Generation of Pareto optimal solutions for multi-objective optimization problems via a reduced interior-point algorithm

B. El-Sobky\textsuperscript{a} and Y. Abo-Elnaga\textsuperscript{b}

\textsuperscript{a}Department of Mathematics and Computer Science, Faculty of Science, Alexandria University, Alexandria, Egypt; \textsuperscript{b}Department of basic science, Higher Technological Institute, Tenth of Ramadan City, Egypt

\textbf{ABSTRACT}

In this paper, a reduced interior-point (RIP) algorithm is introduced to generate a Pareto optimal front for multi-objective constrained optimization (MOCP) problem. A weighted Tchebychev metric approach is used together with achievement scalarizing function approach to convert MOCP problem to a single-objective constrained optimization (SOCO) problem. An active-set technique is used together with a Coleman–Li scaling matrix and a decrease interior-point method to solve SOCO problem. A Matlab implementation of RIP algorithm was used to solve three cases and application. The results showed that the RIP algorithm is promising when compared with well-known algorithms and the computations may be superior relevant for comprehending real-world application problems.

\section{1. Introduction}

A wide variety of problems in engineering, industry and many other fields involves multi-objective optimization problems (MOPs). We consider in this paper on the following MOCP problem:

\begin{equation}
\begin{aligned}
\text{minimize} & \quad [f_1(x), \ldots, f_q(x)]^T \\
\text{subject to} & \quad H_e(x) = 0, \quad e = 1, \ldots, m, \\
& \quad H_i(x) \leq 0, \quad i = 1, \ldots, p, \\
& \quad \alpha \leq x \leq \beta,
\end{aligned}
\end{equation}

where \(\alpha \in \{\mathbb{R} \cup (-\infty)\}^n\), \(\beta \in \{\mathbb{R} \cup \{\infty\}\}^n\), \(m < n\), and \(\alpha < \beta\). The functions \(f_1, \ldots, f_q, H_e\) and \(H_i\) are assumed to be at least twice continuously differentiable. Let \(\mathbf{F} = \{x : \alpha \leq x \leq \beta\}\) and \(\text{int}(\mathbf{F}) = \{x : \alpha < x < \beta\}\).

In MOCP problem, there is more than one objective function and there is no single optimal solution that at the same time improves all the objective functions. The idea of optimality is supplanted with that of Pareto optimality or effectiveness in MOCP problem. The Pareto optimal (PO) solutions are the solutions that can’t be progressed in one objective function without breaking down their execution in at any rate one of the rest because of the confliction of the objectives. The decision maker (DM) is searching for the most favoured solution among the PO solutions of MOCP problem.

There are a large variety of methods for accomplishing MOCP problem. None of them can be said to be generally superior to all of the others. Wang and Zhou [1] classified these methods to posteriori, no preference, and priori techniques according to the participation of the DM in the solution process.

In this paper, we intrigue with the posteriori techniques, where the DM takes a section in the solution procedure. A posteriori techniques can be called strategies for producing PO solutions. Since, there are infinitely many PO solutions, and then the DM chooses the most favoured one from the PO set.

The weighted Tchebychev metric approach which is introduced by [2] is one of the posteriori methods. It is utilized to convert MOCP problem to the SOCO problem by minimizing the distance between the ideal objective vector and the feasible objective region. If the global ideal objective vector is unknown, we may fail in delivering PO solutions. In the other word, if the reference point utilized as a part of the objective vector inside the feasible objective region, the minimal distance to it is zero and we don’t obtain the PO solution.

So, we will use the achievement scalarizing function approach [2] which is a special type of scalarizing functions to overcome the weakness of the weighted Tchebychev metric approach by replacing metrics by achievement scalarizing function, and the PO solutions
can be generated with any reference point. Using this approach, the MOCP problem (1) transforms to the following SOCO problem:

\[
\begin{align*}
\text{minimize} & \quad \max_{j=1,\ldots,q} \{ w_j(f_j(x) - y_j) \} \\
\text{subject to} & \quad H_k(x) = 0, \quad e = 1, \ldots, m, \\
& \quad H_i(x) \leq 0, \quad i = 1, \ldots, p, \\
& \quad \alpha \leq x \leq \beta,
\end{align*}
\]

where \( \sum_{j=1}^{q} a_j = 1, a_j \geq 0, \) and \( y_j \) is the reference point for all \( j = 1, \ldots, q. \)

Problem (2) can be rewritten as follows:

\[
\begin{align*}
\text{minimize} & \quad F(x) \\
\text{subject to} & \quad H_k(x) = 0, \quad e = 1, \ldots, m, \\
& \quad H_i(x) \leq 0, \quad i = 1, \ldots, p, \\
& \quad \alpha \leq x \leq \beta,
\end{align*}
\]

where \( F(x) = \max_{j=1,\ldots,q} \{ w_j(f_j(x) - y_j) \} \). To convert the above problem to an equality constrained optimization (EQCO) problem with bound on the variables, we use an active-set strategy which is introduced in [3]. Many authors have used active-set algorithms to solve the general SOCO problems. For example, see [3–6].

Using the active-set mechanism in [3], we have a diagonal matrix \( W(x_k) \), whose diagonal elements are defined as follows:

\[
w_i(x) = \begin{cases} 1, & \text{if } H_i(x_k) \geq 0, \\ 0, & \text{if } H_i(x_k) < 0. \end{cases}
\]

The above matrix utilizes to transform Problem (3) to the following bounded EQCO problem:

\[
\begin{align*}
\text{minimize} & \quad F(x) + \frac{r}{2} \| W(x) H_k(x) \|^2 \\
\text{subject to} & \quad H_k(x) = 0, \\
& \quad \alpha \leq x \leq \beta,
\end{align*}
\]

where \( r \) is positive parameter. The augmented Lagrangian function associated with Problem (5) without the bounded constraint \( \alpha \leq x \leq \beta \) is defined as follows:

\[
\ell(x, \lambda; r) = F(x) + \lambda^T H_k(x) + \frac{r}{2} \| W(x) H_k(x) \|^2,
\]

where \( \lambda \) represents the Lagrange multiplier vector associated with the equality constraint \( H_k(x) \).

The augmented Lagrangian function associated with Problem (5) is given by

\[
L(x, \lambda, \mu, \nu; r) = \ell(x, \lambda; r) - \mu^T (x - \alpha) - \nu^T (\beta - x),
\]

where \( \mu \) and \( \nu \) are Lagrange multiplier vectors associated with inequality constraints \( (x - \alpha) \) and \( (\beta - x) \) respectively.

The first-order necessary conditions for a point \( x_0 \) to be a solution of Problem (5) are the existence of multiplier vectors \( \lambda_0, \mu_0 \) and \( \nu_0 \) such that the point \( (x_0, \lambda_0, \mu_0, \nu_0) \) satisfies

\[
\nabla_x \ell(x_0, \lambda_0; r_0) - \mu_0 + \nu_0 = 0,
\]

where \( \mu_0 \) and \( \nu_0 \) are defined as follows:

\[
\begin{align*}
\alpha \leq x_0 & \leq \beta, \\
\mu_0^{(b)}(x_0^{(b)} - \alpha^{(b)}) & = 0, \\
\nu_0^{(b)}(\beta^{(b)} - x_0^{(b)}) & = 0,
\end{align*}
\]

for all \( b \) corresponding to \( x^{(b)} \) with finite bound and

\[
\nabla_x \ell(x_0, \lambda_0; r_0) = \nabla F(x_0) + \nabla H_k(x_0) \lambda_0 + r_0 \nabla H_k(x_0) W(x_0) H_k(x_0).
\]

Using the scaling diagonal matrix \( A(x) \) whose diagonal elements are defined as follows:

\[
a^{(b)}(x) = \begin{cases} \sqrt{(\beta^{(b)} - x^{(b)}),} & \text{if } \nabla_x \ell(x, \lambda; r) \geq 0 \text{ and } \alpha^{(b)} > -\infty, \\ \sqrt{(\beta^{(b)} - x^{(b)}),} & \text{if } \nabla_x \ell(x, \lambda; r) < 0 \text{ and } \beta^{(b)} < +\infty, \\ 1, & \text{otherwise,} \end{cases}
\]

the system (7–11) converted to the following system:

\[
A^2(x) \nabla_x \ell(x, \lambda; r) = 0,
\]

where \( A(x) \) is the above system is not everywhere differentiable but it is continuous. More details about the derivation of the above system are given in [8].

In the rest of the paper, we use the symbol \( F_k \equiv F(x_k), \quad H_k \equiv H_k(x_k), \quad A_k \equiv A_k(x_k), \quad \nabla_k \ell \equiv \nabla \ell(x_k, \lambda_k; r_k), \quad \nabla_k \lambda \equiv \nabla \lambda(x_k, \lambda_k; r_k), \quad \nabla_k \mu \equiv \nabla \mu(x_k, \lambda_k; r_k), \quad \nabla_k \nu \equiv \nabla \nu(x_k, \lambda_k; r_k), \quad Z_k \equiv Z(x_k) \) and so on to denote the function value at a particular point.

Applying Newton’s method on the above nonlinear system, we have the following \( (n + m) \times (n + m) \) system:

\[
\begin{bmatrix}
A_k^2 \nabla^2 \ell_k + \text{diag} (\nabla \ell_k) \text{diag} (\eta_k) A_k^2 \nabla \lambda_k \\
\nabla \nabla^T \ell_k H_k \\
0
\end{bmatrix}
= \begin{bmatrix}
\Delta \lambda_k \\
\Delta \ell_k \\
\Delta H_k
\end{bmatrix},
\]

where \( \text{diag} (\eta_k) \) is a diagonal matrix whose diagonal elements are defined as follows:

\[
\eta^{(b)}(x_k) = \begin{cases} 1, & \text{if } \nabla_x \ell(x_k, \lambda_k; r_k) \geq 0 \text{ and } \alpha^{(b)} > -\infty, \\ -1, & \text{if } \nabla_x \ell(x_k, \lambda_k; r_k) < 0 \text{ and } \beta^{(b)} < +\infty, \\ 0, & \text{otherwise.} \end{cases}
\]
For more details about diag ($\eta(x_k)$), see [5,9]. This system is called an extended system.

One of the disadvantages of obtaining Newton’s step ($\Delta x_k, \Delta \lambda_k$) by solving the extended system lies in the fact that the dimension of the extended system is large for large-scale problems. In this paper, we present a method that computes Newton’s step ($\Delta x_k, \Delta \lambda_k$) by solving a smaller dimension linear system.

This paper is arranged as follows. In Section 2, a detailed description of RIP algorithm is offered. Section 3 contains a Matlab implementation of the main algorithm. The main algorithm is applied to three cases study and problem was chosen from the engineering application (Two-Bar Truss), and the results are reported. Finally, Section 4 contains concluding remarks.

In the following section, we offered a detailed description of the main steps to RIP algorithm to solve the system (15).

2. A reduced interior-point algorithm

In this section, we introduce a detailed description of RIP algorithm to compute Newton’s step ($\Delta x_k, \Delta \lambda_k$) by solving a smaller dimension system.

Let $Z(x_k) \in \mathbb{R}^{m \times (n-m)}$ be a matrix belongs to the null space of $(A_k^2 \nabla H_{e_k})^T$. That is

$$Z_k^T A_k^2 \nabla H_{e_k} = 0.$$  \hfill (17)

Applying QR factorization on $A_k^2 \nabla H_{e_k}$, we have

$$A_k^2 \nabla H_{e_k} = \begin{bmatrix} Y(x_k) & Z(x_k) \end{bmatrix} \begin{bmatrix} R(x_k) \\ 0 \end{bmatrix},$$  \hfill (18)

where $R(x_k)$ is an $(m \times m)$ upper triangular matrix and $Y(x_k)$ is an $n \times m$ matrix whose orthonormal columns form a basis for the column space of $A_k^2 \nabla H_{e_k}(x_k)$. Notice that, $Y(x_k)^T Y(x_k) = I_m, Z(x_k)^T Z(x_k) = I_{n-m}$ and $Y(x_k)^T Y(x_k) + Z(x_k)^T Z(x_k) = I_n$. The matrix $R(x_k)$ is nonsingular, if $x_k$ lies in a neighbourhood of $x_*$, and the matrix $A_k^2(x_k) \nabla H_{e_k}(x_k)$ is a nonsingular at $x_*$.

Using a continuous differentiable matrix $Z(x_k)$, the system of equations (13)–(14) are equivalent to the following system:

$$Z_k^T A_k^2 \nabla \tilde{e}(x_k; r_k) = 0,$$

$$H_{e_k} = 0,$$

where $\nabla \tilde{e}(x_k; r_k) = \nabla F(x_k) + r_k \nabla H_{e_k}(x_k)W(x_k)H_{e_k}(x_k)$. This system consists of $n$ nonlinear equations. Applying Newton’s method on the above system, we have

$$\begin{bmatrix} Z_k^T A_k^2 \nabla \tilde{e}(x_k; r_k) \rangle \\ H_{e_k} \end{bmatrix} \Delta x = \begin{bmatrix} -Z_k^T A_k^2 \nabla \tilde{e}(x_k; r_k) \rangle \\ H_{e_k} \end{bmatrix}.$$

To compute the Jacobian of the above system, we have

$$Z_k^T A_k^2 \nabla \tilde{e}(x_k; r_k) \rangle = Z_k^T A_k^2 \nabla^2 \tilde{e}(x_k) + \text{diag} (\nabla \tilde{e}(x_k)) \text{diag} (\eta_k),$$  \hfill (20)

where $\nabla^2 \tilde{e}(x_k) = \nabla^2 F(x_k) + r_k \nabla H_{e_k}(x_k)W(x_k) \nabla H_{e_k}(x_k)^T$ and we need to compute $Z_k^T A_k^2 \nabla H_{e_k}$.

To compute $Z(x_k)'$, we differentiate Equation (17) to obtain

$$Z_k^T A_k^2 \nabla H_{e_k} = \begin{bmatrix} \nabla^2 \tilde{e}(x_k) \end{bmatrix} \begin{bmatrix} \nabla \tilde{e}(x_k) \rangle \\ H_{e_k} \end{bmatrix}.$$

From (18) and (21), we have

$$[Z_k^T A_k^2 \nabla H_{e_k}]' = -[Z_k^T A_k^2 \nabla H_{e_k}]' R(x_k)^{-1}.$$

Since $[Z_k^T A_k^2 \nabla H_{e_k}]' = [Z_k^T A_k^2 \nabla H_{e_k}]' = [Z_k^T A_k^2 \nabla H_{e_k}]' = 0$, then

$$[Z_k^T A_k^2 \nabla H_{e_k}]' = -[Z_k^T A_k^2 \nabla H_{e_k}]' R(x_k)^{-1} Y_k^T.$$

Hence

$$[Z_k^T A_k^2 \nabla H_{e_k}]' = -[Z_k^T A_k^2 \nabla H_{e_k}]' R(x_k)^{-1} Y_k^T.$$

If we take

$$\lambda_k = -R(x_k)^{-1} Y_k^T A_k^2 \nabla \tilde{e}(x_k),$$

then

$$[Z_k^T A_k^2 \nabla H_{e_k}]' = [Z_k^T A_k^2 \nabla H_{e_k}]' \lambda_k.$$

From (19), (20) and (22), Newton’s step is computed by solving the following system:

$$\begin{bmatrix} Z_k^T A_k^2 \nabla \tilde{e}(x_k; \lambda_k) \\ H_{e_k} \end{bmatrix} \Delta x = \begin{bmatrix} -Z_k^T A_k^2 \nabla \tilde{e}(x_k; \lambda_k) \\ H_{e_k} \end{bmatrix}.$$  \hfill (23)

But we need to estimate the positive parameter $r_{k+1}$. To estimate it, the following subproblem is used

minimize $|\nabla F_{k+1} + r_k \nabla H_{e_{k+1}}W_{k+1}H_{e_{k+1}}|^2$, \hfill (24)

see [10]. The Lagrange multiplier $\lambda$ should be given by

$$R(x_k) \lambda = -r_k \nabla H_{e_k}$$

Once the step $\Delta x$ is evaluated, the damping parameter $r_k$ is computed to ensure that $x_{k+1} \in \text{int}(F)$. To ensure that $x_{k+1} \in \text{int}(F)$, we need to compute a damping parameters $r_k$ and $\lambda_k$. The steps for computing $r_k$ and $\lambda_k$ are presented in the following algorithm.

Algorithm 2.1 (Computing the damping parameters $r_k$ and $\lambda_k$): Compute the damping parameter $r_k$ and $\lambda_k$.
as follows: For \( i = 1:n \), do

\[
u_k^{(i)} = \begin{cases} \frac{\alpha^{(i)} - x_k^{(i)}}{\Delta x_k^{(i)}}, & \text{if } \alpha^{(i)} > -\infty \text{ and } \Delta x_k^{(i)} < 0, \\ 1, & \text{otherwise}, \end{cases}
\]

and

\[
u_k^{(i)} = \begin{cases} \frac{\beta^{(i)} - x_k^{(i)}}{\Delta x_k^{(i)}}, & \text{if } \beta^{(i)} < \infty \text{ and } \Delta x_k^{(i)} > 0, \\ 1, & \text{otherwise}. \end{cases}
\]

Set \( \phi(i) = \min\{u_k^{(i)}, v_k^{(i)}\} \).

End.

Set \( t_k = \min\{1, \min \phi\} \).

Set \( x_{k+1} = x_k + t_k \Delta x_k \).

For \( i = 1:n \), do

If \( x_{k+1}(i) \in \text{int}(F(i)) \)

Set \( x_{k+1}(i) = x_k(i) + t_k \Delta x_k(i) \).

Else, compute \( \theta_k \in [1 - \sigma \|\Delta x_k\|, 1] \) and \( \sigma > 0 \) is a pre-specified fixed constant.

Set \( x_{k+1}(i) = x_k(i) + \theta_k t_k \Delta x_k(i) \).

End if.

End for.

Master steps of RIP algorithm to solve SOCO problem are offered in the following algorithm.

**Algorithm 2.2 (RIP algorithm):** A formal description of RIP algorithm is offered in the following algorithm.

**Step 0.** Given \( x_0 \in \mathcal{F} \). Evaluate \( r_0, \lambda_0, W_0, A_0 \) and \( \eta_0 \). Choose \( \epsilon = 10^{(-10)} \). Set \( k = 0 \).

**Step 1.** If \( \|z^T A_k \nabla \tilde{e}_k\| + \|H_k\| \leq \epsilon \), then stopping the algorithm.

**Step 2.** Compute Newton’s step \( \Delta x_k \) by solving the system (23).

**Step 3.** Using Algorithm (2.1) to compute the damping parameters \( t_k \) and \( \theta_k \).

**Step 4.** Compute \( r_{k+1} \) by solving (24) and \( \lambda_{k+1} \) by solving (25).

**Step 5.** Compute \( W_{k+1} \) by using (4).

**Step 6.** Compute \( A_{k+1} \) and \( \eta_{k+1} \) by using (12) and (16) respectively.

**Step 7.** Set \( k = k+1 \) and return to Step 1.

The major steps of our strategy to generate Pareto optimal front points for MOCP problem are expounded in detail as follows.

**Algorithm 2.3 (The major algorithm):**

**Step 1.** Ask the decision maker to specify the reference point \( \hat{y}_j \) for all \( j = 1, \ldots, q \).

**Step 2.** Using the achievement scalarizing function approach to convert Problem (1) to SOCO Problem (3).

**Step 3.** Algorithm (2.2) is used to solve Problem (2) for a certain values of \( \omega_j \) where \( \omega_j \geq 0 \), \( \sum_{j=1}^q \omega_j = 1 \).

**Step 4.** Formulate Problem (3) where \( F(x_k) = \max_{j=1,...,q} \{\omega_j f_j(x_k) - \hat{y}_j\} \).

**Step 5.** Compute the PO solutions for the objective functions by using Algorithm (2.2) to solve Problem (2).

**Step 6.** Repeat again Steps 4–5 for different values of \( \omega_j \) where \( \omega_j \geq 0 \), \( \sum_{j=1}^q \omega_j = 1 \) to generate the Pareto-optimal front for the objective functions.

### 3. Implementations

To clarify the effectiveness of Algorithm (2.2), we introduce an extensive variety of possible numeric constrained test functions. In this section, a comparison study for many benchmark test functions with distinct Pareto-optimal front was used in [11,12, and 14]. The degree of difficulty of these problems varies from simple to difficult. Graphical presentation of the experimental results and associated observations is presented in this section. The first benchmark test problem was used in [11], the second problem was used in [12], which is a heavily constrained, six decision variable problems. Tanaka [13] is the third MOP selected as benchmark test problem such that its constraints make the Pareto-optimal set discontinuous. All these problems are constrained multi-objective problems and we solve it by using RIP algorithm and results illustrate perform equally well as the results that introduced by different authors using well-known algorithms [14].

The numerical results of Algorithm (2.2) have been performed on a laptop with Intel Core(TM) i7 – 2670QM CPU 2.2 GHz and 8 GB RAM. Algorithm (2.2) was implemented as a MATLAB code and run under MATLAB version 7.10.0.499 (R2013a).

### 3.1. Test problems

The problem in Binh2 is genuinely basic in that way, the constraints may not present extra trouble in finding the PO solutions as appeared in Figure (4.2). It was watched that all multi-objective optimization algorithms performed similarly well and gave a dense sampling of solutions along the true Pareto-ideal bend. The Osyczka Figure 2 and the Tanaka Figure 1 are generally troublesome. The PO set is discontinuous in the Tanaka

Figure 1. Results for Binh problem.
problem due to the constraints. The constraints in the TNK problem partition the Pareto-ideal set into five. We compare our strategy and a dependable and efficient multi-objective algorithm. The results demonstrate that our strategy can be utilized productively for constrained MOP problem. For the Osyczka problem, it can be seen that our strategy gave a good sampling of points at the midsection of the curve and a good sampling of points at the extreme ends of Pareto curve.

In the following section, our strategy applicable to the problem was looked over the engineering application (Two-Bar Truss) [15,16].

3.2. Application: (Two-Bar Truss)

In this section, RIP algorithm is applicable to the Two-Bar Truss problem that studied in [15,16] and illustrated in Figure 3. It is included two stationary stuck joints, A and B, where everyone is associated with one of the two bars in the truss. The two bars are stuck where the go along with each other at joint C, and a 100 kN force acts specifically descending by then. The cross-sectional areas of the two bars are symbolized by $x_1$ and $x_2$, the cross-sectional areas of trusses AC and BC individually. Finally, $y$ represents the perpendicular distance from the line AB that contains the two-pinned base joints to the connection of the bars where the force acts (joint C). The problem has been adjusted into a two-objective problem so as to demonstrate the non-inferior Pareto set obviously in two dimensions. The stresses in AC and BC should not override 100,000 kPa and the total volume of material should not override 0.1. The reason the constraints have been forced is that the Pareto-set is asymptotic and stretches out from $-\infty$ to $\infty$. As $x_1 \to 0$ and $x_2 \to 0$, $f_{\text{volume}} \to 0$ and, $f_{\text{stressAC}}$ and $f_{\text{stressBC}}$ go to $\infty$. As $x_1 \to \infty$ and $x_2 \to \infty$, $f_{\text{volume}}$ goes to $\infty$ and $f_{\text{stressAC}}$, and $f_{\text{stressBC}}$ go to zero. Hence, to generate PO solutions in a credible range, constraints are imposed. The mathematical formulation of the Two-Bar Truss application is shown below.

$$\begin{align*}
\text{minimize} & \quad f_{\text{volume}} = x_1 \sqrt{16 + y^2} + x_2 \sqrt{1 + y^2} \\
\text{minimize} & \quad f_{\text{stressAC}} = \frac{20 \sqrt{16 + y^2}}{x_1y}, \\
\text{subject to} & \quad f_{\text{volume}} \leq 0.1, \\
& \quad f_{\text{stressAC}} \leq 100000, \\
& \quad f_{\text{stressBC}} = \frac{80 \sqrt{1 + y^2}}{x_2y} \leq 100000, \\
& \quad 0 \leq x_1, \\
& \quad x_2 \\
& \quad 1 \leq y \leq 3.
\end{align*}$$

We show the PO solutions of the Two-Bar Truss problem in Figure 4. From the results, we can say that RIP algorithm is able to get a uniform set of non-dominated solution points along the true PO front. Obviously RIP algorithm beats the algorithm in [15,16].

On comparing the RIP algorithm with a reliable and efficient multi-objective genetic algorithm NSGA II introduced in [17], it is clear that RIP algorithm is capable to maintain an almost uniform set of non-dominated solution points along the true Pareto-optimal front and could find a good distribution of solutions near
the Pareto optimal front as that introduced in NSGA II. Finally, the results provided by RIP algorithm for benchmark problems and engineering applications are promising when compared with exiting well-known algorithms.

4. Conclusion

On treating the multi-objective programming problems, we hope to find good solutions near the PO front in small computational time. We introduce a reduced interior-point method to solve MOP problem, where the active-set technique and Coleman–Li scaling matrix are used together with Newton’s method to solve the MOP problem after transforming it to SOCO problem and generate approximate true PO front. Algorithm RIP retains path of all the feasible solutions found during the optimization. It is clear from the results that RIP algorithm for different test problems and engineering application is one of the promising approaches for generating a good approximation of solutions near the PO front in tiny computational times. Also, our results propose that RIP algorithm could be applied to solve application problems of the real world. We hope to apply RIP algorithm for more complex real-world application in the future.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] Wang S, Zhou X. Multiobjective optimization in spillway profile. Approx Optim Comput Theory Appl. 1990;309–312.
[2] Miettinen K, Makela M. Interactive bundle-based method for non-differentiable multi-objective optimization. NIMBUS, Optim. 1995;34(3):231–246.
[3] Dennis J, El-Alem M, Williamson K. A trust-region approach to nonlinear systems of equalities and inequalities. SIAM J Optim. 1999;9:291–315.
[4] El-Sobky B. A multiplier active-set trust-region algorithm for solving constrained optimization problem. Appl Math Comput. 2012;219:127–157.
[5] El-Sobky B. An interior-point penalty active-set trust-region algorithm. J Egyptian Math Soc. 2016;24:672–680.
[6] El-Sobky B. An active-set interior-point trust-region algorithm. Pac J Optim. 2018;14(1):125–159.
[7] Coleman T, Li Y. An interior trust region approach for nonlinear minimization subject to bounds. SIAM J Optim. 1996;6:418–445.
[8] El-Alem M, El-Sayed M, El-Sobky B. Local convergence of interior-point Newton method for constrained optimization. J Optim Theory Appl. 2004;120(3):487–502.
[9] El-Sobky B, Abouel-Naga Y. Multi-objective optimal load flow problem with interior-point trust-region strategy. Electric Power Syst Res. 2017;148:127–135.
[10] Bazaraa M, Sherali H, Shetty C. Nonlinear programming theory and algorithms. Hoboken: John Wiley & Sons; 2006.
[11] Binh T, Korn U. A parallel multiobjective evolutionary algorithm, Technical report. Barleben, Germany: Institute for Automation and Communication; 1999.
[12] Oszczka A, Kundai S. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. Struct Optim. 1995;10:94–99.
[13] Tanaka M. GA-based decision support system for multicriteria optimization. Proc Int Conf Syst, Man Cybernatics. 1995;2:1556–1561.
[14] Mousa A, El-Shorbagy M, Abd El-Wahed W. Local search based hybrid particle swarm optimization for multiobjective optimization. Int J Swarm Evol Comput. 2012;3(1):14–29.
[15] Kirsch U. Optimal structural design. New York: McGraw-Hill; 1981.
[16] Sarker R, Abbass H, Karim S. An evolutionary algorithm for constrained multiobjective optimization problems. 5th Australasia–Japan Joint Workshop; 2001. Dunedin: University of Otago. p. 1–10.
[17] Al-Thobaiti M, Kotb A, Mousa A. Integrating optimization technique coupling an evolutionary algorithm and local search scheme. J Global Res Math Arch. 2014;2(1):11–21.

Appendix

Test problem 1: (MOP-C1 Binh2)

\[
\text{minimize } f_1(x, y) = 4x^2 + 4y^2 \\
\text{minimize } f_2(x, y) = (5 - x)^2 + (5 - y)^2, \\
\text{subject to } (5 - x)^2 + y^2 - 25 = 0, \\
\quad 7.7 - (8 - x)^2 - (3 + y)^2 \leq 0, \\
\quad 0 \leq x \leq 5, \\
\quad 0 \leq y \leq 3.
\]

Test problem 2: (MOP-C2 Oszczka 2)

\[
\text{minimize } f_1 = -[25(2 - x_1)^2 + (2 - x_2)^2 + (1 - x_3)^2 + (4 - x_4)^2 + (1 - x_5)^2] \\
\text{minimize } f_2 = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2, \\
\text{subject to } x_1 + x_2 = 2 = 0, \\
\quad x_1 + x_2 - 2 \leq 0, \\
\quad x_1 - 3x_2 - 2 \leq 0, \\
\quad (3 - x_3)^2 + x_4 - 4 \leq 0, \\
\quad -((3 - x_3)^2 + x_5) + 4 \leq 0, \\
\quad 0 \leq x_1, x_2, x_3, x_4 \leq 10, \\
\quad 1 \leq x_3, x_5 \leq 5, \\
\quad 0 \leq x_4 \leq 6.
\]

Test problem 3: (Tanaka)

\[
\text{minimize } f_1(x, y) = x \\
\text{minimize } f_2(x, y) = y, \\
\text{subject to } -x^2 + y^2 + 1 + 0.1 \cos(16 \tan^{-1}(\frac{x}{y})) \leq 0, \\
\quad (0.5 - x)^2 + (0.5 - y)^2 - 0.5 \leq 0, \\
\quad 0 \leq x \leq \pi, \\
\quad 0 \leq y \leq \pi.
\]