

An Active-Set Trust-Region Algorithm for Solving Constrained Multi-Objective Optimization Problem

Bothina El-Sobky

Department of Mathematics, Faculty of Science
Alexandria University, Alexandria, Egypt
bothinaelsobky@yahoo.com

Yousria Abo-Elnaga

Department of Basic Science
Higher Technological Institute
Tenth of Ramadan City, Egypt

Abstract

In this paper, the multi-objective optimization problem with equality and inequality constraints is solved by using interactive sequential optimization technique (I-SHOT) which convert the constrained multi-objective optimization problem to a single objective optimization problem with equality and inequality constraints. A trust-region algorithm for solving a single objective optimization problem with equality and inequality constraints is used. In this algorithm, an active set strategy is used together with a reduced Hessian technique is used to convert the computation of the trial step to two easy trust-region subproblems and the conjugate-gradient method is used to compute the trial step.

A Matlab implementation of our algorithm was used in solving one case studies and the result are reported

Keywords: Multi-objective optimization, Interactive sequential optimization technique, Pareto optimal solution, Single objective optimization problem, Trust-region, Active-set

1 Introduction

The multi-objective optimization problem with equality and inequality constraints is

$$\begin{aligned}
 & \text{minimize} && f_j(x) && j = 1, \dots, Q \\
 & \text{subject to} && Ci(x) \leq 0, && i = 1, 2, \dots, p, \\
 & && Ce(x) = 0, && e = 1, 2, \dots, m.
 \end{aligned} \tag{1.1}$$

We suppose that $f_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $Ce : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and $Ci : \mathbb{R}^n \rightarrow \mathbb{R}^p$ are twice continuously differentiable. We assume that $n \geq m + p$ and no restriction is imposed on p .

To solve this problem we use interactive sequential optimization method (I-SHOT) to convert it to a single objective optimization problem with equality and inequality constraints. To solve the single objective optimization problem we use a trust-region algorithm. In this algorithm, an active set strategy is used together with a reduced Hessian technique to convert the single objective optimization problem with equality and inequality constraints to equality constrained optimization problem and the computation of the trial step to two easy trust-region subproblems similar to those for the unconstrained case. A conjugate gradient method for equality constrained optimization problem is used to compute a trial step such that this method is suitable to solve a large-scale optimization problem.

The paper is organized as follows. In Section 2, we describe interactive sequential optimization technique (I-SHOT). In Section 3, we introduce how to solve a single optimization problem with equality and inequality constraints by using a trust-region algorithm. In Section 4, we introduce a global convergence theory for this algorithm. Section 5 contains a Matlab implementation of our algorithm. Finally, Section 6 contains concluding remarks.

2 Interactive Sequential Optimization Technique (I-SHOT)

Interactive approach is one of the most important methods in solving multi-objective optimization problems. In this approach the decision maker (DM) works together with the analyst or interactive computer programming. One can say that the analyst tries to determine the preference structure to the DM in an interactive way. The basic steps in interactive algorithms are, find an initial solution, interact with the DM, and then obtain the new solution. If the new solution (or one of them) is acceptable to the DM, STOP.

Otherwise we make a new interaction with the DM, the main problem in the most of known interactive methods is the computational efforts and the preference point. So in this paper we introduce one of the famous approach in solving multi-objective optimization problem is Interactive sequential optimization technique (I-SHOT) ([1]). In this method, we convert the constrained multi-objective optimization problem 1.1 to a single optimization problem with equalities and inequalities constraints.

We introduce some definitions before we discuss the existence of Pareto solutions.

Definition 2.1 A decision vector $x^* \in S$ is Pareto optimal if there does not exist another decision vector $x \in S$ such that $f_j(x) < f_j(x^*)$ for all $j = 1, 2, \dots, Q$ and $f_j(x) \leq f_j(x^*)$ for at least one objective function f_l . [9]

Definition 2.2 A decision vector $x^* \in S$ is weakly Pareto optimal if there does not exist another decision vector $x \in S$ such that $f_j(x) \leq f_j(x^*)$ for all $j = 1, 2, \dots, Q$ and $f_j(x) < f_j(x^*)$ for at least one objective function f_l . [9]

By using interactive approach, we transform the Problem (1.1) as follows:

$$\begin{aligned} & \text{minimize} && F(x) = \sum_{j=1}^k u_j \left(\frac{f_j(x) - \varepsilon_{j_{good}}}{\varepsilon_{j_{bad}} - \varepsilon_{j_{good}}} \right) \\ & \text{subject to} && f_j(x) \leq \varepsilon_j \quad j = 1, \dots, Q, \\ & && Ci(x) \leq 0, \quad i = 1, 2, \dots, p, \\ & && Ce(x) = 0, \quad e = 1, 2, \dots, m, \end{aligned} \tag{2.1}$$

where

$$u_j(x) = \begin{cases} U_j + \frac{(1-\hat{U})}{\hat{U}} U_j & \text{if } \hat{U} \geq 1, \\ U_j & \text{if } \hat{U} = 1, \\ U_j + \frac{(1-\hat{U})}{k-\hat{U}} (1 - U_j) & \text{if } \hat{U} \leq 1, \end{cases} \tag{2.2}$$

such that $U_j = \text{random}(0, 1)$ and $\hat{U} = \sum_{j=1}^k U_j$.

2.1 I-SHOT Algorithm

In this subsection, we introduce the formal description of I-SHOT algorithm. A formal description of our algorithm is presented in the following algorithm.

Algorithm 2.1 (I-SHOT algorithm)

Step 1. (Compute ε_j and $\hat{\varepsilon}_j$)

a) Compute ε_j by solving problem (1.1) for all $j = 1, \dots, Q$.

b) Compute $\hat{\varepsilon}_j$ by solving the following problem for all $j = 1, \dots, Q$.

$$\begin{aligned} & \text{maximize} && f_j(x) \quad j = 1, \dots, Q \\ & \text{subject to} && Ci(x) \leq 0, \quad i = 1, 2, \dots, p, \\ & && Ce(x) = 0, \quad e = 1, 2, \dots, m. \end{aligned}$$

Step2. (Asked the DM)

The DM asked to select the targets of $\varepsilon_j \leq \varepsilon_{j_{good}} \leq \hat{\varepsilon}_j$ and $\varepsilon_j \leq \varepsilon_{j_{bad}} \leq \hat{\varepsilon}_j$ values for the first iteration only. For subsequent iterations, these targets may be obtained according to step 5. The DM is asked to choose the quantity of the number of Pareto optimal solutions to be generated.

Step3. (Generate weighing vector)

The weighing vector is generated using equation (2.2)

Step4. (Solve the hybrid problem)

Solve the hybrid problem (2.1).

Step5. (Test for stopping)

If the DM is satisfied with the Pareto set then the process is stopped.
Otherwise go to step 6.

Step6. (Compute a new value for $\varepsilon_{j_{bad}}$ and $\varepsilon_{j_{good}}$)

$$\varepsilon_{j_{bad}}^{new} = \varepsilon_{j_{bad}} - \frac{\Delta\varepsilon_j}{\hat{r}}, \quad (2.3)$$

and

$$\varepsilon_{j_{good}}^{new} = \varepsilon_{j_{good}} + \frac{\Delta\varepsilon_j}{\hat{r}}, \quad (2.4)$$

where $\Delta\varepsilon_j = \varepsilon_{j_{bad}} - \varepsilon_{j_{good}}$ and \hat{r} is the reduction factor.

Set $\varepsilon_{j_{bad}} = \varepsilon_{j_{bad}}^{new}$ and $\varepsilon_{j_{good}} = \varepsilon_{j_{good}}^{new}$.

Go to Step 2.

Continuously until the DM is satisfied with the proper Pareto set is generated.

In the following section, we introduce how to solve a single optimization problem with equality and inequality constraints by using a trust-region algorithm.

3 A Trust-Region Algorithm for Solving A Single Optimization Problem

In this section, we consider the single optimization problem with equality and inequality constraints

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && Ci(x) \leq 0, \quad i = 1, 2, \dots, p, \\ & && Ce(x) = 0, \quad e = 1, 2, \dots, m. \end{aligned} \quad (3.1)$$

We suppose that $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$, $Ce : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$, and $Ci : \mathfrak{R}^n \rightarrow \mathfrak{R}^p$ are twice continuously differentiable. We assume that $n \geq m + p$ and no restriction is imposed on p .

Following Dennis, El-Alem, and Williamson [3], we define the indicator matrix $W(x) \in \mathfrak{R}^{p \times p}$, whose diagonal entries are

$$w_i(x) = \begin{cases} 1 & \text{if } Ci(x) \geq 0, \\ 0 & \text{if } Ci(x) < 0. \end{cases} \quad (3.2)$$

Using this matrix, the Problem (3.1) can be transformed to the following equality constrained optimization problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && 1/2Ci(x)^T W(x)Ci(x) = 0, \\ & && Ce(x) = 0, \end{aligned}$$

The above problem can be written as

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && g(x) = 0, \end{aligned} \quad (3.3)$$

where $g(x) = [Ce(x) \ 1/2Ci(x)^T W(x)Ci(x)]$.

The matrix $W(x)$ is discontinuous; however, the function $W(x)Ci(x)$ is Lipschitz continuous and the function $Ci(x)^T W(x)Ci(x)$ is continuously differentiable [3].

The Lagrangian function associated with Problem (3.3) is given by

$$L(x_k, \lambda_k) = f(x_k) + \lambda_k^T g(x_k), \quad (3.4)$$

where $\lambda_k \in \mathfrak{R}$ is the lagrange multiplier vector associated with equality constraint $g(x_k) \in \mathfrak{R}$.

The augmented Lagrangian is the function

$$\Phi(x, \lambda; r) = L(x, \lambda) + r\|g(x)\|^2, \quad (3.5)$$

where $r > 0$ is a parameter usually called the penalty parameter.

Observe that when Problem (3.1) is to Problem (3.3). A trust-region algorithm which we use is designed such that, if the point (x_*, λ_*) satisfies the first order necessary conditions for Problem (3.3). Then it also satisfies the KKT conditions for Problem (3.1), i.e. two Problems (3.3) and (3.2) are equivalent.

A trust-region quadratic subproblem associated with Problem (3.3) is

$$\begin{aligned} & \text{minimize} && L_k + \nabla L_k^T s + \frac{1}{2} s^T B_k s, \\ & \text{subject to} && g(x_k) + \nabla g(x_k)^T s = 0, \\ & && \|s\| \leq \delta_k \end{aligned}$$

where $\nabla g(x_k) = [\nabla C e(x_k) \nabla C i(x_k) W(x_k) C i(x_k)]$ and B_k is the Hessian of the Lagrangian function $L(x_k, \lambda_k)$ or an approximation to it. This subproblem may be infeasible because there may be no intersecting points between the trust-region constraint $\|s\| \leq \delta_k$ and the hyperplane $g(x_k) + \nabla g(x_k)^T s = 0$. Even if they intersect, there is no guarantee that this will remain true if the trust-region radius δ_k must be decreased. Note that the global convergence of the trust-region methods is based on being able to reduce δ_k until the model trust-region subproblem accurately represents the actual problem. To avoid the difficulty of infeasibility, the reduced Hessian is a successful approach to overcoming the difficulty of having an infeasible trust-region subproblem. The approach was suggested by Byrd [2] and Omojokun [8].

3.1 A Trust-Region Algorithm Outline

This section is devoted to presenting the detailed description of our trust-region algorithm for solving Problem (3.3).

The reduced Hessian approach is used to compute a trial step s_k . In this approach, the trial step s_k is decomposed into two orthogonal components; the normal component s_k^n and the tangential component s_k^t . The trial step s_k has the form $s_k = s_k^n + Z_k \bar{s}_k^t$, where Z_k is a matrix whose columns form an orthonormal basis for the null space of ∇g_k^T .

We obtain the normal component s_k^n by solving the following trust-region sub problem

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\nabla g_k^T s^n + g_k\|^2 \\ & \text{subject to} && \|s^n\| \leq \theta \delta_k, \end{aligned}$$

for some $\theta \in (0, 1)$, where δ_k is the trust-region radius. For solving the above problem we use a conjugate-gradient method.

We summarize the computation of s_k^n in the following algorithm.

Algorithm 3.2 : (The normal component s_k^n)

Step 0: Given $x_k \in \mathfrak{R}^n$, $\delta_k > 0$, and $0 < \epsilon < \epsilon_c < 1$.

Step 1: Set $s_0^n = 0$, $\rho_0 = -\nabla g_k g_k$ and $d_0 = \rho_0$.

Step 2: For $i = 0, 1, \dots$ do

2.1 Compute $\hat{\gamma}_i = \frac{\rho_i^T \rho_i}{d_i^T \nabla g_k \nabla g_k^T d_i}$.

2.2 Compute μ such that $\|s_i^n + \mu d_i\| = \delta_k$.

2.3 If $d_i^T \nabla g_k \nabla g_k^T d_i \leq 0$, then

Set $s_k^n = s_i^n + \mu d_i$ and stop.

Else, set $s_{i+1}^n = s_i^n + \hat{\gamma}_i d_i$.

2.4 If $\|s_{i+1}^n\| > \delta_k$,

Set $s_k^n = s_i^n + \mu d_i$ and stop.

2.5 Update the residual $\rho_{i+1} = \rho_i - \hat{\gamma}_i \nabla g_k \nabla g_k^T d_i$.

2.6 If $\frac{\|\rho_{i+1}\|}{\|\rho_0\|} \leq \epsilon_c$,

Set $s_k^n = s_{i+1}^n$ and stop.

2.7 Compute $\beta_i = \frac{\rho_{i+1}^T \rho_{i+1}}{\rho_i^T \rho_i}$, and the new direction

$d_{i+1} = \rho_{i+1} + \beta_i d_i$.

Given the normal component s_k^n , we compute the tangential component $s_k^t = Z_k \bar{s}_k^t$ by solving the following trust-region subproblem

$$\begin{aligned} & \text{minimize} && [Z_k^T (\nabla_x L_k + B_k s_k^n)]^T \bar{s}^t + \frac{1}{2} \bar{s}^{tT} \bar{B}_k \bar{s}^t \\ & \text{subject to} && \|Z_k \bar{s}^t\| \leq \Delta_k, \end{aligned} \quad (3.6)$$

where $\Delta_k = \sqrt{\delta_k^2 - \|s_k^n\|^2}$ and $\bar{B}_k = Z_k^T B_k Z_k$.

We summarize the computation of \bar{s}_k^t in the following algorithm.

Algorithm 3.3 : (The tangential component \bar{s}_k^t)

Step 0: Given $x_k \in \mathfrak{R}^n$, $\delta_k > 0$, and $0 < \epsilon < \epsilon_c < 1$.

Step 1: Set $\bar{s}_0^t = 0$, $\rho_0 = -Z_k^T (\nabla L_k + B_k s_k^n)$ and $d_0 = \rho_0$.

Step 2: For $i = 0, 1, \dots$ do

2.1 Compute $\hat{\gamma}_i = \frac{\rho_i^T \rho_i}{d_i^T Z_k^T B_k Z_k d_i}$.

2.2 Compute μ such that $\|\bar{s}_i^t + \mu d_i\| = \Delta_k$.

2.3 If $d_i^T Z_k^T B_k Z_k d_i \leq 0$, then

Set $\bar{s}_k^t = \bar{s}_i^t + \mu d_i$ and stop.

Else, set $\bar{s}_{i+1}^t = \bar{s}_i^t + \hat{\gamma}_i d_i$.

2.4 If $\|\bar{s}_{i+1}^t\| > \Delta_k$,

Set $\bar{s}_k^t = \bar{s}_i^t + \mu d_i$ and stop.

2.5 Update the residual $\rho_{i+1} = \rho_i - \hat{\gamma}_i Z_k^T B_k Z_k d_i$.

2.6 If $\frac{\|\rho_{i+1}\|}{\|\rho_0\|} \leq \epsilon_c$,

Set $\bar{s}_k^t = \bar{s}_{i+1}^t$ and stop.

2.7 Compute $\beta_i = \frac{\rho_{i+1}^T \rho_{i+1}}{\rho_i^T \rho_i}$, and the new direction

$$d_{i+1} = \rho_{i+1} + \beta_i d_i.$$

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. We use the augmented Lagrangian function (3.5) as a merit function. To test the step, we compare the actual reduction in the merit function in moving from x_k to $x_k + s_k$ versus the predicted reduction.

The actual reduction in the merit function is defined as

$$Ared_k = L(x_k, \lambda_k) - L(x_{k+1}, \lambda_{k+1}) + r_k [\|g_k\|^2 - \|g_{k+1}\|^2],$$

The predicted reduction in the merit function is defined as

$$\begin{aligned} Pred_k = & -\nabla_x L(x_k, \lambda_k)^T s_k - \frac{1}{2} s_k^T B_k s_k - \Delta \lambda_k^T (g_k + \nabla g_k^T s_k) \\ & + r_k [\|g_k\|^2 - \|g_k + \nabla g_k^T s_k\|^2], \end{aligned} \quad (3.7)$$

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

3.2 Main Trust-Region Algorithm

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

Algorithm 3.4 (The Main Algorithm)

Step 0. (Initialization)

Given $x_0 \in \mathbb{R}^n$. Compute W_0 and evaluate λ_0 . Set $r_0 = 1$ and $\beta > 0$

Choose ε , α_1 , α_2 , η_1 , and η_2 , δ_{min} , δ_{max} , and δ_0 such that $\varepsilon > 0$, $0 < \alpha_1 <$

$1 < \alpha_2$, $0 < \eta_1 < \eta_2 < 1$ and $\delta_{min} \leq \delta_0 \leq \delta_{max}$.

Set $k = 0$.

Step 1. (Test for convergence)

If $\|Z_k^T \nabla_x L_k\| + \|g_k\| \leq \varepsilon$, **then** terminate the algorithm.

Step 2. (Compute a trial step)

- a) Compute the normal component s_k^n by using Algorithm(3.2).
- b) compute \bar{s}_k^t by using Algorithm(3.3).
- c) Set $s_k = s_k^n + Z_k \bar{s}_k^t$ and $x_{k+1} = x_k + s_k$.

Step 3. (Update the active set)

Compute W_{k+1} .

Step 4. (Compute the Lagrange multiplier λ_{k+1})

Compute λ_{k+1} by solving

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

Step 5. (Update the penalty parameter r_k)

If $\text{Pred}_k \geq \frac{r_k}{2} [\|g_k\|^2 - \|g_k + \nabla g_k^T s_k\|^2]$

then set $r_k = r_{k-1}$

else set

$$r_k = \frac{2[q_k(s_k) - q_k(0) + \Delta \lambda_k^T (g_k + \nabla g_k^T s_k)]}{\|g_k\|^2 - \|g_k + \nabla g_k^T s_k\|^2} + \beta, \quad (3.8)$$

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 $\delta_k = \alpha_1 \|s_k\|$.

and compute a new trial step s_k

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

Accept the step: $x_{k+1} = x_k + s_k$.

Set the trust-region radius: $\delta_{k+1} = \max(\delta_k, \delta_{\min})$.

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

Accept the step: $x_{k+1} = x_k + s_k$.

Set the trust-region radius: $\delta_{k+1} = \min\{\delta_{\max}, \max\{\delta_{\min}, \alpha_2 \delta_k\}\}$.

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

4 Global Convergence Theory

In this section, we prove our main global convergence results for our trust-region algorithm for solving Problem (1.1). 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 Ω be a convex subset of \mathfrak{R}^n that contains all iterates x_k and $x_k + s_k$, for all trial steps s_k examined in the course of the algorithm. On the set Ω , the following assumptions are imposed.

Assumptions:

A_1 . The functions $f, Ce(x), e = 1, \dots, m$ and $Ci(x), i = 1, \dots, p$ are twice continuously differentiable for all $x \in \Omega$.

A_2 . The matrix $\nabla g(x) = [\nabla Ce(x) \ \nabla Ci(x)W(x)Ci(x)]$ has full column rank.

A_3 . All of $f(x), \nabla f(x), \nabla^2 f(x), Ce(x), \nabla Ce(x), \nabla^2 Ce_i(x), Ci(x), \nabla Ci(x), \nabla^2 Ci_i(x), \nabla Ci(x)W(x)Ci(x), (\nabla Ce(x)^T \nabla Ce(x))^{-1}$, and $Z(x)$ are uniformly bounded in Ω .

A_4 . The sequence $\{\lambda_k\}$ is bounded.

A_5 . If an approximation to the Hessian of the Lagrangian is used, then the sequence of approximated Hessian matrices $\{B_k\}$ is bounded.

Theorem 4.1 Assume A_1 - A_5 . Then the sequence of iterates generated by the algorithm satisfies

$$\lim_{k \rightarrow \infty} \|g_k\| = 0.$$

Proof. See the proof in El-Sobky[5]

Theorem 4.2 . Assume A_1 - A_5 . The algorithm terminated because for all $\epsilon > 0$

$$\|Z_k^T \nabla_x L_k\| + \|g_k\| \leq \epsilon. \quad (4.1)$$

Proof. See the proof in El-Sobky[5].

5 Numerical Implementation

In this section, we introduce a multi-objective test problem and we convert this problem to a single objective optimization problem by using I-SHOT algorithm(3.2). Then we solve the single objective optimization problem by trust-region algorithm(3.4). Our program was written in MATLAB and run under MATLAB Version 7.0 with machine epsilon 10^{-16} .

In the following subsection, we present the implementation details.

5.1 Implementation Details

Given a starting point x_0 , we choose the initial trust-region radius of our trust-region algorithm to be

$$\delta_0 = \max(\|s_0^{cp}\|, \delta_{min}).$$

The initial trust-region radius satisfy $\delta_{min} \leq \delta_0 \leq \delta_{max}$. So we choose $\delta_{min} = 10^{-3}$ and $\delta_{max} = 10^5 \delta_0$. The values of the constants that are needed to evaluate the step and update the trust-region radius are: $\alpha_1 = 0.05$, $\alpha_2 = 2$, $\eta_1 = 10^{-4}$, $\eta_2 = 0.5$, $\beta = 0.1$ and $\varepsilon = 10^{-8}$.

In the following subsection, we use an optimal control problem as a case study.

5.2 Multi-objective Test Problem

This test problem is an optimal control of continuous Casting by Multi-objective Optimization[6]and [7]

$$\begin{aligned}
 \text{minimize } & f_1(x) = -4.75 - 2.27(x_1 - 0.3), \\
 & f_2(x) = -2 - 0.524(x_1 - 0.3) - 2.79(x_2 - 0.3) - 0.882(y_1 - 0.3) - 2.65(y_2 - 0.3), \\
 & f_3(x) = -5.1 - 0.177(x_1 - 0.3) - 0.978(x_2 - 0.3) - 0.216(y_1 - 0.3) - 0.768(y_2 - 0.3), \\
 & f_4(x) = -7.5 + 0.012(59v_1 - 59), \\
 & f_5(x) = 0.0018(532v_2 - 532), \\
 & f_6(x) = 0.0025(450v_3 - 450), \\
 \\
 \text{subject to } & 2.5 - 0.0332(x_1 - 0.3) - 0.0186(x_2 - 0.3) - 3.34(x_3 - 0.3) \\
 & -0.0204(y_1 - 0.3) - 0.778(y_2 - 0.3) - 2.62(y_3 - 0.3) \leq 0, \\
 \\
 & 0.3 \leq x_i \leq 1, \quad i = 1, 2, 3.
 \end{aligned}$$

(5.1)

where $y_i = \frac{0.39}{1.39-x_i^2}$, $v_i = \frac{1}{1.09-x_i^2}$, and $i = 1, 2, 3$. To find the solution of the above multi-objective constrained optimization problem.

First, compute ε_j by solving the following single objective constrained optimization problems using a trust-region algorithm (3.4)

$$\begin{aligned} & \text{minimize } f_j(x) \quad j = 1, \dots, 6 \\ & \text{subject to } 2.5 - 0.0332(x_1 - 0.3) - 0.0186(x_2 - 0.3) - 3.34(x_3 - 0.3) \\ & \quad - 0.0204(y_1 - 0.3) - 0.778(y_2 - 0.3) - 2.62(y_3 - 0.3) \leq 0, \\ & \quad 0.3 \leq x_i \leq 1, \quad i = 1, 2, 3, \end{aligned}$$

and compute $\hat{\varepsilon}_j$ by solving the following single objective constrained optimization problems using a trust-region algorithm (3.4)

$$\begin{aligned} & \text{maximize } f_j(x) \quad j = 1, \dots, 6 \\ & \text{subject to } 2.5 - 0.0332(x_1 - 0.3) - 0.0186(x_2 - 0.3) - 3.34(x_3 - 0.3) \\ & \quad - 0.0204(y_1 - 0.3) - 0.778(y_2 - 0.3) - 2.62(y_3 - 0.3) \leq 0, \\ & \quad 0.3 \leq x_i \leq 1, \quad i = 1, 2, 3. \end{aligned}$$

The values of ε_j and $\hat{\varepsilon}_j$ are given in the following Table

ε_j	-6.608	-4.4735	-5.9663	-7.5461	-0.05788	11.375
$\hat{\varepsilon}_j$	4.1074	1.4511	4.9089	-579.24	749.48	1257.9

Second, the DM asked to select the targets of $\varepsilon_{j_{\text{bad}}}$ and $\varepsilon_{j_{\text{good}}}$ values. Assume that DM select $\varepsilon_{j_{\text{bad}}}$ and $\varepsilon_{j_{\text{good}}}$ values as the same values of ε_j and $\hat{\varepsilon}_j$.i.e

$\varepsilon_{j_{\text{bad}}}$	-6.608	-4.4735	-5.9663	-7.5461	-0.05788	11.375
$\varepsilon_{j_{\text{good}}}$	4.1074	1.4511	4.9089	-579.24	749.48	1257.9

Third, the DM asked to select the values of U_j , $j = 1, 2, \dots, k$. Assume that DM select U_j values as follows $U_j = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 0, 0)$.

Fourth, We convert problem (5.1) to the following problem

$$\begin{aligned}
 & \text{minimize} && F(x) = \sum_{j=1}^6 u_j \left(\frac{f_j(x) - \varepsilon_{j_{good}}}{\varepsilon_{j_{bad}} - \varepsilon_{j_{good}}} \right) \\
 & \text{subject to} && f_j(x) \leq \varepsilon_j, j = 1, \dots, 6 \\
 & && 2.5 - 0.0332(x_1 - 0.3) - 0.0186(x_2 - 0.3) - 3.34(x_3 - 0.3) \\
 & && - 0.0204(y_1 - 0.3) - 0.778(y_2 - 0.3) - 2.62(y_3 - 0.3) \leq 0, \\
 & && 0.3 \leq x_i \leq 1, \quad i = 1, 2, 3,
 \end{aligned} \tag{5.2}$$

where $\sum_{j=1}^6 u_j = 1$, $u_j \geq 0$, and $\varepsilon_j \in [\varepsilon_{j_{bad}}, \varepsilon_{j_{good}}]$. We assume the values of ε_j are

ε_j	-4.608	-2.4735	-1.9663	-10	1	13.375
-----------------	--------	---------	---------	-----	---	--------

Using algorithm (3.4) to find the solution of problem (5.2) at any initial point $x_0 = (0.3, 0.7, 1)$. The numerical results are summarized in the following Table.

f_j	-4.75	-3.116	-5.4912	-7.5	0.6384	11.375
-------	-------	--------	---------	------	--------	--------

6 Concluding Remarks

In this manuscript, we use the trust region algorithm in solving an interactive approach for multiobjective nonlinear programming problems. The methodology and the convergence analysis of our algorithm is introduced. A numerical example is represented to clarify the proposed algorithm.

References

- [1] S.Azarm and S.Narayanan, "Multiobjective Interactive Sequential Hybrid Optimization Technique for Decision Making", University of Maryland, College Park, Maryland, USA, 1999.
- [2] R.Byrd, "Robust trust-region methods for nonlinearly constrained" optimization, A talk presented at the Second SIAM Conference on Optimization, Houston, TX, 1987 .

- [3] J.Dennis , M.El-Alem, and K.Williamson , ”‘A trust-region approach to nonlinear systems of equalities and inequalities”’, SIAM J Optimization, vol.9, 1999, 291-315.
- [4] B.El-Sobky, ”‘A robust trust-region algorithm for general nonlinear constrained optimization problems”’, PhD thesis, Department of Mathematics, Alexandria University, Alexandria, Egypt, 1998.
- [5] B.El-Sobky, ”‘A Global Convergence Theory for Trust-Region Algorithm for General Nonlinear Programming Problem”’, Publicities in Fourth Saudi Science Conference ”Contribution of Science Faculties in the development process of Kingdom of Saudi Arabia ” sponsored by Taibah University, Al-Madinah Al-Munawarah, Kingdom of Saudi Arabia, March, 21st -24th, 2010.
- [6] K.Miettinen , ”‘Nonlinear Multiobjective Optimization”’, WWW.NIMBUS on the Internet Computer Optimization Research, 2000.
- [7] K.Miettinen, M.Makela, T. Manniko, ”‘Optimal Control of Containouos Casting by Multiobjective Optimization”’, Computational Optimization and Applications, 1998.
- [8] E.Omojokun, ”‘Trust-region strategies for optimization with nonlinear equality and inequality constraints”’, PhD thesis, Department of Computer Science, University of Colorado, Boulder, Colorado, 1989. ditor, Spares Matrices and Thier Uses, pages 57-87. Academic Press, New York.
- [9] R.E. Steuer, ”‘Multiple criteria optimization: Theory, computation and applications”’, New York ,Viley, 1986.

Received: October, 2011