AIMS Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(2): 2509–2540.
DOI: 10.3934/math.2025117
Received: 16 October 2024
Revised: 23 January 2025
Accepted: 29 January 2025
Published: 12 February 2025

Research article

A nonmonotone trust region technique with active-set and interior-point methods to solve nonlinearly constrained optimization problems

Bothina El-Sobky[1,*], **Yousria Abo-Elnaga**[2] **and Gehan Ashry**[1]

[1] Department of Mathematics and Computer Science, Faculty of Science, Alexandria University, Alexandria, Egypt
[2] Department of basic science,Tenth of Ramadan City, Higher Technological Institute, Egypt

* **Correspondence:** Email: bothina_elsobky@alexu.edu.eg.

Abstract: This study is devoted to incorporating a nonmonotone strategy with an automatically adjusted trust-region radius to propose a more efficient hybrid of trust-region approaches for constrained optimization problems. First, the active-set strategy was used with a penalty and Newton's interior point method to convert a nonlinearly constrained optimization problem to an equivalent nonlinear unconstrained optimization problem. Second, a nonmonotone trust region was utilized to guarantee convergence from any starting point to the stationary point. Third, a global convergence theory for the proposed algorithm was presented under some assumptions. Finally, the proposed algorithm was tested by well-known test problems (the CUTE collection); three engineering design problems were resolved, and the results were compared with those of other respected optimizers. Based on the results, the suggested approach generally provides better approximation solutions and requires fewer iterations than the other algorithms under consideration. The performance of the proposed algorithm was also investigated, and computational results clarified that the suggested algorithm was competitive and better than other optimization algorithms.

Keywords: active set; penalty method; interior point; nonmonotone trust region; global convergence
Mathematics Subject Classification: 49M37, 65K05, 90C30, 90C55

---

1. Introduction

In this paper, we will consider the following nonlinear constrained optimization problem

$$
\begin{aligned}
\textit{minimize} \quad & f(x), \\
\textit{subject to} \quad & P_i(x) = 0, \quad i \in \tilde{E}, \\
& P_i(x) \leq 0, \quad i \in \tilde{I}, \\
& \hat{u} \leq x \leq \hat{v},
\end{aligned}
\tag{1.1}
$$

where f: $\Re^n \to \Re$ and $P_i(x)$: $\Re^n \to \Re^m$, such that

$$\tilde{E} \bigcup \tilde{I} = \{1, \cdots, m\}$$

and

$$\tilde{E} \bigcap \tilde{I} = \emptyset$$

are twice continuously differentiable and $m < n$. We denote the feasible set

$$\boldsymbol{E} = \{x : \hat{u} \leq x \leq \hat{v}\}$$

and the strict interior feasible set

$$int(\boldsymbol{E}) = \{x : \hat{u} < x < \hat{v}\},$$

where

$$\hat{u} \in \{\Re \bigcup \{-\infty\}\}^n, \quad \hat{v} \in \{\Re \bigcup \{\infty\}\}^n,$$

and $\hat{u} < \hat{v}$.

This work combines an active-set strategy with the penalty approach to transform problem (1.1) into an unconstrained optimization problem with bounded variables. To solve the unconstrained optimization problem with bounded-on variables, Newton's interior-point method, which was suggested in [1] and used by [2–5], is utilized. However, Newton's method is a local one, and it may not converge if the starting point is far from a stationary point. A nonmonotone trust-region mechanism deals with this problem and guarantees convergence from any starting point to the stationary point.

A trust-region technique can induce strong global convergence and is a very important method for solving unconstrained and constrained optimization problems [6–9]. The trust-region technique is more robust when dealing with rounding errors. One advantage of this technique is that it does not require the model's objective function to be convex.

A critical aspect in trust-region approaches is the strategy for determining the trust-region radius Δ_k at each iteration. The standard trust-region strategy is predicated on the objective function and the model agreement. The radius of the trust region is updated by paying attention to the ratio

$$r_k = \frac{Ared_k}{Pred_k},$$

where $Ared_k$ represents the actual reduction, and $Pred_k$ represents the predicted reduction. It is safe to increase Δ_k in the next iterate when r_k is close to 1, and this is due to a good agreement between the objective function and the model over a current region of trust. Otherwise, Δ_k must be reduced.

It is well-known that the standard trust-region radius Δ_k is independent of the gradient and Hessian of the objective function, so we cannot know if the radius Δ_k is convenient for the whole implementation. This situation increases the number of subproblems to solve in the inner steps of the method, decreasing its efficiency. If we reduce the number of ineffective iterations, we can decrease the number of subproblems solved in each step. Authors in [10] proposed a method for determining the initial radius monitoring agreement between the objective function and the model along the steepest descent path evaluated.

Authors in [11] proposed the first customizable technique to reduce the number of solved subproblems. This technique used the gradient and Hessian information from the current iteration to calculate the trust-region radius Δ_k without requiring an initial trust-region radius.

Motivated by the idea proposed in [11], authors in [12] proposed an automatically adjustable radius for trust-region methods and demonstrated that nonmonotone trust-region methods inherit the conventional trust-region method's strong convergence features. On the other hand, authors in [13] presented a nonmonotone strategy to line search methods for unconstrained optimization problems. Numerical experiments and theoretical analysis have referred to the effectiveness of this method in improving both the possibility of obtaining the global optimum and the rate of convergence of algorithms [14]. Motivated by these outstanding results, many researchers have been interested in combining the nonmonotone strategy with the trust-region methods [15, 16].

Nonmonotone approaches have altered the ratio r_k when comparing $Ared_k$ to $Pred_k$. The following is a definition of one of the most common nonmonotone ratios:

$$\tilde{r}_k = \frac{f_{l(k)} - f(x_k + d_k)}{Pred_k},$$

where

$$f_{l(k)} = \max_{0 \le j \le m(k)} \{f_{k-j}\}$$

in which

$$m(0) = 0$$

and

$$m(k) = \min\{m(k-1) + 1, N\}$$

with an integer constant $N \ge 0$. It has been proven that the nonmonotone trust-region methods inherit the robust convergence properties of the trust-region method. The numerical experiments of the nonmonotone trust-region methods have shown that it is more efficient than the monotone versions, especially in the presence of the narrow, curved valley. Although the nonmonotone strategy in [13] performs well in many cases, it contains some drawbacks, and two important instances of these drawbacks can be described as follows:

- In any iterate, a good function value generated is essentially discarded due to $\{max\}$ term in

$$f_{l(k)} = \max_{0 \le j \le m(k)} \{f_{k-j}\}.$$

- The numerical performances in some cases seriously depend on the choice of parameter N.

Authors in [17] proposed a new nonmonotone strategy for line search methods. It was based on a weighted average of previous successive iterations. This strategy is generally efficient and promising when encountered with unconstrained optimization and can overcome the mentioned drawbacks. In this strategy, $f_l(k)$ is replaced with a weighted average of previous successive iterations C_k, which is defined as follows:

$$C_k = \begin{cases} f(x_k), & \text{if } k = 0, \\ \frac{\eta_{k-1} Q_{k-1} C_{k-1} + f(x_k)}{Q_k}, & \text{if } k \ge 1, \end{cases}$$

and

$$Q_k = \begin{cases} 1, & \text{if } k = 0, \\ \eta_{k-1} Q_{k-1} + 1, & \text{if } k \geq 1, \end{cases}$$

such that

$$0 \leq \eta_{\min} \leq \eta_{k-1} \leq \eta_{\max} \leq 1,$$

where η_k is updated as follows

$$\eta_k = \begin{cases} 0.5\eta_0, & \text{if } k = 1, \\ 0.5(\eta_{k-1} + \eta_{k-2}), & \text{if } k \geq 2. \end{cases} \tag{1.2}$$

Motivated by the advantage of the nonmonotone strategy in the trust-region framework [17], authors in [18] suggested a new nonmonotone trust-region method such that the ratio $\tilde{r}_k$ in their proposal changed as follows

$$\tilde{r}_k = \frac{C_k - f(x_k + d_k)}{Pred_k}.$$

The investigation has proved that the combination of the nonmonotone strategy of [17] with the trust region a new method that has inherited the strong theoretical properties of the trust-region method as well as the computational robustness of the nonmonotone strategy.

Motivated by the nonmonotone trust-region strategy in [18], we will utilize it in our proposed method. We expect it will significantly decrease the total number of iterations and function evaluations. We will clarify that under some conditions, the proposed nonmonotone trust-region active-set penalty (NTRAI) algorithm has global convergence properties.

Furthermore, the applicability of the NTRAI approach to solving problem (1.1) was examined using well-known test problems (the CUTE collection), three mechanical engineering problems from the most recent test suite [19], and a nonconvex problem from [20]. Numerical experiments show that the NTRAI method exceeds rival algorithms in terms of efficacy.

Some notations are utilized throughout this paper, and this is clarified in the rest of this section. The paper is organized as follows: In Section 2, a detailed description of the main steps of the NTRAI algorithm for constrained optimization problems is introduced. Section 3 is devoted to the global convergence theory of the NTRAI algorithm under some suitable conditions. Section 4 contains a Matlab implementation of the NTRAI algorithm and numerical results for three mechanical engineering problems. Finally, Section 5 contains concluding remarks.

To express the function value at a particular point, we use the symbol

$$f_k = f(x_k),\ \nabla f_k = \nabla f(x_k),\ \nabla^2 f_k = \nabla^2 f(x_k),\ P_k = P(x_k),\ \nabla P_k = \nabla P(x_k),\ Y_k = Y(x_k),\ Z_k = Z(x_k)$$

and so on. We denote the Hessian of the objective function f_k or an approximation to it by P_k . Finally, every norm is l_2-norms.

2. Nonmonotone trust-region active-set penalty algorithm

In this section, we will first present a complete description of the significant steps of the active-set strategy using the penalty technique and Newton's interior-point approach. Then the nonmonotone trust-region algorithm's essential phases are presented. Finally, the key stages for applying the NTRAI method to problem (1.1) are shown.

2.1. An active-set penalty interior-point method

Consider the active-set strategy introduced in [21] and used by [5, 7]. We will introduce a diagonal matrix $Z(x) \in \Re^{m\times m}$ whose diagonal entries are

$$z_i(x) = \begin{cases} 1, & \text{if } i \in \tilde{E}, \\ 1, & \text{if } i \in \tilde{I} \text{ and } P_i(x) \geq 0, \\ 0, & \text{if } i \in \tilde{I} \text{ and } P_i(x) < 0, \end{cases} \tag{2.1}$$

where $i = 1, \cdots, m$. Utilizing the diagonal matrix (2.1), we can write problem (1.1) as follows

$$\begin{array}{ll} \textit{minimize} & f(x), \\ \textit{subject to} & P(x)^T Z(x) P(x) = 0, \\ & \hat{u} \leq x \leq \hat{v}, \end{array}$$

where

$$P(x) = (P_1(x), \cdots, P_m(x))^T$$

is a twice continuously differentiable function. The above problem is converted to the following equivalent problem by utilizing the penalty method [22]

$$\begin{array}{ll} \textit{minimize} & f(x) + \frac{\rho}{2}\|Z(x)P(x)\|^2, \\ \textit{subject to} & \hat{u} \leq x \leq \hat{v}, \end{array} \tag{2.2}$$

where $\rho > 0$ is a parameter.

Let

$$\tilde{\phi}(x; \rho) = f(x) + \frac{\rho}{2}\|Z(x)P(x)\|^2, \tag{2.3}$$

and then we will define a Lagrangian function associated with problem (2.2) as follows

$$L(x, \lambda_{\hat{u}}, \lambda_{\hat{v}}; \rho) = \tilde{\phi}(x; \rho) - \lambda_{\hat{u}}^T (x - \hat{u}) - \lambda_{\hat{v}}^T (\hat{v} - x), \tag{2.4}$$

where $\lambda_{\hat{u}}$ and $\lambda_{\hat{v}}$ are Lagrange multiplier vectors associated with the inequality constraints $x - \hat{u} \geq 0$ and $\hat{v} - x \geq 0$, respectively.

A point $x_* \in \boldsymbol{E}$ will be a local minimizer of problem (2.2) if there exists multiplier vectors $\lambda_{\hat{u}}(x_*) \in \Re^n_+$ and $\lambda_{\hat{v}}(x_*) \in \Re^n_+$ such that the following conditions are satisfied

$$\nabla\tilde{\phi}(x_*; \rho) - \lambda_{\hat{u}}(x_*) + \lambda_{\hat{v}}(x_*) = 0, \tag{2.5}$$

$$\lambda_{\hat{u}}^{(j)}(x_*^{(j)} - \hat{u}^{(j)}) = 0, \tag{2.6}$$

$$\lambda_{\hat{v}}^{(j)}(\hat{v}^{(j)} - x_*^{(j)}) = 0, \tag{2.7}$$

where

$$\nabla\tilde{\phi}(x_*; \rho) = \nabla f(x_*) + \rho \nabla P(x_*) Z(x_*) P(x_*).$$

Motivated by the interior point method introduced in [1] and used by [2,4,8,9], we define a diagonal scaling matrix $Y(x)$ whose diagonal elements are given by

$$y^{(j)}(x) = \begin{cases} \sqrt{(x^{(j)} - \hat{u}^{(j)})}, & \text{if } (\nabla\tilde{\phi}(x;\rho))^{(j)} \geq 0 \text{ and } \hat{u}^{(j)} > -\infty, \\ \sqrt{(\hat{v}^{(j)} - x^{(j)})}, & \text{if } (\nabla\tilde{\phi}(x;\rho))^{(j)} < 0 \text{ and } \hat{v}^{(j)} < +\infty, \\ 1, & \text{otherwise.} \end{cases} \tag{2.8}$$

Utilizing the matrix $Y(x)$, conditions (2.5)–(2.7) are equivalent to the following nonlinear system

$$Y^2(x)\nabla\tilde{\phi}(x;\rho) \quad = \quad 0. \tag{2.9}$$

Nonlinear Eq (2.9) is continuous but is not differentiable at some point $x \in \boldsymbol{E}$ for the following reasons:

- It may be non-differentiable when $y^{(j)} = 0$. To overcome this problem, we restrict $x \in int(\boldsymbol{E})$.
- It may be non-differentiable when $y^{(j)}$ has an infinite upper bound and a finite lower bound, and

$$(\nabla\tilde{\phi}(x;\rho))^{(j)} = 0.$$

To overcome this problem, we define a vector $\Psi(x)$ whose components

$$\Psi^{(j)}(x) = \frac{\partial((y^{(j)})^2)}{\partial x^{(j)}}, \quad j = 1, \cdots, n$$

are defined as follows

$$\Psi^{(j)}(x) = \begin{cases} 1, & \text{if } (\nabla\tilde{\phi}(x;\rho))^{(j)} \geq 0 \text{ and } \hat{u}^{(j)} > -\infty, \\ -1, & \text{if } (\nabla\tilde{\phi}(x;\rho))^{(j)} < 0 \text{ and } \hat{v}^{(j)} < +\infty\ , \\ 0, & \text{otherwise.} \end{cases} \tag{2.10}$$

When we apply Newton's technique to the system (2.9), we get

$$[Y^2(x)\nabla^2\tilde{\phi}(x;\rho) + diag(\nabla\tilde{\phi}(x;\rho))diag(\Psi(x))]\Delta x = -Y^2(x)\nabla\tilde{\phi}(x;\rho), \tag{2.11}$$

where

$$\nabla^2\tilde{\phi}(x;\rho) = H + \rho\nabla P(x)Z(x)\nabla P(x)^T, \tag{2.12}$$

and H is the Hessian of the objective function $f(x)$ or an approximation to it.

Assuming that $x \in int(\boldsymbol{E})$, then the matrix $Y(x)$ must be non-singular. Multiply both sides of Eq (2.9) by $Y^{-1}(x)$ and set

$$\Delta x = Y(x)d,$$

we have

$$[Y(x)\nabla^2\tilde{\phi}(x;\rho)Y(x) + diag(\nabla\tilde{\phi}(x;\rho))diag(\Psi(x))]d = -Y(x)\nabla\tilde{\phi}(x;\rho). \tag{2.13}$$

Notice that the step d_k, which is generated by system (2.13) is equivalent the step generated by solving the following quadratic programming subproblem

$$minimize \quad \tilde{\phi}(x;\rho) + (Y\nabla\tilde{\phi}(x;\rho))^T d + \tfrac{1}{2}d^T Bd, \tag{2.14}$$

where

$$B = Y(x)\nabla^2\tilde{\phi}(x;\rho)Y(x) + diag(\nabla\tilde{\phi}(x;\rho))diag(\Psi(x)). \tag{2.15}$$

Newton's approach has the advantage of being quadratically convergent under reasonable assumptions, but it has the drawback of requiring the initial point to be close to the solution. The nonmonotone trust-region globalization approach ensures convergence from any starting point. It is a crucial method for solving a smooth, nonlinear, unconstrained, or constrained optimization problem that can produce substantial global convergence.

For the purpose of solving problem (2.14), we introduce a detailed description of the nonmonotone trust-region algorithm.

2.2. A nonmonotone trust-region algorithm

The trust-region subproblem associated with the problem (2.14) is

$$\begin{array}{ll} minimize & q_k(Y_k d) = \tilde{\phi}(x_k;\rho_k) + (Y_k \nabla\tilde{\phi}(x_k;\rho_k))^T d, +\frac{1}{2} d^T B_k d, \\ subject\ to & \| d \| \leq \Delta_k, \end{array} \tag{2.16}$$

where $\Delta_k > 0$ is the radius of the trust region.

Using a dogleg method, which is a very cheap method, to solve subproblem (2.16) and obtain the step d_k. For more details, see [23].

- **To compute the step** d_k

In a dogleg method, the solution curve to the subproblem (2.16) is approximated by a piecewise linear function connecting the Newton point to the Cauchy point. The following algorithm explains the key phases of the dogleg approach to solve subproblem (2.16) and obtain d_k.

Algorithm 2.1. *Dogleg algorithm.*
Step 1. *Compute the parameter t_k^{cp} as follows:*

$$t_k^{cp} = \begin{cases} \frac{\|(Y_k \nabla\tilde{\phi}(x_k;\rho_k))\|^2}{(Y_k \nabla\tilde{\phi}(x_k;\rho_k))^T B_k (Y_k \nabla\tilde{\phi}(x_k;\rho_k))} & if\ \frac{\|Y_k \nabla\tilde{\phi}(x_k;\rho_k)\|^3}{(Y_k \nabla\tilde{\phi}(x_k;\rho_k))^T B_k (Y_k \nabla\tilde{\phi}(x_k;\rho_k))} \leq \Delta_k\ , \\ & and \quad (Y_k \nabla\tilde{\phi}(x_k;\rho_k))^T B_k (Y_k \nabla\tilde{\phi}(x_k;\rho_k)) > 0, \\ \frac{\Delta_k}{\|Y_k \nabla\tilde{\phi}(x_k;\rho_k)\|}, & otherwise. \end{cases} \tag{2.17}$$

Step 2. *Compute the Cauchy step*

$$d_k^{cp} = -t_k^{cp}(Y_k \nabla\tilde{\phi}(x_k;\rho_k)).$$

Step 3. *If*

$$\|d_k^{cp}\| = \Delta_k,$$

then set $d_k = d_k^{cp}$.
Else, If $Y_k \nabla\tilde{\phi}(x_k;\rho_k) + B_k d_k^{cp} = 0$, then set $d_k = d_k^{cp}$.
Else, compute Newton's step s^N by solving the following subproblem:

$$\min(Y_k \nabla\tilde{\phi}(x_k;\rho_k))^T d^N + \frac{1}{2} d^{N^T} B_k d^N.$$

End if.
Step 4. *If $\|d_k^N\| \leq \Delta_k$, then set $d_k = d_k^N$.*
Else, computing dogleg step between d_k^{cp} and d_k^N and compute d_k as follows

$$d_k = d_k^N + d_k^{cp}.$$

End if.

By using dogleg Algorithm 2.1, the step d_k satisfies the following condition, which is called a fraction-of-Cauchy decrease condition

$$q_k(0) - q_k(Y_k d_k) \geq \varphi[q_k(0) - q_k(Y_k d_k^{cp})] \tag{2.18}$$

for some $\varphi \in (0, 1]$.

When a condition is referred to as a fraction-of-Cauchy decrease, it indicates that the predicted reduction obtained by the step d_k is more than or equal to a fraction of the predicted decrease obtained by the $d_k cp$ (Cauchy step).

To ensure that the matrix Y_k is nonsingular, we need to guarantee that $x_{k+1} \in int(\boldsymbol{E})$. To do this, we need to a damping parameter τ_k.

- **To obtain damping parameter** τ_k

The main steps to obtain the damping parameter τ_k are clarified in the following algorithm:

Algorithm 2.2. *Damping parameter* τ_k.
Step 1. *Compute the parameter* α_k *as follows:*

$$\alpha_k^{(i)} = \begin{cases} \frac{\hat{u}^{(i)} - x_k^{(i)}}{Y_k^{(i)} d_k^{(i)}}, & \text{if } \hat{u}^{(i)} > -\infty \text{ and } Y_k^{(i)} d_k^{(i)} < 0, \\ 1, & \text{otherwise}. \end{cases}$$

Step 2. *Compute the parameter* β_k *as follows:*

$$\beta_k^{(i)} = \begin{cases} \frac{\hat{v}^{(i)} - x_k^{(i)}}{Y_k^{(i)} d_k^{(i)}}, & \text{if } \hat{v}^{(i)} < \infty \text{ and } Y_k^{(i)} d_k^{(i)} > 0, \\ 1, & \text{otherwise}. \end{cases}$$

Step 3. *Compute the damping parameter* τ_k *as follows*

$$\tau_k = \min\{1, \min_i\{\alpha_k^{(i)}, \beta_k^{(i)}\}\}. \tag{2.19}$$

Step 4. *Set*

$$x_{k+1} = x_k + \tau_k Y_k d_k.$$

To test the scaling step $\tau_k Y_k d_k$ to decide whether it will be accepted or not, a merit function is required. The merit function ties the objective function and the constraints in such a way that progress in the merit function means progress in solving the problem. The merit function that is used in our algorithm is the penalty function $\tilde{\phi}(x_k; \rho_k)$.

Let the actual reduction $Ared_k$ be defined as follows

$$Ared_k = \tilde{\phi}(x_k; \rho_k) - \tilde{\phi}(x_k + \tau_k Y_k d_k; \rho_k).$$

Additionally, $Ared_k$ can be expressed as follows

$$Ared_k = f(x_k) - f(x_k + \tau_k Y_k d_k) + \frac{\rho_k}{2}[\| Z_k P_k \|^2 - \| Z_{k+1} P_{k+1} \|^2]. \tag{2.20}$$

Let the predicted reduction $Pred_k$ be defined as follows

$$\begin{aligned} Pred_k &= q_k(0) - q_k(\tau_k Y_k d_k) \\ &= -(Y_k \nabla f_k)^T \tau_k d_k - \frac{1}{2}\tau_k^2 d_k^T G_k d_k + \frac{\rho_k}{2}[\| Z_k P_k \|^2 - \| Z_k(P_k + \nabla P_k^T Y_k \tau_k d_k) \|^2], \end{aligned} \tag{2.21}$$

where

$$G_k = Y_k P_k Y_k + diag(\nabla \tilde{\phi}(x_k; \rho_k)) diag(\eta_k).$$

- **To test $\tau_k Y_k d_k$ and update Δ_k**

Motivated by the nonmonotone trust-region strategy in [18], we define

$$\hat{r}_k = \frac{C_k - \tilde{\phi}(x_k + \tau_k Y_k d_k; \rho_k)}{q_k(0) - q_k(\tau_k Y_k d_k)}, \tag{2.22}$$

where

$$C_k = \begin{cases} \tilde{\phi}(x_k; \rho_k), & \text{if } k = 0, \\ \frac{\eta_{k-1} Q_{k-1} C_{k-1} + \tilde{\phi}(x_k; \rho_k)}{Q_k}, & \text{if } k \geq 1, \end{cases} \tag{2.23}$$

and

$$Q_k = \begin{cases} 1, & \text{if } k = 0, \\ \eta_{k-1} Q_{k-1} + 1, & \text{if } k \geq 1, \end{cases} \tag{2.24}$$

such that

$$0 \leq \eta_{\min} \leq \eta_{k-1} \leq \eta_{\max} \leq 1.$$

The following algorithm clarifies how the trial step will be tested and updated the trust region radius Δ_k:

Algorithm 2.3. *Test $\tau_k Y_k d_k$ and update Δ_k.*
Step 0. *Choose $0 < \theta_1 < \theta_2 \leq 1$, $\Delta_{max} > \Delta_{min}$, and $0 < \tilde{\alpha_1} < 1 < \tilde{\alpha_2}$.*
Step 1. *Compute Q_k using (2.24) and C_k using (2.23).*
Evaluate $\hat{r}_k$ using (2.22).
While $\hat{r}_k < \theta_1$, or $Pred_k \leq 0$.
Set $\Delta_k = \tilde{\alpha_1} \parallel d_k \parallel$.
Return to algorithm (2.1) to evaluate a new step d_k.
Step 2. *If $\theta_1 \leq \hat{r}_k < \theta_2$, then set $x_{k+1} = x_k + \tau_k Y_k d_k$.*
Set $\Delta_{k+1} = \max(\Delta_{min}, \Delta_k)$.
End if.
Step 3. *If $\hat{r}_k \geq \theta_2$, then set $x_{k+1} = x_k + \tau_k Y_k d_k$.*
Set $\Delta_{k+1} = \min\{\max\{\Delta_{min}, \tilde{\alpha_2}\Delta_k\}, \Delta_{max}\}$.
End if.

- **To update the parameter ρ_k**

To update ρ_k, a scheme proposed by [24] is used; and we will clarify this in the algorithm that follows:

Algorithm 2.4. *Updating ρ_k.*
Step 1. *Set $\rho_0 = 1$ and use Eq (2.21) to evaluate $Pred_k$.*
Step 2. *If*

$$Pred_k \geq \parallel Y_k \nabla P_k Z_k P_k \parallel \min\{\parallel Y_k \nabla P_k Z_k P_k \parallel, \Delta_k\}. \tag{2.25}$$

Then, set $\rho_{k+1} = \rho_k$.
Else, set $\rho_{k+1} = 2\rho_k$.
End if.

Finally, the nonmonotone trust-region algorithm is terminated, if

$$\| Y_k \nabla f_k \| + \| Y_k \nabla P_k Z_k P_k \| \leq \epsilon_1$$

or

$$\| d_k \| \leq \epsilon_2$$

for some $\epsilon_1 > 0$ and $\epsilon_2 > 0$.

- **Nonmonotone trust-region algorithm**

We will clarify the main steps of the nonmonotone trust-region algorithm to solve subproblem (2.16) in the algorithm that follows:

Algorithm 2.5. *The nonmonotone trust-region algorithm.*
Step 0. *Initial value*

Starting $x_0 \in int(\boldsymbol{E})$. *Compute matrices* Z_0, Y_0 *and* Ψ_0. *Set* $\rho_0 = 1$.
Choose $\epsilon_1, \epsilon_2, \tilde{\alpha_1}, \tilde{\alpha_2}, \theta_1, \theta_2$, *such that* $\epsilon_1 > 0, \epsilon_2 > 0, \tilde{\alpha_2} > 1 > \tilde{\alpha_1} > 0$ *and* $0 < \theta_1 < \theta_2 \leq 1$.
Choose Δ_0, Δ_{min}, *and* Δ_{max}, *such that* $\Delta_{min} \leq \Delta_0 \leq \Delta_{max}$.
Set $k = 0$.

Step 1. *If* $\| Y_k \nabla f_k \| + \| Y_k \nabla P_k Z_k P_k \| \leq \epsilon_1$, *then stop.*
Step 2. *Evaluating the trial step* d_k *using the Algorithm 2.1.*
Step 3. *Stop and end the algorithm if* $\| d_k \| \leq \epsilon_2$.
Step 4. *Compute both* τ_k *and* Y_k *using Algorithm 2.2 and Eq* (2.8), *respectively. Set* $x_{k+1} = x_k + \tau_k Y_k d_k$.
Step 5. *Utilize* (2.1) *to evaluate* Z_{k+1}.
Step 6. *To test the scaling step and update* Δ_k*:*

i) Computing Q_k *using* (2.24) *and* C_k *using* (2.23).
ii) Compute $\hat{r}_k$ *using* (2.22).
iii) Using Algorithm 2.3 to test the scaling step $\tau_k Y_k d_k$ *and update the radius of the trust-region* Δ_k.

Step 7. *Updating the parameter* ρ_k *using Algorithm 2.4.*
Step 8. *Utilize* (2.10) *to evaluate* Ψ_{k+1}.
Step 9. *Set* $k = k + 1$ *and return to Step 1.*

2.3. Nonmonotone trust-region active-set penalty algorithm

The main steps for the NTRAI algorithm to solve problem (1.1) will be clarified in the following algorithm:

Algorithm 2.6. *NTRAI algorithm.*
Step 1. *Utilize active-set strategy and penalty method to convert a nonlinearly constrained optimization problem* (1.1) *to an unconstrained optimization problem with bounded variables* (2.2).
Step 2. *Utilize an interior-point method and a diagonal scaling matrix* $Y(x)$ *given in* (2.8), *and the first-order necessary conditions* (2.5)–(2.7) *equivalent to the nonlinear system in* (2.9).

Step 3. *Utilize Newton's method to solve the nonlinear system* (2.9) *and obtain the equivalent subproblem* (2.14).
Step 4. *Solve subproblem* (2.14) *using nonmonotone trust-region Algorithm 2.5.*

The global convergence analysis for NTRAI Algorithm 2.6 is conducted in the following section.

3. Analysis of global convergence

In this section, a global convergence analysis for the NTRAI Algorithm 2.6 to solve problem (1.1) will be presented. First, we will introduce the necessary assumptions that are requested to prove the theory of global convergence for the NTRAI Algorithm 2.6. Second, we will introduce some lemmas that are required to prove the main results. Third, we will study the iteration sequence convergence when ρ_k is unbounded and bounded, respectively. Finally, the main global convergence results for the NTRAI Algorithm 2.6 will be proved.

3.1. Necessary assumptions

Let $\{x_k\}$ be the sequence of points generated by the NTRAI Algorithm 2.6 and let Ω be a convex subset of $\Re^n$ that contains all iterates $x_k \in int(\boldsymbol{E})$ and $x_k + \tau_k Y_k d_k \in int(\boldsymbol{E})$, for all trial steps d_k. On the set Ω, we assume the following assumptions, under which the global convergence theory will be proved.

- **Assumptions:**

[As_1**]** For all $x \in \Omega$, the functions $f(x)$ and $P(x)$ are at least twice continuously differentiable.
[As_2**]** All of $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$, $P(x)$, and $\nabla P(x)$ are uniformly bounded in Ω.
[As_3**]** The sequence of Hessian matrices $\{B_k\}$ is bounded.

Some lemmas are required to prove the main global convergence theory. These lemmas are introduced in the following section:

3.2. Required lemmas

We shall introduce some lemmas that are necessary to support the main results.

Lemma 3.1. *Under assumptions As_1–As_3, there exists a constant $K_1 > 0$ such that,*

$$Pred_k \geq K_1 \tau_k \parallel Y_k \nabla \tilde{\phi}(x_k; \rho_k) \parallel \min\{\Delta_k, \frac{\parallel Y_k \nabla \tilde{\phi}(x_k; \rho_k) \parallel}{\parallel B_k \parallel}\}. \tag{3.1}$$

Proof. Since the fraction-of-Cauchy decrease condition (2.18) is satisfied by the trial step d_k, then we will consider the following two cases:
i) If

$$d_k^{cp} = -\frac{\Delta_k}{\parallel Y_k \nabla \tilde{\phi}(x_k; \rho_k) \parallel} (Y_k \nabla \tilde{\phi}(x_k; \rho_k))$$

and

$$\parallel Y_k \nabla \tilde{\phi}(x_k; \rho_k) \parallel^3 \geq \Delta_k [(Y_k \nabla \tilde{\phi}(x_k; \rho_k))^T B_k (Y_k \nabla \tilde{\phi}(x_k; \rho_k))],$$

then

$$\begin{aligned} q_k(0) - q_k(Y_k d_k^{cp}) &= -(Y_k \nabla \tilde{\phi}(x_k;\rho_k))^T d_k^{cp} - \frac{1}{2} d_k^{cp^T} B_k d_k^{cp} \\ &= \frac{\Delta_k}{\| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|} \| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|^2 \\ &\quad - \frac{1}{2} \frac{\Delta_k^2}{\| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|^2} ((Y_k \nabla \tilde{\phi}(x_k;\rho_k))^T B_k (Y_k \nabla \tilde{\phi}(x_k;\rho_k))) \\ &\geq \frac{1}{2} \Delta_k \| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \| . \end{aligned} \tag{3.2}$$

ii) If

$$d_k^{cp} = -\frac{\| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|^2}{(Y_k \nabla \tilde{\phi}(x_k;\rho_k))^T B_k (Y_k \nabla \tilde{\phi}(x_k;\rho_k))} (Y_k \nabla \tilde{\phi}(x_k;\rho_k))$$

and

$$\| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|^3 \leq \Delta_k ((Y_k \nabla \tilde{\phi}(x_k;\rho_k))^T B_k (Y_k \nabla \tilde{\phi}(x_k;\rho_k))),$$

then we have

$$\begin{aligned} q_k(0) - q_k(Y_k d_k^{cp}) &= -(Y_k \nabla \tilde{\phi}(x_k;\rho_k))^T d_k^{cp} - \frac{1}{2} d_k^{cp^T} B_k d_k^{cp} \\ &= \frac{1}{2} \frac{\| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|^4}{(Y_k \nabla \tilde{\phi}(x_k;\rho_k))^T B_k (Y_k \nabla \tilde{\phi}(x_k;\rho_k))} \\ &\geq \frac{\| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|^2}{2 \| B_k \|}. \end{aligned} \tag{3.3}$$

From inequalities (2.18), (3.2), and (3.3), we have

$$q_k(0) - q_k(Y_k d_k) \geq K_1 \| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \| \min\{\Delta_k, \frac{\| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|}{\| B_k \|}\}. \tag{3.4}$$

From inequality (3.4) and the following fact

$$q_k(0) - q_k(Y_k \tau_k d_k) \geq \tau_k [q_k(0) - q_k(Y_k d_k)],$$

where $0 \leq \tau_k \leq 1$, then we have

$$q_k(0) - q_k(Y_k \tau_k d_k) \geq K_1 \tau_k \| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \| \min\{\Delta_k, \frac{\| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|}{\| B_k \|}\}.$$

From (2.21), we have

$$Pred_k = q(0) - q(Y_k \tau_k d_k).$$

Hence

$$Pred_k \geq K_1 \tau_k \| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \| \min\{\Delta_k, \frac{\| Y_k \nabla \tilde{\phi}(x_k;\rho_k) \|}{\| B_k \|}\}.$$

This completes the proof. □

Lemma 3.2. *Under assumptions As_1 and As_3, then $Z(x)P(x)$ is Lipschitz continuous in Ω.*

Proof. The proof of this lemma similar [21, Lemma 4.1].

We can verify that $\nabla P(x)Z(x)P(x)$ is Lipschitz continuous in Ω and $\|Z(x)P(x)\|^2$ is differentiable from Lemma 3.2. □

Lemma 3.3. *At any iteration k, we have*

$$Z_{k+1} = Z_k + A_k, \tag{3.5}$$

where $A(x_k) \in \Re^{m\times m}$ *is a diagonal matrix whose diagonal entries are defined as follows*

$$(a_k)_i = \begin{cases} 1, & \text{if } (P_k)_i < 0 \text{ and } (P_{k+1})_i \geq 0, \\ -1, & \text{if } (P_k)_i \geq 0 \text{ and } (P_{k+1})_i < 0, \\ 0, & \text{otherwise}, \end{cases} \tag{3.6}$$

where $i = 1, 2, \cdots, m$.

Proof. See [6, Lemma 6.2]. □

Lemma 3.4. *Under assumptions* As_1–As_3*, there exists a constant* $K_2 > 0$*, such that*

$$\|A_k P_k\| \leq K_2 \|d_k\|. \tag{3.7}$$

Proof. See [6, Lemma 6.3]. □

Lemma 3.5. *Under assumptions* As_1–As_3*, there exists a constant* $K_3 > 0$*, such that*

$$| Ared_k - Pred_k | \leq K_3 \tau_k \rho_k \|d_k\|^2. \tag{3.8}$$

Proof. From Eqs (2.20) and (3.5), we have

$$Ared_k = f(x_k) - f(x_k + \tau_k Y_k d_k) + \frac{\rho_k}{2}[P_k^T Z_k P_k - P(x_k + \tau_k Y_k d_k)^T (Z_k + A_k) P(x_k + \tau_k Y_k d_k)].$$

Subtracting the above equation from (2.21), and using Cauchy-Schwarz inequality, we have

$$\begin{aligned} |Ared_k - Pred_k| \leq & \frac{\tau_k^2}{2} \mid d_k^T Y_k (\nabla^2 f(x_k) - \nabla^2 f(x_k + \xi_1 \tau_k Y_k d_k)) Y_k d_k \mid \\ & + \frac{\tau_k^2}{2} \mid d_k^T diag(\nabla \tilde{\phi}(x_k; \rho_k)) diag(\Psi) d_k \mid \\ & + \rho_k \tau_k \mid Y_k (\nabla P_k - \nabla P(x_k + \xi_2 \tau_k Y_k d_k)) Z_k P_k d_k \mid \\ & + \frac{\rho_k \tau_k^2}{2} \mid d_k^T Y_k [\nabla P_k Z_k \nabla P_k^T - \nabla P(x_k + \xi_2 Y_k \tau_k d_k) Z_k \nabla P(x_k + \xi_2 Y_k \tau_k d_k)^T] Y_k d_k \mid \\ & + \frac{\rho_k \tau_k^2}{2} \|A_k P_k\|^2 + \rho_k \tau_k \mid Y_k \nabla P(x_k + \xi_2 Y_k \tau_k d_k) A_k P_k d_k \mid \\ & + \frac{\rho_k \tau_k^2}{2} \mid d_k^T Y_k [\nabla P(x_k + \xi_2 Y_k \tau_k d_k) A_k \nabla P(x_k + \xi_2 Y_k \tau_k d_k)^T] Y_k d_k \mid \end{aligned}$$

for some ξ_1 and $\xi_2 \in (0, 1)$. From assumptions As_1–As_3 and using Lemma 3.4, the proof is completed. □

The following section is devoted to the analysis of global convergence for NTRAI Algorithm 2.6 when ρ_k is unlimited.

3.3. Global convergence when ρ_k is unbounded

Observe that we do not assume that $\nabla P(x)$ has full column rank for all $x \in \Omega$ in assumptions As_1-As_3; therefore, we may have alternative types of stationary points. The definitions that follow describe these stationary spots.

Definition 3.1. *(A Fritz John (FJ) point.) If there is $\omega_* \in \Re$ and a Lagrange multiplier vector $\sigma_* \in \Re^m$ that is not all zero, then the point $x_* \in \Re$ is said to be a FJ point if the following conditions are satisfied*

$$\omega_* Y_* \nabla f(x_*) + Y_* \nabla P(x_*) \sigma_* = 0, \tag{3.9}$$

$$Z(x_*) P(x_*) = 0, \tag{3.10}$$

$$(\sigma_*)_i P_i(x_*) = 0, \quad i = 1, 2, \cdots, m \tag{3.11}$$

$$\omega_*,\ (\sigma_*)_i \geq 0, \quad i = 1, 2, \cdots, m. \tag{3.12}$$

The conditions (3.9)–(3.12) are referred to as FJ conditions. See [25] for further information.

The FJ conditions are referred to as a Karush-Kuhn-Tucker (KKT) conditions, and the point $(x_*, 1, \frac{\sigma_*}{\omega_*})$ is referred to as the KKT point if $\omega_* \neq 0$.

Definition 3.2. *(Infeasible Fritz John (IFJ) point.) If there is $\omega_* \in \Re$ and a Lagrange multiplier vector $\sigma_* \in \Re^m$ that is not all zero, then the point $x_* \in \Re$ is said to be a IFJ point if the following conditions satisfy*

$$\omega_* Y_* \nabla f(x_*) + Y_* \nabla P(x_*) \sigma_* = 0, \tag{3.13}$$

$$Y(x_*) \nabla P(x_*) Z(x_*) P(x_*) = 0, \quad \textit{but} \ \ \|Z(x_*) P(x_*)\| > 0, \tag{3.14}$$

$$(\sigma_*)_i P_i(x_*) = 0, \quad i = 1, 2, \cdots, m, \tag{3.15}$$

$$\omega_*,\ (\sigma_*)_i \geq 0, \quad i = 1, 2, \cdots, m. \tag{3.16}$$

The conditions (3.13)–(3.16) are referred to as IFJ conditions. See [25] for further information.

The IFJ conditions are referred to as the infeasible KKT conditions and the point $(x_*, 1, \frac{\sigma_*}{\omega_*})$ is referred to as the infeasible KKT point if $\omega_* \neq 0$.

The next two lemmas provide conditions that are equivalent to those stated in Definitions 3.1 and 3.2.

Lemma 3.6. *Under assumptions As_1–As_3, there exists $\{x_{k_i}\} \subseteq \{x_k\}_{k \geq 0}$, satisfies IFJ conditions if:*
1) $\lim_{k_i \to \infty} \|Z_{k_i} P_{k_i}\| > 0$.
2) $\lim_{k_i \to \infty} \left\{ \min_d \left\{ \|Z_{k_i}(P_{k_i} + \nabla P_{k_i}^T Y_{k_i} \tau_{k_i} d)\|^2 \right\} \right\} = \lim_{k_i \to \infty} \|Z_{k_i} P_{k_i}\|^2$.

Proof. The proof of this lemma is similar to the proof of [2, Lemma 3.1]. □

Lemma 3.7. *Under assumptions As_1–As_3, there exists $\{x_{k_i}\} \subseteq \{x_k\}_{k \geq 0}$ satisfies FJ conditions if:*
1) For all k_i, $\|Z_{k_i} P_{k_i}\| > 0$ and $\lim_{k_i \to \infty} Z_{k_i} P_{k_i} = 0$.
2) $\lim_{k_i \to \infty} \left\{ \min_d \left\{ \frac{\|Z_{k_i}(P_{k_i} + \nabla P_{k_i}^T Y_{k_i} \tau_{k_i} d)\|^2}{\|Z_{k_i} P_{k_i}\|^2} \right\} \right\} = 1$.

Proof. The proof of this lemma similar the proof of [26, Lemma 3.2]. □

According to the Algorithm 2.4, the sequence of parameters $\{\rho_k\}$ is only unlimited when there is an infinite subsequence of indexes k_i indexing iterates of acceptable steps that fulfill, for every $k \in \{k_i\}$,

$$Pred_k < \|Y_k \nabla P_k Z_k P_k\| \min\{\|Y_k \nabla P_k Z_k P_k\|, \Delta_k\}. \tag{3.17}$$

A subsequence of iterates $\{x_k\}$ satisfies the FJ conditions or IFJ conditions if $\rho_k \to \infty$ as $k \to \infty$. This is demonstrated by the next two lemmas.

Lemma 3.8. *Under assumptions As_1–As_3 and $\rho_k \to \infty$ as $k \to \infty$.*

If $\|Z_k P_k\| \geq \varepsilon > 0$ for all $k \in \{k_i\}$, a subsequence of the iteration sequence with index k_i exists and fulfills the IFJ conditions in the limit.

Proof. For simplification, we assume k_i is k. This lemma's proof is due to a contradiction, so, assume that there is no subsequence of iterates that satisfies the IFJ conditions in the limit. From Lemma 3.6 and Definition 3.2, we have

$$|\ \|Z_k P_k\|^2 - \|Z_k(P_k + \nabla P_k^T Y_k \tau_k d)\|^2\ | \geq \varepsilon_1$$

and

$$\|Y_k \nabla P_k Z_k P_k\| \geq \varepsilon_2,$$

respectively. Hence

$$\begin{aligned}\|Y_k \nabla \tilde{\phi}(x_k; \rho_k)\| &= \|Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k)\| \\ &\geq \rho_k \|Y_k \nabla P_k Z_k P_k\| - \|Y_k \nabla f_k\| \\ &\geq \rho_k \varepsilon_2 - \|Y_k \nabla f_k\| \geq \rho_k \varepsilon_2.\end{aligned}$$

From (2.15) and (2.12), we have

$$\begin{aligned}\|B_k\| &= \|Y_k P_k Y_k + \rho_k Y_k \nabla P_k Z_k \nabla P_k^T Y_k + diag(\nabla \tilde{\phi}(x;\rho)) diag(\Psi(x))\| \\ &\leq \rho_k(\|Y_k \nabla P_k Z_k \nabla P_k^T Y_k\| + diag(\frac{1}{\rho_k} \nabla f_k + \nabla P_k Z_k P_k) diag(\Psi(x))) + \|Y_k P_k Y_k\|.\end{aligned} \tag{3.18}$$

From inequalities (3.1) and (3.18), we have

$$Pred_k \geq K_1 \rho_k \tau_k \varepsilon_2 \min\{\Delta_k, \frac{\varepsilon_2}{\|Y_k \nabla P_k Z_k \nabla P_k^T Y_k\| + diag(\nabla P_k Z_k P_k) diag(\Psi(x))}\} \tag{3.19}$$

for k sufficiently large. There is an infinite number of acceptable iterates at which inequality (3.17) holds since $\rho_k \to \infty$. From inequalities (3.17) and (3.19), we have

$$\begin{aligned}&\|Y_k \nabla P_k Z_k P_k\| \min\{\|Y_k \nabla P_k Z_k P_k\|, \Delta_k\} \geq K_1, \\ &\qquad \rho_k \tau_k \varepsilon_2 \min\{\Delta_k, \frac{\varepsilon_2}{\|Y_k \nabla P_k Z_k \nabla P_k^T Y_k\| + diag(\nabla P_k Z_k P_k) diag(\Psi(x))}\}.\end{aligned}$$

According to the assumption As_2, the preceding inequality's right side tends toward infinity as $k \to \infty$ and the left-hand side is bounded such that

$$\lim_{k \to \infty} \tau_k = 1.$$

This result gives a contradiction unless $\rho_k\Delta_k$ is bounded. That is $\Delta_k \to 0$.

Now, if $\rho_k \to \infty$, at $k \to \infty$, we will consider two cases:

First, if

$$\|Z_kP_k\|^2 - \|Z_k(P_k + \nabla P_k^T\tau_kY_kd_k)\|^2 \geq \varepsilon_1,$$

then

$$\rho_k\{\|Z_kP_k\|^2 - \|Z_k(P_k + \nabla P_k^T\tau_kY_kd_k)\|^2\} \geq \rho_k\varepsilon_1 \to \infty.$$

Hence, $Pred_k \to \infty$ using assumptions As_2 and As_3. In other words, the left-hand side of inequality (3.17) goes to infinity since $k \to \infty$, but the right-hand side goes to zero since $\Delta_k \to 0$. Then we get a contradiction with the assumption in this case.

Second, if

$$\|Z_kP_k\|^2 - \|Z_k(P_k + \nabla P_k^T\tau_kY_kd_k)\|^2 \leq -\varepsilon_1,$$

then we have

$$\rho_k\{\|Z_kP_k\|^2 - \|Z_k(P_k + \nabla P_k^T\tau_kY_kd_k)\|^2\} \leq -\rho_k\varepsilon_1 \to -\infty.$$

Similar to the first case, $Pred_k \to -\infty$, but $Pred_k > 0$ and this gives a contradiction. These two contradictions prove the lemma. □

The following lemma shows that the behavior of NTRAI Algorithm 2.6 when

$$\liminf_{k\to\infty} \|Z_kP_k\| = 0$$

and $\rho_k \to \infty$ as $k \to \infty$.

Lemma 3.9. *Under assumptions As_1–As_3 and at $\rho_k \to \infty$ as $k \to \infty$, then there exists a subsequence $\{k_i\}$ of iterates that satisfies the FJ conditions in the limit if $\|Z_kP_k\| > 0$ for all $k \in \{k_i\}$ and*

$$\lim_{k_i\to\infty} \|Z_{k_i}P_{k_i}\| = 0.$$

Proof. For simplification, we assume k_i is k. Assume that there is no subsequence of iterations that fulfills the FJ conditions in the limit since the demonstration of this lemma relies on contradiction. From Lemma 3.7, we have

$$\frac{|\ \|Z_kP_k\|^2 - \|Z_k(P_k + \nabla P_k^T\tau_kY_kd_k)\|^2\ |}{\|Z_kP_k\|^2} \geq \varepsilon > 0 \tag{3.20}$$

for each k very large.

Now we will consider three cases if $\rho_k \to \infty$, at $k \to \infty$:

First, if

$$\liminf_{k\to\infty} \frac{d_k}{\|Z_kP_k\|} = 0,$$

then there is a contradiction with inequality (3.20).

Second, if

$$\limsup_{k\to\infty} \frac{d_k}{\|Z_kP_k\|} = \infty.$$

From subproblem (2.16), we have

$$Y_k\nabla f_k + \rho_k Y_k \nabla P_k Z_k P_k = -(B_k + \upsilon_k I)d, \tag{3.21}$$

where $0 < \upsilon_k$ is the Lagrange multiplier vector of the trust region constraint. Hence, we can write inequality (3.1) as follows

$$Pred_k \geq K_1\tau_k \parallel Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k) \parallel \min\{\Delta_k, \frac{\parallel (B_k + \upsilon_k I)d_k \parallel}{\parallel B_k \parallel}\}.$$

From (2.15) and the above inequality, we have

$$Pred_k \geq K_1\tau_k \parallel Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k) \parallel \min\{\Delta_k, \frac{\parallel (Y_k\nabla P_k Z_k \nabla P_k^T Y_k + \hat{G}_k + \frac{\upsilon_k}{\rho_k} I)d_k \parallel}{\parallel Y_k \nabla P_k Z_k \nabla P_k^T Y_k + \hat{G}_k \parallel}\}, \tag{3.22}$$

where

$$\hat{G} = \frac{1}{\rho_k} Y_k P_k Y_k + diag(\frac{1}{\rho_k}\nabla f_k + \nabla P_k Z_k P_k) diag(\Psi(x)).$$

As a result, there are an infinite number of acceptable steps at which inequality (3.17) holds. From inequality (3.17), we have

$$Pred_k < \|Y_k \nabla P_k\|^2 \|Z_k P_k\|^2, \tag{3.23}$$

and using inequalities (3.22) and (3.23), we have

$$K_1\tau_k \parallel Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k) \parallel \min\{\Delta_k, \frac{\parallel (Y_k\nabla P_k Z_k \nabla P_k^T Y_k + \hat{G}_k + \frac{\upsilon_k}{\rho_k} I)d_k \parallel}{\parallel Y_k \nabla P_k Z_k \nabla P_k^T Y_k + \hat{G}_k \parallel}\} < \kappa^2 \|Z_k P_k\|^2,$$

where

$$\kappa = sup_{x\in\Omega} \|Y_k \nabla P_k\|.$$

Dividing the above inequality by $\|Z_k P_k\|$, then

$$K_1\tau_k \parallel Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k) \parallel \min\{\frac{\Delta_k}{\|Z_k P_k\|}, \frac{\parallel (Y_k\nabla P_k Z_k \nabla P_k^T Y_k + \hat{G}_k + \frac{\upsilon_k}{\rho_k} I)d_k \parallel}{\parallel Y_k \nabla P_k Z_k \nabla P_k^T Y_k + \hat{G}_k \parallel \|Z_k P_k\|}\} < \kappa^2 \|Z_k P_k\|, \tag{3.24}$$

As $k \to \infty$, the right-hand side of the previous inequality goes to zero. That is,

$$\parallel Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k) \parallel \frac{\parallel (Y_k\nabla P_k Z_k \nabla P_k^T Y_k + \hat{G}_k + \frac{\upsilon_k}{\rho_k} I)d_k \parallel}{\parallel Y_k \nabla P_k Z_k \nabla P_k^T Y_k + \hat{G}_k \parallel \|Z_k P_k\|}$$

is bounded along the subsequence $\{k_i\}$ where

$$\lim_{k_i\to\infty} \frac{d_{k_i}}{\|Z_{k_i} P_{k_i}\|} = \infty.$$

That is, either $\frac{d_{k_i}}{\|Z_{k_i}P_{k_i}\|}$ lies in the null space of

$$Y_{k_i}\nabla P_{k_i}Z_{k_i}\nabla P_{k_i}^T Y_{k_i} + \frac{\upsilon_{k_i}}{\rho_{k_i}} I$$

or

$$\| Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k) \| \to 0.$$

The first possibility only occurs when $\frac{\upsilon_{k_i}}{\rho_{k_i}} \to 0$ as $k_i \to \infty$ and $\frac{d_{k_i}}{\|Z_{k_i}P_{k_i}\|}$ lies in the matrix's null space. $Y_{k_i}\nabla P_{k_i}Z_{k_i}\nabla P_{k_i}^T Y_{k_i}$ contradicts assumption (3.20) and implies that a subsequence of $\{k_i\}$ satisfies the FJ conditions in the limit.

The second possibility implies

$$\| Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k) \| \to 0$$

as $k_i \to \infty$. Hence, $\rho_{k_i}\|Y_{k_i}\nabla P_{k_i}Z_{k_i}P_{k_i}\|$ must be bounded, and we have $\nabla f_{k_i} = 0$. This implies that a subsequence of $\{k_i\}$ satisfies the FJ conditions in the limit.

Third, if

$$\limsup_{k\to\infty} \frac{d_k}{\|Z_k P_k\|} < \infty$$

and

$$\liminf_{k\to\infty} \frac{d_k}{\|Z_k P_k\|} > 0.$$

Therefore $\|d_k\| \to 0$. As a result, in the second case, as $k \to \infty$, the right-hand side of (3.24) goes to zero. Hence

$$\| Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k) \| \frac{\| (Y_k\nabla P_k Z_k \nabla P_k^T Y_k + diag(\nabla P_k Z_k P_k) diag(\Psi(x)) + \frac{\upsilon_k}{\rho_k} I) d_k \|}{\| Y_k \nabla P_k Z_k \nabla P_k^T Y_k + diag(\nabla P_k Z_k P_k) diag(\Psi(x)) \| \, \|Z_k P_k\|} \to 0.$$

This implies that, either

$$\| Y_k(\nabla f_k + \rho_k \nabla P_k Z_k P_k) \| \to 0$$

or

$$\frac{\| (Y_k\nabla P_k Z_k \nabla P_k^T Y_k + diag(\nabla P_k Z_k P_k) diag(\Psi(x)) + \frac{\upsilon_k}{\rho_k} I) d_k \|}{\| Y_k \nabla P_k Z_k \nabla P_k^T Y_k + diag(\nabla P_k Z_k P_k) diag(\Psi(x)) \| \, \|Z_k P_k\|} \to 0.$$

In a similar way to the above second case, we can prove that a subsequence of $\{k_i\}$ satisfies the FJ conditions in the limit. The proof is now complete. □

3.4. Convergence when ρ_k is bounded

We continue our analysis in this section on the assumption that the penalty parameter ρ_k is bounded. In other words, we proceed with our analysis assuming that there is an integer $\bar{k}$ such that $\rho_k = \bar{\rho} < \infty$ for all $k \geq \bar{k}$.

Lemma 3.10. *Assume that $\{x_k\}$ is the sequence of iterations generated by the NTRAI algorithm, then we have*

$$\tilde{\phi}(x_{k+1}; \bar{\rho}) \leq C_{k+1} \leq C_k. \tag{3.25}$$

Proof. Let iterate k be a successive iterate, then from Algorithm 2.3, we have

$$\hat{r}_k = \frac{C_k - \tilde{\phi}(x_k + \tau_k Y_k d_k; \bar{\rho})}{Pred_k} \geq \theta_1.$$

That is,

$$C_k - \tilde{\phi}(x_k + \tau_k Y_k d_k; \bar{\rho}) \geq \theta_1 Pred_k$$

and by using inequality (3.1), we have

$$\tilde{\phi}(x_{k+1}; \bar{\rho}) \leq C_k - K_1\theta_1\tau_k \parallel Y_k \nabla\tilde{\phi}(x_k; \bar{\rho}) \parallel \min\{\Delta_k, \frac{\parallel Y_k \nabla\tilde{\phi}(x_k; \bar{\rho}) \parallel}{\parallel B_k \parallel}\}. \tag{3.26}$$

From (2.23), (2.24) and using inequality (3.26), then we have

$$\begin{aligned}
C_{k+1} &= \frac{\eta_k Q_k C_k + \tilde{\phi}(x_{k+1}; \bar{\rho})}{Q_{k+1}} \\
&\leq \frac{\eta_k Q_k C_k + C_k - K_1\theta_1\tau_k \parallel Y_k \nabla\tilde{\phi}(x_k; \bar{\rho}) \parallel \min\{\Delta_k, \frac{\|Y_k \nabla\tilde{\phi}(x_k;\bar{\rho})\|}{\|B_k\|}\}}{Q_{k+1}} \\
&\leq \frac{C_k(\eta_k Q_k + 1) - K_1\theta_1\tau_k \parallel Y_k \nabla\tilde{\phi}(x_k; \bar{\rho}) \parallel \min\{\Delta_k, \frac{\|Y_k \nabla\tilde{\phi}(x_k;\bar{\rho})\|}{\|B_k\|}\}}{Q_{k+1}} \\
&\leq C_k - \frac{K_1\theta_1\tau_k \parallel Y_k \nabla\tilde{\phi}(x_k; \bar{\rho}) \parallel \min\{\Delta_k, \frac{\|Y_k \nabla\tilde{\phi}(x_k;\bar{\rho})\|}{\|B_k\|}\}}{Q_{k+1}}.
\end{aligned}$$

That is,

$$C_{k+1} \leq C_k. \tag{3.27}$$

From (2.23), we have

$$C_{k+1} = \frac{\eta_k Q_k C_k + \tilde{\phi}(x_{k+1}; \bar{\rho})}{Q_{k+1}}.$$

Using (2.24), then we have

$$C_{k+1}(\eta_k Q_k + 1) = \eta_k Q_k C_k + \tilde{\phi}(x_{k+1}; \bar{\rho}).$$

Hence

$$C_{k+1} - C_k = \frac{\tilde{\phi}(x_{k+1}; \bar{\rho}) - C_{k+1}}{\eta_k Q_k}. \tag{3.28}$$

From (3.27) and (3.28), we have

$$\tilde{\phi}(x_{k+1}; \bar{\rho}) \leq C_{k+1}. \tag{3.29}$$

From (3.27) and (3.29), we have

$$\tilde{\phi}(x_{k+1}; \bar{\rho}) \leq C_{k+1} \leq C_k.$$

This completes the proof. □

Lemma 3.11. *Under assumptions As_1–As_3 and at any iteration k at which*

$$\|Y_k\nabla\tilde{\phi}(x_k;\bar{\rho}_k)\| + \|Y_k\nabla P_k D_k P_k)\| > \epsilon_1.$$

Then, there exists a constant $K_4 > 0$ such that

$$Pred_k \geq K_4\tau_k\Delta_k. \tag{3.30}$$

Proof. From (2.12), (2.15), and using assumptions As_1–As_3, then there exists a constant $b_1 > 0$ such that $\|B_k\| \leq b_1$ for all k. Let

$$\|Y_k\nabla\tilde{\phi}(x_k;\bar{\rho}_k)\| > \frac{\epsilon_1}{2}$$

and using inequality (3.1), we have

$$\begin{aligned} Pred_k &\geq K_1\tau_k \parallel Y_k\nabla\tilde{\phi}(x_k;\bar{\rho}_k) \parallel \min\{\Delta_k, \frac{\parallel Y_k\nabla\tilde{\phi}(x_k;\bar{\rho}_k) \parallel}{\parallel B_k \parallel}\} \\ &\geq \frac{1}{2}K_1\tau_k\epsilon_1 \min\{1, \frac{\epsilon_1}{2b_1\Delta_{max}}\}\Delta_k \\ &\geq K_4\tau_k\Delta_k, \end{aligned}$$

where

$$K_4 = \frac{1}{2}K_1\epsilon_1 \min\{1, \frac{\epsilon_1}{2b_1\Delta_{max}}\}.$$

This completes the proof. □

Lemma 3.12. *Under assumptions As_1–As_3 and if*

$$\|Y_k\nabla\tilde{\phi}(x_k;\bar{\rho}_k)\| + \|Y_k\nabla P_k Z_k P_k)\| > \epsilon_1,$$

then an acceptable step is found after finitely many trials. That is, the condition

$$\frac{C_k - \tilde{\phi}(x_{k+1};\bar{\rho}_k)}{Pred_k} \geq \theta_1$$

will be satisfied.

Proof. Since

$$\|Y_k\nabla\tilde{\phi}(x_k;\bar{\rho}_k)\| + \|Y_k\nabla P_k Z_k P_k)\| > \epsilon_1,$$

then from Lemmas 3.5, 3.10, and 3.11, we have

$$\begin{aligned} \left| \frac{C_k - \tilde{\phi}(x_{k+1};\bar{\rho}_k)}{Pred_k} - 1 \right| &\leq \left| \frac{\tilde{\phi}(x_k;\bar{\rho}_k) - \tilde{\phi}(x_{k+1};\bar{\rho}_k)}{Pred_k} - 1 \right| \\ &= \left| \frac{Ared_k}{Pred_k} - 1 \right| \\ &= \frac{| Ared_k - Pred_k |}{Pred_k} \\ &\leq \frac{K_3\bar{\rho}\tau_k\Delta_k^2}{K_4\tau_k\Delta_k} \leq \frac{K_3\bar{\rho}\Delta_k}{K_4}. \end{aligned}$$

As a result of step d_k being rejected, Δ_k is now small, and after a finite number of trials, the acceptance rule will finally be satisfied. That is

$$\frac{C_k - \tilde{\phi}(x_{k+1};\bar{\rho}_k)}{Pred_k} \geq \theta_1$$

and this ends the proof. □

Lemma 3.13. *Under assumptions As_1–As_3 and if*

$$\|Y_k\nabla\tilde{\phi}(x_k;\bar{\rho}_k)\| + \|Y_k\nabla P_k Z_k P_k)\| > \epsilon_1,$$

at a given iteration k, the j^{th} trial step satisfies

$$\|d_{k^j}\| \leq \frac{(1-\theta_1)K_4}{2\bar{\rho}K_3}, \tag{3.31}$$

then it has to be accepted.

Proof. Since the proof of this lemma is by a contradiction, we assume that the inequality (3.31) is true and the step d_{k^j} is rejected. Since d_{k^j} is rejected, then we have from Algorithm 2.3

$$\frac{C_{k^j} - \tilde{\phi}(x_{k^j+1};\bar{\rho})}{Pred_{k^j}} \leq \theta_1.$$

Using, inequalities (3.8) and (3.30), we have Hence

$$\begin{aligned}(1-\theta_1) <\mid \frac{C_{k^j} - \tilde{\phi}(x_{k^j+1};\bar{\rho})}{Pred_{k^j}} \mid \leq& \mid \frac{\tilde{\phi}(x_{k^j};\bar{\rho}) - \tilde{\phi}(x_{k^j+1};\bar{\rho})}{Pred_{k^j}} \mid \\ =& \frac{\mid Ared_{k^j} - Pred_{k^j} \mid}{Pred_{k^j}} \\ <& \frac{K_3\bar{\rho}\tau_{k^j}\|d_{k^j}\|^2}{K_4\tau_{k^j}\|d_{k^j}\|} \leq \frac{(1-\theta_1)}{2}.\end{aligned}$$

This demonstrates the lemma and provides a contradiction. □

3.5. Global convergence theory

The fundamental global convergence theorem for Algorithm 2.6 is covered in this section.

Theorem 3.1. *Under assumptions As_1–As_3, the sequence of iterates generated by the NTRAI algorithm satisfies*

$$\liminf_{k\to\infty}\ [\|Y_k\nabla f_k\| + \|Y_k\nabla P_k Z_k P_k\|\ ] \ = \ 0. \tag{3.32}$$

Proof. First, we will prove the following limit by contradiction

$$\liminf_{k\to\infty} \|Y_k\nabla\tilde{\phi}(x_k;\bar{\rho}_k)\| + \|Y_k\nabla P_k Z_k P_k\| = 0. \tag{3.33}$$

So, suppose that,

$$\|Y_k\nabla\tilde{\phi}(x_k;\bar{\rho}_k)\| + \|Y_k\nabla P_k Z_k P_k\| > \epsilon_1$$

for all k. Consider a trial step indexed j of the iteration indexed k such that $k^j \geq \bar{k}$ and $k \geq \bar{k}$. Using Algorithm 2.3 and Lemma 3.11, we have the following for any acceptable step indexed k^j,

$$C_{k^j} - \tilde{\phi}_{k^j+1} \geq \theta_1 Pred_{k^j} \geq \theta_1 K_4 \tau_{k^j} \Delta_{k^j}. \tag{3.34}$$

As k goes to infinity, we have

$$\lim_{k\to\infty} \Delta_{k^j} = 0. \tag{3.35}$$

This implies that the value of Δ_{k^j} is not bounded below:

If we consider an iteration with indexed $k^j > \bar{k}$ and if the preceding step was approved, that is, if $j = 1$, then $\Delta_{k^1} \geq \Delta_{min}$. Thus, in this case, Δ_{k^j} is bounded.

If $j > 1$, at least one trial step has been rejected, and according to Lemma 3.13, we have

$$\|d_{k^i}\| > \frac{(1-\theta_1)K_4}{2\bar{\rho}K_3},$$

for all $i = 1, 2, \cdots, j-1$. Since d_{k^i} is a rejected trial step, then from algorithm 2.3, we have

$$\Delta_{k^j} = \tilde{\alpha}_1 \|d_{k^{j-1}}\| > \tilde{\alpha}_1 \frac{(1-\theta_1)K_4}{2\bar{\rho}K_3}.$$

Hence, Δ_{k^j} is bounded and this contradicts condition (3.35). Hence, the supposition is wrong and the limit in (3.33) holds. Hence, limit in (3.32) holds and this completes the proof of the theorem. □

4. Numerical results

This section compares the performance of the NTRAI algorithm and demonstrates its robustness and efficiency using a collection of test problems with varying features that are commonly utilized in the literature. First, the tested problems are the Hock and Schittkowski's subset of the general nonlinear programming testing environment (the CUTE collection) [27]. Second, three engineering design problems are also tested.

We provided the numerical results of NTRAI algorithm obtained on a laptop with 8 GB RAM, USB 3 (10x), Nvidia GEFORCE GT, and Intel inside Core (TM)i7-2670QM CPU 2.2 GHz. NTRAI was run under MATLAB (R2013a)(8.2.0.701)64-bit(win64). The values of the required constants in Step 0 of nonmonotone trust-region Algorithm 2.5 were selected to be

$$\theta_1 = 0.25, \theta_2 = 0.75, \tilde{\alpha}_1 = 0.5, \tilde{\alpha}_2 = 2, \epsilon_1 = 10^{-10}, \text{ and } \epsilon_2 = 10^{-8}.$$

Successful termination with respect to the nonmonotone trust-region Algorithm 2.5 means that the termination condition of the algorithm is met with ϵ_1.

4.1. Benchmark test problems

Benchmark problems are listed in Hock and Schittkowski [27] to show the effectiveness of the NTRAI algorithm. For comparison, we have included the corresponding results of the NTRAI algorithm against the numerical results in [3, 28, 29]. This is summarized in Table 1, where Niter refers to the number of iterations. The algorithm has the ability to locate the optimal solution for either a feasible or infeasible initial reference point.

Table 1. Comparison between the methods in [3, 28, 29], and NTRAI algorithm.

Problem	name	Method [29]	Method [28]	Method [3]	NTRAI algorithm
		Niter	Niter	Niter	Niter
Prob1	hs006	7	5	4	4
Prob2	hs007	9	8	7	6
Prob3	hs008	14	6	8	6
Prob4	hs009	10	6	7	5
Prob5	hs012	5	7	4	4
Prob6	hs024	14	4	9	6
Prob7	hs026	19	19	14	12
Prob8	hs027	14	18	12	12
Prob9	hs028	6	2	3	2
Prob10	hs029	8	6	9	7
Prob11	hs030	7	6	8	4
Prob12	hs032	24	5	6	5
Prob13	hs033	29	6	8	5
Prob14	hs034	30	5	7	9
Prob15	hs036	10	7	9	6
Prob16	hs037	7	6	6	4
Prob17	hs039	19	23	5	7
Prob18	hs040	4	3	6	4
Prob19	hs042	6	3	7	5
Prob20	hs043	9	7	6	6
Prob21	hs046	25	10	10	8
Prob22	hs047	25	17	12	10
Prob23	hs048	6	2	3	3
Prob24	hs049	69	16	10	12
Prob25	hs050	11	8	6	5
Prob26	hs051	8	2	3	3
Prob27	hs052	4	2	3	2
Prob28	hs053	5	4	4	3
Prob29	hs056	12	5	4	3
Prob30	hs060	7	7	5	4
Prob31	hs061	44	7	8	6
Prob32	hs063	5	5	4	3
Prob33	hs073	16	7	8	6
Prob34	hs078	4	4	5	4
Prob35	hs079	7	4	4	4
Prob36	hs080	6	5	6	4
Prob37	hs081	9	6	7	5
Prob38	hs093	12	6	5	5

For all problems, these algorithms achieved the same optimal solution in [27]. Figure 1 shows the numerical results, which are summarized in Table 1 by using the performance profile that is proposed by Dolan and More [30].

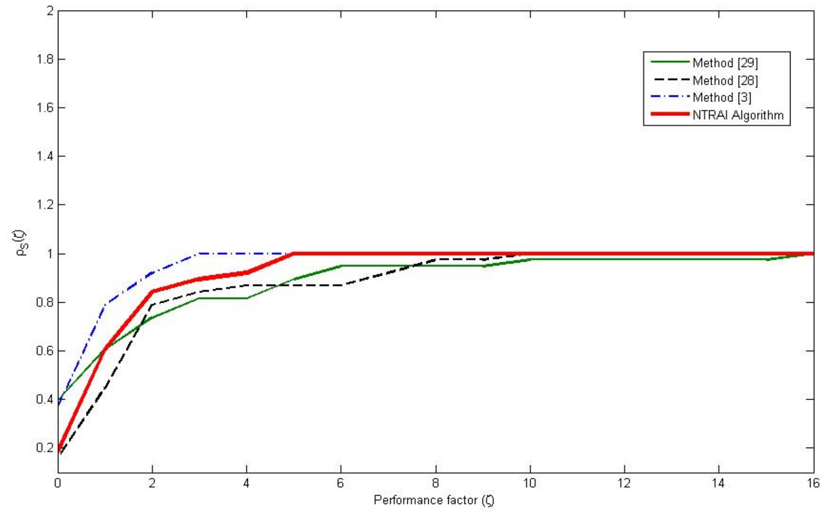

Figure 1. Performance profile based on the Niter of methods in [3, 28, 29], and NTRAI algorithm.

The performance profile in terms of Niter is given in Figure 1, which shows a distinctive difference between the NTRAI algorithm and the other algorithms [3, 28, 29]. Figure 2 represents the number of iterations required for each problem with different methods.

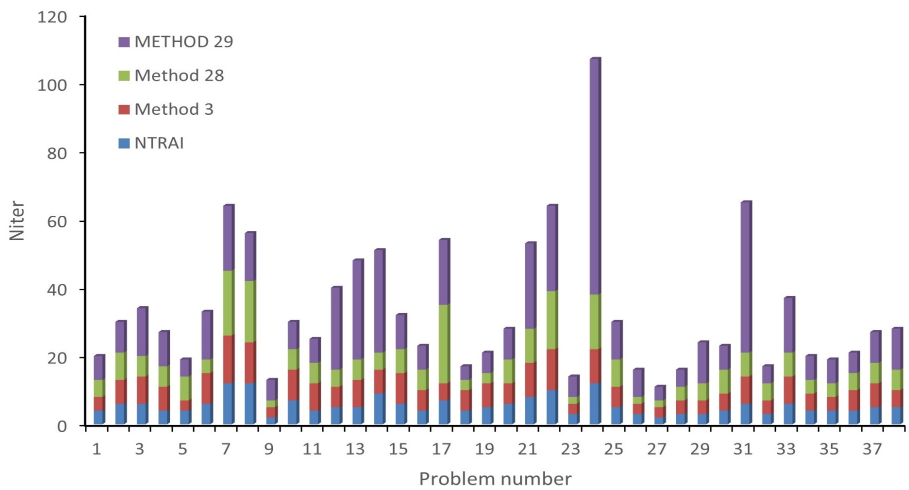

Figure 2. Comparison between the Niter of methods in [3, 28, 29] and the NTRAI algorithm.

4.2. Applicability of NTRAI algorithm to solve mechanical engineering problems

In this section, to evaluate the applicability of the NTRAI algorithm in real-world applications, we will consider three constrained mechanical engineering problems from the latest test suite [19].

In this experimental estimation, the NTRAI algorithm was compared with algorithms AOA [31], CGA [32], ChOA [33], SA [34], LMFO [35], I-MFO [36], MFO [37], WOA [38], GWO [39], SMFO [40], and WCMFO [41]. All algorithms attempt to solve three distinct problems, including: a gas transmission compressor design problem, a three-bar truss design problem, and a tension/compression spring design problem.

- P_1. **Gas transmission compressor design (GTCD) problem**

Minimizing the objective function utilizing four design variables is the fundamental objective of the GTCD problem. Figure 3 clarifies the GTCD problem. The mathematical formulation for the GTCD problem is

$$\begin{aligned}
&\textit{minimize} && 8.61\times 10^{5}\sqrt{\tfrac{x_1}{x_4}}x_2x_3^{\frac{-2}{3}}+3.69\times 10^{4}x_3+7.72\times 10^{8}\tfrac{x_2^{0.219}}{x_1}-765.43\times 10^{6}x_1^{-1},\\
&\textit{subject to} && \tfrac{x_4+1}{x_2^2}-1\le 0,\\
& && 20\le x_1\le 50,\\
& && 1\le x_2\le 10,\\
& && 20\le x_3\le 45,\\
& && 0.1\le x_4\le 60.
\end{aligned}$$

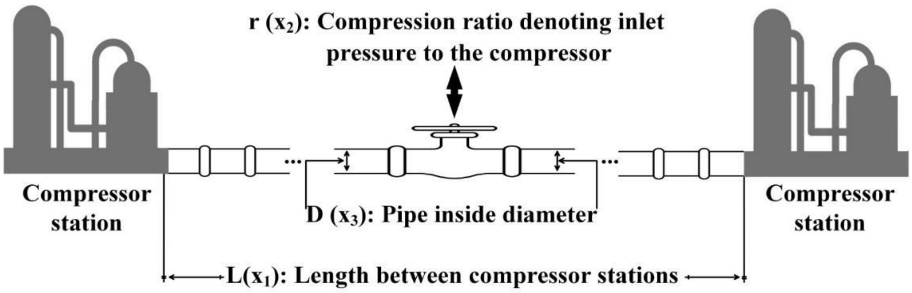

Figure 3. Gas transmission compressor design problem.

The performance of the NTRAI algorithm was evaluated against other methods when solving the GTCD problem. Table 2 presents the numerical results and comparisons for the GTCD problem. As seen in the table, the NTRAI algorithm is the most effective in addressing this problem.

Table 2. The numerical results and comparison for GTCD problem.

Name of algorithm	x_1	x_2	x_3	x_4	Optimal cost
SA [34]	46.76	1.62	25.79	0.55	4.390311×10^6
CGA [32]	49.97	20.01	31.47	49.83	1.735023×10^7
GWO [39]	20.00	7.81	20.00	60.00	2.964974×10^6
MFO [37]	50.00	1.18	24.57	0.39	2.964902×10^6
WOA [38]	50.00	1.18	24.86	0.39	2.965002×10^6
LMFO [35]	49.46	1.18	24.64	0.39	2.965456×10^6
WCMFO [41]	50.00	1.18	24.61	0.39	2.964897×10^6
ChOA [33]	50.00	1.19	24.24	0.41	2.966828×10^6
AOA [31]	50.00	1.23	20.00	0.51	3.014615×10^6
SMFO [40]	23.66	1.09	23.66	0.19	3.052254×10^6
I-MFO [36]	50.00	1.18	24.60	0.39	2.964896×10^6
NTRAI	49.6	1.175	24.9	0.382	$2.962714204361616\times 10^6$

- P_2. **Three-bar truss design (TBTD) problem**

In the TBTD problem, three constraints and two variables are used to formulate the weight of the bar structures, which is the objective function. Figure 4 clarifies the schematic for the TBTD problem. The mathematical formula for the TBTD problem is

$$\begin{aligned} \textit{minimize} \quad & 100(x_2 + 2\sqrt{2}x_1), \\ \textit{subject to} \quad & \frac{2x_2}{2x_1x_2 + \sqrt{2}x_1^2} - 2 \le 0, \\ & \frac{2x_2 + 2\sqrt{2}x_1}{2x_1x_2 + \sqrt{2}x_1^2} - 2 \le 0, \\ & \frac{2}{x_1 + \sqrt{2}x_2} - 2 \le 0, \\ & 0 \le x_1 \le 1, \\ & 0 \le x_2 \le 11. \end{aligned}$$

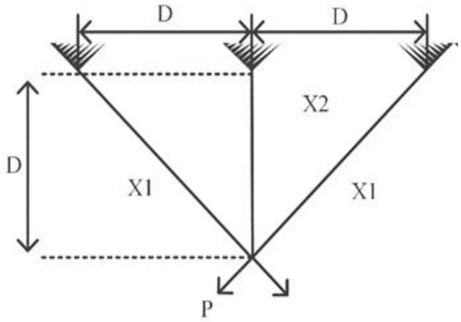

Figure 4. Three-bar truss design problem.

The NTRAI algorithm is compared with the other algorithms when solving the TBTD problem. Table 3 shows the numerical results for the TBTD problem. The NTRAI algorithm outperforms other algorithms in approximating the optimal values for variables with minimum weight.

Table 3. The numerical results and comparison for TBTD.

Name of algorithm	x_1	x_2	Optimal weight
SA [34]	0.768630	0.474232	2.6482456×10^2
CGA [32]	0.792428	0.397752	2.6390770×10^2
GWO [39]	0.787771	0.410862	2.6389619×10^2
MFO [37]	0.789186	0.406806	2.6389603×10^2
WOA [38]	0.787713	0.410977	2.6389653×10^2
LMFO [35]	0.791713	0.399909	2.6392114×10^2
WCMFO [41]	0.788472	0.408822	2.6389589×10^2
ChOA [33]	0.787802	0.410724	2.6389653×10^2
AOA [31]	0.792789	0.396906	2.6392526×10^2
SMFO [40]	0.792044	0.398859	2.6390973×10^2
I-MFO [36]	0.788792	0.407919	2.6389585×10^2
NTRAI	0.7	0.4	2.3798989873×10^2

- P_3. **Tension/compression spring design (TCSD) problem**

In the TCSD problem, four constraints and three variables are utilized to formulate the weight of the tension/compression spring, which is an objective function. As shown in Figure 5, the variables are wire diameter **d**, the mean coil diameter **D**, and the number of active coils **N**. These variables are denoted in mathematical formulation by x_1, x_2, and x_3, respectively.

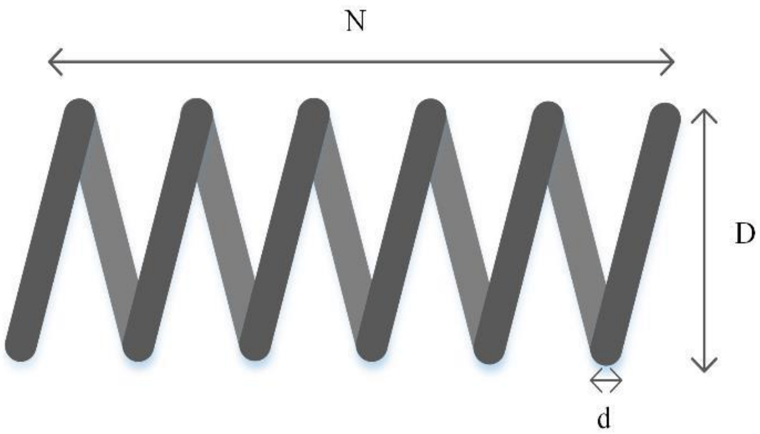

Figure 5. Tension/compression spring design problem.

The mathematical formulation for the TCSD problem is

$$\begin{aligned}
\textit{minimize} \quad & x_1^2 x_2 (2 + x_3), \\
\textit{subject to} \quad & 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0, \\
& \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \le 0, \\
& 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0, \\
& \frac{2}{3}(x_1 + x_2) - 1 \le 0, \\
& 0.05 \le x_1 \le 2, \\
& 0.25 \le x_2 \le 1.3, \\
& 2 \le x_3 \le 15.
\end{aligned}$$

The NTRAI Algorithm 2.6 is compared with other algorithms when solving the tension/compression spring design problem. The numerical results and the comparison between algorithms for the TCSD problem are shown in Table 4. The NTRAI algorithm is better than other algorithms in approximating the optimal values for variables with minimum weight.

Table 4. The numerical results and comparison for the TCSD problem.

Name of algorithm	$d = x_1$	$D = x_2$	$N = x_3$	Optimum weight
SA [34]	0.075935	0.993094	3.879891	0.033670
CGA [32]	0.071031	1.019975	1.726076	0.019749
GWO [39]	0.051231	0.345699	11.970135	0.012676
MFO [37]	0.053064	0.390718	9.542437	0.012699
WOA [38]	0.050451	0.327675	13.219341	0.012694
LMFO [35]	0.050000	0.317154	14.107156	0.012771
WCMFO [41]	0.051509	0.352411	11.545969	0.012666
ChOA [33]	0.051069	0.341746	12.251078	0.012702
AOA [31]	0.050000	0.310475	15.000000	0.013195
SMFO [40]	0.050000	0.314692	14.696505	0.013136
I-MFO [36]	0.051710	0.357217	11.259785	0.012665
NTRAI	0.05179848439	0.35946589	11.12481959619885	0.0126585842553172

- **Example. Nonconvex optimization problem [20]**

Consider the following nonconvex nonlinear constrained optimization problem

$$\begin{aligned} \textit{minimize} \quad & -x_1 - x_2, \\ \textit{subject to} \quad & x_1 x_2 \leq 4 \\ & 0 \leq x_1 \leq 6, \\ & 0 \leq x_2 \leq 4. \end{aligned}$$

The above problem possesses two strong local minima points $(1, 4)$ and $(6, 0.66667)$. Applying the NTRAI Algorithm 2.6 on this nonconvex problem, we have the local points $(1.0000001220725, 4)$ are obtained and the value of objective function is -5.0000001220725.

5. Conclusions

This research focused on combining a nonmonotone technique with an autonomously modified trust-region radius to provide a more efficient hybrid of trust-region approaches for constrained optimization problems. The active-set strategy was combined with a penalty and Newton's interior point method to transform a nonlinearly constrained optimization problem into an identical unconstrained one. A nonmonotone trust region was used to ensure convergence from any starting point to the stationary point. A global convergence theory for the suggested method was developed based on certain assumptions. Well-known test problems (the CUTE collection) were used to evaluate the suggested method; three engineering design problems were performed, and the outcomes were compered with those of other reputable optimizers. The results showed that, compared with the other algorithms under discussion, the proposed method typically yields better approximation solutions and requires fewer iterations. Computational findings, which also examined the algorithm's performance, demonstrated the suggested algorithm's competitiveness and superiority over alternative optimization algorithms.

Several questions should be answered in future research:

- Improving the nonmonotone trust-region algorithm to be able to handle nondifferentiation-constrained optimization problems.

- Improving the nonmonotone trust-region algorithm to be able to handle large-scale constrained optimization problems.
- Utilize a secant approximation of the Hessian matrix to output a more effective algorithm.

Author contributions

Bothina Elsobky: conceived the study, developed the theoretical framework and performed the numerical experiments; Yousria Abo-Elnaga: conceived the study, supervised the application; Gehan Ashry: aided in the analysis. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors are very grateful to the anonymous reviewers for their valuable and insightful comments, which have aided us in improving the quality of this paper.

Conflict of interest

The authors declare no conflicts of interest.

References

1. I. Das, *An interior point algorithm for the general nonlinear programming problem with trust region globlization*, Institute for Computer Applications in Science and Engineering, 1996. https://doi.org/10.5555/870136
2. B. El-Sobky, An active-set interior-point trust-region algorithm, *Pacific J. Optim.*, **14** (2018), 125–159.
3. B. El-Sobky, Y. Abo-Elnaga, A. Mousa, A. El-Shorbagy, Trust-region based penalty barrier algorithm for constrained nonlinear programming problems: an application of design of minimum cost canal sections, *Mathematics*, **9** (2021), 1551. https://doi.org/10.3390/math9131551
4. B. El-Sobky, G. Ashry, An interior-point trust-region algorithm to solve a nonlinear bilevel programming problem, *AIMS Math.*, **7** (2022), 5534–5562. https://doi.org/10.3934/math.2022307
5. B. El-Sobky, G. Ashry, Y. Abo-Elnaga, An active-set with barrier method and trust-region mechanism to solve a nonlinear Bilevel programming problem, *AIMS Math.*, **7** (2022), 16112–16146. https://doi.org/10.3934/math.2022882
6. B. El-Sobky, A multiplier active trust-region algorithm for solving general nonlinear programming problem, *Appl. Math. Comput.*, **219** (2012), 928–946.

7. B. El-Sobky, G. Ashry, An active-set Fischer-Burmeister trust-region algorithm to solve a nonlinear bilevel optimization problem, *Fractal Fract.*, **6** (2022), 412–441. https://doi.org/10.3390/fractalfract6080412

8. B. El-Sobky, M. F. Zidan, A trust-region based an active-set interior-point algorithm for fuzzy continuous static games, *AIMS Math.*, **8** (2023), 13706–13724. https://doi.org/10.3934/math.2023696

9. B. El-Sobky, Y. Abo-Elnaga, G. Ashry, M. Zidan, A nonmonotone active interior point trust region algorithm based on CHKS smoothing function for solving nonlinear bilevel programming problems, *AIMS Math.*, **9** (2024), 6528–6554. https://doi.org/10.3934/math.2024318

10. A. Sartenaer, Automatic determination of an initial trust region in nonlinear programming, *SIAM J. Sci. Comput.*, **18** (1997), 1788–1803. https://doi.org/10.1137/S1064827595286955

11. X. S. Zhang, J. L. Zhang, L. Z. Liao, An adaptive trust region method and its convergence, *Sci. China Ser. A*, **45** (2002), 620–631. https://doi.org/10.1360/02ys9067

12. Z. J. Shiand, J. H. Guo, A new trust region methods for unconstrained optimization, *J. Comput. Appl. Math.*, **213** (2008), 509–520. https://doi.org/10.1016/j.cam.2007.01.027

13. L. Grippo, F. Lampariello, S. Lucidi, A truncated Newton method with nonmonotone line search for unconstrained optimization, *J. Optim. Theory Appl.*, **60** (1989), 401–419. https://doi.org/10.1007/BF00940345

14. P. L. Toint, An assessment of nonmonotone linesearch technique for unconstrained optimization, *SIAM J. Sci. Comput.*, **17** (1996), 725–739. https://doi.org/10.1137/S106482759427021X

15. N. Y. Deng, Y. Xiao, F. J. Zhou, Nonmonotonic trust region algorithm, *J. Optim. Theory Appl.*, **76** (1993), 259–285. https://doi.org/10.1007/BF00939608

16. J. L. Zhang, X. S. Zhang, A nonmonotone adaptive trust region method and its convergence, *Comput. Math. Appl.*, **45** (2003), 1469–1477. https://doi.org/10.1016/S0898-1221(03)00130-5

17. H. C. Zhang, W. W. Hager, A nonmonotone line search technique for unconstrained optimization, *SIAM J. Optim.*, **14** (2004), 1043–1056.

18. J. Mo, C. Liu, S. Yan, A nonmonotone trust region method based on nonincreasing technique of weighted average of the successive function value, *J. Comput. Appl. Math.*, **209** (2007), 97–108. https://doi.org/10.1016/j.cam.2006.10.070

19. A. Kumar, G. Wu, M. Ali, R. Mallipeddi, P. Suganthan, S. Das, A test-suite of non-convex constrained optimization problems from the real-world and some baseline results, *Swarm Evol. Comput.*, **56** (2020), 100693. https://doi.org/10.1016/j.swevo.2020.100693

20. N. Sahinidis, I. E. Grossmann Convergence properties of generalized benders decomposition, *Comput. Chem. Eng.*, **15** (1991), 481–491. https://doi.org/10.1016/0098-1354(91)85027-R

21. J. Dennis, M. El-Alem, K. Williamson, A trust-region approach to nonlinear systems of equalities and inequalities, *SIAM J. Optim.*, **9** (1999), 291–315. https://doi.org/10.1137/S1052623494276208

22. R. Fletcher, An l_1 penalty method for nonlinear constraints, *Numer. Optim.*, 1984, 26–40.

23. J. Dennis, R. Schnabel, *Numerica methods for unconstrained optimization and nonlinear equations*, Prentice-Hall, 1983. https://doi.org/10.1137/1.9781611971200

24. Y. Yuan, On the convergence of a new trust region algorithm, *Numer. Math.*, **70** (1995), 515–539. https://doi.org/10.1007/s002110050133

25. O. Mangasarian, *Nonlinear programming*, McGraw-Hill Book Company, 1969.

26. B. El-Sobky, Y. Abouel-Naga, A penalty method with trust-region mechanism for nonlinear bilevel optimization problem, *J. Comput. Appl. Math.*, **340** (2018), 360–374. https://doi.org/10.1016/j.cam.2018.03.004

27. W. Hock, K. Schittkowski, *Test examples for nonlinear programming codes*, Springer-Verlag, 1981. https://doi.org/10.1007/978-3-642-48320-2

28. J. Nocedal, F. Oztoprak, R. Waltz, An interior point method for nonlinear programming with infeasibility detection capabilities, *Optim. Methods Software*, **29** (2014), 837–854. https://doi.org/10.1080/10556788.2013.858156

29. A. L. Tits, A. Wachter, S. Bakhtiari, T. J. Urban, G. T. Lawrence, A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties, *SIAM J. Optim.*, **14** (2003), 173–199. https://doi.org/10.1137/S1052623401392123

30. E. Dolan, J. More, Benchmarking optimization software with performance profiles, *Math. Programm.*, **91** (2002), 201–213. https://doi.org/10.1007/s101070100263

31. L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609. https://doi.org/10.1016/j.cma.2020.113609

32. R. Chelouah, P. Siarry, A continuous genetic algorithm designed for the global optimization of multimodal functions, *J. Heuristics*, **6** (2000), 191–213. https://doi.org/10.1023/A:1009626110229

33. M. Khishe, M. R. Mosavi, Chimp optimization algorithm, *Expert Syst. Appl.*, **149** (2020), 113338. https://doi.org/10.1016/j.eswa.2020.113338

34. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, *Science*, **220** (1983), 671–680. https://doi.org/10.1126/science.220.4598.671

35. Z. Li, Y. Zhou, S. Zhang, J. Song, Levy-flight moth-flame algorithm for function optimization and engineering design problems, *Math. Probl. Eng.*, **126** (2016), 1423930. https://doi.org/10.1155/2016/1423930

36. M. H. Nadimi-Shahraki, A. Fatahi, H. Zamani, S. Mirjalili, L. Abualigah, An improved moth-flame optimization algorithm with adaptation mechanism to solve numerical and mechanical engineering problems, *Entropy*, **23** (2021), 1637. https://doi.org/10.3390/e23121637

37. S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, *Knowl. Based Syst.*, **89** (2015), 228–249. https://doi.org/10.1016/j.knosys.2015.07.006

38. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

39. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

40. C. Chen, X. Wang, H. Yu, M. Wang, H. Chen, Dealing with multi-modality using synthesis of Moth-flame optimizer with sine cosine mechanisms, *Math. Comput. Simul.*, **188** (2021), 291–318. https://doi.org/10.1016/j.matcom.2021.04.006

41. S. Khalilpourazari, S. Khalilpourazary, An efficient hybrid algorithm based on water cycle and moth-flame optimization algorithms for solving numerical and constrained engineering optimization problems, *Soft Comput.*, **23** (2019), 1699–1722. https://doi.org/10.1007/s00500-017-2894-y