holds. Then, from Inequalities (3.4) and (3.16) we have

$$(1-\eta_1) < \frac{| Ared_{k^j} - Pred_{k^j} |}{Pred_{k^j}} < \frac{K_2\bar{\rho}\|s_{k^j}\|^2}{K_4\|s_{k^j}\|} \leq \frac{(1-\eta_1)}{2}.$$

This gives a contradiction and proves the lemma. □

Theorem 3.1. *Suppose that assumptions A_1-A_4 hold. Then the sequence of iterates generated by the algorithm satisfies*

$$\liminf_{k\to\infty} \ [\ \|\nabla_{\tilde{x}}\ell_k\| + \|\nabla G_k D_k G_k\|\ ] = 0. \tag{3.19}$$

Proof. First, we prove that

$$\liminf_{k\to\infty}[\|\nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_k D_k G_k\| + \|\nabla G_k D_k G_k\|] = 0. \tag{3.20}$$

We prove this equation by contradiction. Suppose that, for all k, $\|\nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_k D_k G_k\| + \|\nabla G_k D_k G_k\| > \epsilon_1$. Consider a trial step indexed j of the iteration indexed k, $k \geq \bar{k}$, and such that $k^j \geq \bar{k}$. Using Lemma 3.11, we have for any acceptable step indexed k^j,

$$\Phi_{k^j} - \Phi_{k^j+1} = Ared_{k^j} \geq \eta_1 Pred_{k^j} \geq \eta_1 K_4 \delta_{k^j}. \tag{3.21}$$

As k goes to infinity the above inequality implies that

$$\lim_{k\to\infty} \delta_{k^j} = 0. \tag{3.22}$$

That is, the radius of the trust region is not bounded below.

If we consider an iteration indexed $k^j > \bar{k}$ and if the previous step was accepted; *i.e.* if $j = 1$, then $\delta_{k^1} \geq \delta_{\min}$. Hence δ_{k^j} is bounded in this case.

Now assume that $j > 1$. That is, there exists at least one rejected trial step. From Lemma (3.13), we have for the rejected trial step,

$$\|s_{k^i}\| > \frac{(1-\eta_1)K_4}{2\bar{\rho}K_2},$$

for all $i = 1, 2, ...j-1$. Since s_{k^i} is a rejected trial step, then from the way of updating the radius of trust region (see Step 6 Algorithm 2.1) and using the above inequality, we have

$$\delta_{k^j} = \alpha_1 \|s_{k^{j-1}}\| > \alpha_1 \frac{(1-\eta_1)K_4}{2\bar{\rho}K_2}.$$

Hence δ_{k^j} is bounded. But this contradicts (3.22). Therefore, the supposition is wrong. Hence,

$$\liminf_{k\to\infty}[\|\nabla_{\tilde{x}}\ell_k + \bar{\rho}\nabla G_k D_k G_k\| + \|\nabla G_k D_k G_k\|] = 0.$$

But this also implies (3.19). This completes the proof of the theorem. □

From the above theorem, we conclude that, given any ϵ_1, the algorithm terminates because $\|\nabla_{\tilde{x}}\ell_k\| + \|\nabla G_k D_k G_k\| < \epsilon_1$.

4. Numerical Experiments

In this section, we present the numerical results of the trust-region Algorithm (2.1) which have been performed on a laptop with Intel Core (TM)i7-2670QM CPU 2.2 GHz and 8 GB RAM. Algorithm (2.1) was implemented as a MATLAB code and run under MATLAB version 7.10.0.499 (R2010a)

Given a starting point $\tilde{x}_0 \in \Re^{n+1}$. We chose $\delta_{\min} = 10^{-3}$, $\delta_0 = \max(\|s_0^{ncp}\|, \delta_{\min})$, and $\delta_{\max} = 10^3\delta_0$. Also we chose $\eta_1 = 0.25$, $\eta_2 = 0.75$, $\alpha_1 = 0.5$, $\alpha_2 = 2$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-8}$. The computation terminates when $\|\nabla_{\tilde{x}}\ell_k\| + \|\nabla G_k D_k G_k\| \leq \varepsilon_1$ or $\|s_k\| \leq \varepsilon_2$.

The results are reported in Table 1 where the mini-max test problems are numbered in the same way as in [35]. For comparison, we have included the corresponding results of the number of iteration (iter) and the number of function evaluation (nfunc) obtained by Method in [35] (Table 1). For all mini-max problems, these algorithms achieved the same optimal solution.

5. Concluding Remarks

In this paper, we propose a trust region Algorithm 2.1 for solving mini-max Problem (1.1). To study the global convergence of the proposed algorithm four Assumptions $A_1 - A_4$ are imposed. Under these Assumptions a number of important lemmas are stated and proved. To validate the theoretical analysis of the algorithm, a number of mini-max problems are reported and compared with the method in [35].

Table 5.1: Comparison of Method in [35] with Algorithm 2.1.

Problem Name	Starting point	Method in [35]		Algorithm 2.1	
		iter	nfunc	iter	nfunc
Problem 1 [35]	(1, -0.1)	5	5	3	4
	(0, 0)	6	6	4	5
	(2, 2)	6	6	4	5
	(4, -4)	16	16	16	17
Problem 2 [35]	(3,1)	17	17	10	12
	(1, 3)	7	7	4	5
Problem 3 [35]	(3,1)	13	13	8	9
	(50, 0.05)	9	9	5	6
Problem 4 [35]	(2.1,1.9)	7	8	10	11
	(1.9, 2.1)	7	10	7	9
	(2, 4)	8	9	5	6
	(4, 2)	10	11	11	12
Problem 5 [35]	(0,0,0,0)	10	11	8	9
	(0,1,1,0)	10	13	8	9
	(2,2,5,0)	10	10	8	10
	(1,3,3,1)	10	10	7	8
	(-2,1,1,-2)	10	10	9	10
Problem 6 [35]	(0, 1)	4	4	5	6
	(3, 1)	7	7	5	6
Problem 7 [35]	(1,2,0,4,0,1,1)	15	33	15	20
	(3, 3,0,5,1,3,0)	18	42	16	21
Problem 8 [35]	(-1.2,1)	14	46	10	20
Problem 9 [35]	(50,0.05)	8	8	9	11
	(1,1.1)	11	20	9	11
Problem 10 [35]	(1.41831,-4.79462)	8	8	10	12
Problem 11 [35]	(2,3,5,5,1,2,7,3,6,10)	8	8	7	8
sum		254	347	213	262

For future work, related important questions that have to be looked at are how to use a secant approximation of the Hessian of the Lagrangian function in order to produce a more efficient algorithm and how to update the Lagrange multiplier which will reduce the cost of the computation of the steps.

Acknowledgement. The authors would like to thank the anonymous referees for their valuable comments and suggestions which have helped to greatly improve this paper.

References

[1] V. Dem'yanov and V. Malozemov, Introduction to Minimax, *New York*, Wiley (1974).
[2] D. Du and P. Pardalos, Minimax and applications, *Dordrecht*, Kluwer (1995).
[3] C. Charalambous and A. Conn, An eficient method to solve the minimax problem directly, *SIAM J. Numer. Anal.*, **15** (1978), 162-187 .
[4] G. DiPillo, L. Grippo, and S. Lucidi, A smooth method for the finite minimax problem, *Mathematical Programming*, **60** (1993), 187-214 .
[5] J. Jian and M. Chao, A sequential quadratically constrained quadratic programming method for unconstrained minimax problems, *J. Math. Anal. Appl.*, **362** (2010), 34-45.
[6] E. Polak, D. Mayne, and J. Higgins, Superlinearly convergent algorithm for minimax problems, *Journal of Optimization Theory and Application*, **69** (1991), 407-439 .
[7] L. Qi and W. Sun, An iterative method for the minimax problem, in minimax and applications, *Kulwer Academic Publisher*, Boston, 1995, 55-67.
[8] F. Wang and K. Zhang, A hybrid algorithm for nonlinear minimax problems, *Ann. Oper. Res.*, **164** (2008), 167-191 .
[9] S. Xu, Smoothing method for minimax problems, *Comput. Optim. Appl.*, **20**:3 (2001), 267-279.
[10] Y. Xue, The sequential quadratic programming method for solving minimax problem, *Science and Systems Engineering*, **22**:3 (2002), 355-364.
[11] Yuan. Y, On the superlinear convergence of a trust-region algorithm for nonsmooth optimization, Mathematical Programming, **31** (1985), 269-285 .
[12] Yi.X, The sequential quadratic programming method for solving minimax problem, *Journal of Systems Science and Mathematical Sciences*, **22** (2002), 355-364.
[13] Yi. X, A Newton like algorithm for solving minimax problem, *Journal on Numerical Methods and Computer Applications 2*, 108-115(2004).
[14] M. Gaudioso and M. Monaco, A bundle type approach to the unconstrained minimization of convex nonsmooth functions, *Mathematical Programming*, **23**:1 (1982), 216-226.
[15] J. Zowe, Nondifferentiable optimziation: a motivation and a short introduction into the subgradient and the bundle concept, *Computational Mathematical Programming*, Springer, New York (1985).
[16] F. Ye, H. Liu, S. Zhou, and S. Liu, A smoothing trust-region Newton-CG method for minimax problem, *Applied Mathematics and Computation*, **199**:2 (2008), 581-589.
[17] A. Fiacco and G. McCormick, Nonlinear Programming: Sequential unconstrained minimization techniques, *John Wiley and Sons*, New York, (1968).
[18] J. Dennis, M. El-Alem, and K. Williamson, A trust-region approach to nonlinear systems of equalities and inequalities, *SIAM J. Optimization*, **9** (1999), 291-315.
[19] B. El-Sobky, A global convergence theory for an active trust region algorithm for solving the general nonlinear programming problem, *Applied Mathematics and Computation*, **144**:1 (2003), 127-157.
[20] B. El-Sobky, A Multiplier active-set trust-region algorithm for solving constrained optimization problem, *Applied Mathematics and Computation*, **219** (2012), 127-157.
[21] B. El-Sobky and Y. Abouel-Naga, Multi-objective economic emission load dispatch problem with trust-region strategy, *Electric Power Systems Research*, **108** (2014), 254-269.
[22] M. Hestenes, Mutiplier and gradient methods, *JOTA*, **4** (1969), 303-320.
[23] T. Coleman and Y. Li, An interior trust region approach for nonlinear minimization subject to bounds, *SIAM J. Optimization*, **6** (1996), 418-445.
[24] J. Dennis, M. El-Alem, and M. Maciel, A global convergence theory for general trust-region-based algorithms for equality constrained optimization, *SIAM J. Optimization*, **7**:1 (1997), 177-207.
[25] B. El-Sobky, A new Convergence theory for trust-region algorithm for solving constrained optimization problems, *Applied Mathematics Science*, **7**:110 (2013), 5469-5489.
[26] B. El-Sobky, A penalty active-set trust-region algorithm for solving general nonlinear programming problem, *Sylwan*, **158**:9 (2014), 273-290.
[27] Y. Pei and D. Zhu, A trust-region algorithm combining line search filter technique for nonlinear constrained optimization. *International Journal of Computer Mathematics*, **91**:8 (2014), 1817?839.
[28] Y. Yuan, Recent advances in trust region algorithms, *Math. Program. Ser. B*, **151** (2015), 249-281.

[29] J. Dennis and R. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, *Prentice-Hall*, Englewood N.J. Cliffs, (1983).

[30] J. Moré and D. Sorensen, Computing a trust-region step, *SIAM J. Sci. Stats. Comput.*, **4** (1983), 553-572.

[31] T. Steihaug, The conjugate gradient method and trust-region in large scale optimization, *SIAM J. Numer. Anal.*, **20** (1983), 3.

[32] Y. Yuan, On the convergence of a new trust region algorithm, *Numer. Math.* **70** (1995), 515-539.

[33] O. Mangasarian, Nonlinear Programming, *McGraw-Hill Book Company*, New York, (1969).

[34] M. El-Alem, M. Abdel-Aziz, and A. El-Bakry, A projected Hessian Gauss-Newton algorithm for solving systems of nonlinear equations and inequalities, *Inter.J. of Math. and Math. Sc.*, **25**:6 (2001), 397-409.

[35] B. Rustem and Q. Nguyen, An algorithm for the inequality-constrained discrete min-max problem, *SIAM J. Optim.*, **8**:1 (1998), 265-283.