A Nonmonotone Adaptive Trust Region Penalty Method for Multiobjective Nonlinear Bilevel Problem

Xianfeng Ding, Yiyuan Wang, Tingting Wei and Jiaxin Li

***Abstract*—In this paper, we study a trust region penalty method for multiobjective nonlinear bilevel optimization (MNBLO). Firstly, the MNBLO is transformed into unconstrained programming according to multiobjective programming, KKT optimal condition, penalty function method, and effective set strategy. Secondly, a trial step evaluation criterion and a radius update method are applied to the proposed penalty algorithm. Next, the convergence of the algorithm is explained by some related assumptions and theorems. Finally, we illustrate the feasibility of the algorithm with numerical examples.**

***Index Terms*—Adaptive trust region radius, MNBLO, Nonmonotone technology, Penalty function method.**

I. Introduction

THE MNBLO problem is a nested optimization problem with the first and second level structures. Each of these has corresponding decision variables, constraints, and multiple conflicting objectives, and it has nonlinear expression in it. The second level decision-makers have to optimize their own goals under the parameters given by the first level decision-makers. The first level decision-makers select the corresponding parameters to optimize their goals according to the decisions made by the second level decision-makers.

In this article, a penalty method based on an adaptive trust region mechanism is studied. The "optimistic mode" MNBLO is as follows:

$$\begin{aligned} &\min_{x,y} (F_1(x,y), F_2(x,y), \ldots, F_s(x,y))^T \\ &s.t. G(x,y) \leq 0, \\ &\quad \text{where } y \text{ solves} \\ &\quad \min_{y} (f_1(x,y), f_2(x,y), \ldots, f_t(x,y))^T \\ &\quad s.t. g(x,y) \leq 0, \end{aligned} \tag{1}$$

where $x \in R^{n_1}, y \in R^{\eta_2}$, x and y are decision variables. $F_i(x,y)$ and $f_j(x,y)$ denote decision objectives. $G(x,y)$ and $g(x,y)$ are both constraints. $i = 1,2,\ldots,s, j = 1,2,\ldots,t$.

For multiobjective programming, it is common to choose a solution among effective solutions based on the propensity of decision-makers. The basic idea of dealing with multiobjective problem is to transform it into a single objective programming by using evaluation functions. Common methods are linear weighting, ideal points, weighted quadratic summation, and minimax. In addition, it is noted that the global criterion method (GCM) [1] is the most widely used scaling method in solving the multiobjective optimization problem. The GCM is a compromise method that minimizes the deviation and minimization of the objective function from its ideal points.

Next, different approaches are considered to deal with the two-level objective functions of problem (1), respectively. For the first level multiobjective problem, the global criterion method is combined with L_p -criterion. The linear weighted method is used for the second level multiobjective problem. These two methods convert the two-level objective functions into the following single objectives, respectively:

$$F(x,y) = \left\{ \sum_{k=1}^{s} \left(\frac{F_k(x,y) - F_k^{\min}}{F_k^{\max} - F_k^{\min}} \right)^p \right\}^{\frac{1}{p}} \tag{2}$$

$$f(x,y) = \sum_{k=1}^{t} \beta_k f_k(x,y) \tag{3}$$

where $1 \leq p < \infty$, $\sum\limits_{k=1}^{t} \beta_k = 1$. $F_k^{\max}$ and $F_k^{\min}$ represent the best and worst optimal solutions of the kth objective function under both constraints, respectively. The most commonly used approach in upper level multiobjective processing is to take $p = 2$, which is known as the global criterion method in L_2 norm. Then transform problem (1) into a single-objective nonlinear bilevel programming (NBLP):

$$\begin{aligned} &\min_{x,y} F(x,y) \\ &s.t. G(x,y) \leq 0 \\ &\quad \text{where } y \text{ solves} \\ &\quad \min_{y} f(x,y) \\ &\quad s.t. g(x,y) \leq 0 \end{aligned} \tag{4}$$

where $x \in R^{n_1}, y \in R^{n_2}$, functions $F : R^{n_1+n_2} \to R, f : R^{n_1+n_2} \to R, G : R^{n_1+n_2} \to R^{m_1}$, $g : R^{n_1+n_2} \to R^{m_2}$.

In general, the solution of single objective NBLP refers to [2]. We consider replacing the lower problem in problem (4) with its KKT optimality condition [3], and obtain NP

Manuscript received August 3, 2022; revised January 29, 2023.

Xianfeng Ding is an associate professor in the School of Science, Southwest Petroleum University, Chengdu 610500, PR China (e-mail: fxxd@163.com).

Yiyuan Wang is a postgraduate student in the School of Science, Southwest Petroleum University, Chengdu 610500, PR China (e-mail: wangyiyuan1115@163.com).

Tingting Wei is a postgraduate student in the School of Science, Southwest Petroleum University, Chengdu 610500, PR China (e-mail: 1558862556@qq.com).

Jiaxin Li is a postgraduate student in the School of Science, Southwest Petroleum University, Chengdu 610500, PR China (e-mail: 1195374462@qq.com).

problem:

$$\begin{aligned}&\min_{x,y} F(x,y)\\&\text{s.t } G(x,y)\leq 0,\\&\quad \nabla_y f(x,y)+\nabla_y g(x,y)\lambda=0,\\&\quad g(x,y)\leq 0,\\&\quad \lambda_j g_j(x,y)=0,\quad j=1,\ldots,m_2,\\&\quad \lambda_j\geq 0,\quad j=1,\ldots,m_2.\end{aligned}\tag{5}$$

where $\lambda \in R^{m_2}$ is the Lagrange vector corresponding to inequality constraint $g(x,y)$. $\lambda_j g_j(x,y)=0$ is called the complementary relaxation condition, see [4] for details. If (y^*,λ^*) satisfies the KKT optimal condition of the second problem at x^*, then (x^*,y^*,λ^*) is the optimal solution of problem (5). Next, we express problem (5) as follows:

$$\begin{aligned}&\min_{\bar{x}} F(\bar{x})\\&\text{s.t } C_e(\bar{x})=0,\quad e\in E,\\&\quad C_i(\bar{x})\leq 0,\quad i\in I,\end{aligned}\tag{6}$$

where $\bar{x}=(x,y,\lambda)^T$, $E=\{1,\ldots,n_2+m_2\}$ and $I=\{1,\ldots,m_1+2m_2\}$, $E\cap I=\varnothing$. Assume that the second derivatives of $F(\bar{x})$, $C_e(\bar{x})$, and $C_i(\bar{x})$ are all continuous.

Inspired by the active set method in reference [5], we consider defining a matrix $Z(\bar{x})$, whose diagonal elements are:

$$z_i(\bar{x})=\begin{cases}1 & i\in E,\\ 1 & C_i(\bar{x})\geq 0, i\in I,\\ 0 & C_i(\bar{x})<0, i\in I,\end{cases}\tag{7}$$

Using this matrix, inequality constraints in problem (6) become equality constraints, and the equality constraint programming (ECP) is obtained:

$$\begin{aligned}&\text{minimize} && F(\bar{x})\\&\text{s.t.} && D(\bar{x})^T Z(\bar{x}) D(\bar{x})=0\end{aligned}\tag{8}$$

where the component of $D(\bar{x})$ is $C_l(\bar{x})$, and $l\in E\cup I$. See reference [5] for more details.

For the ECP problem (8), the penalty function method [6] can be applied to transform it into an unconstrained nonlinear programming (UNP):

$$\begin{aligned}&minimize && \hat{F}(\bar{x})=F(\bar{x})+\tfrac{\rho}{2}\|Z(\bar{x})D(\bar{x})\|^2\\&\text{s.t.} && \bar{x}\in R^{n_1+n_2+m_2},\end{aligned}\tag{9}$$

where $\rho\in R$ is a penalty parameter greater than 0. So far, the MNBLO problem (1) has been transformed into unconstrained optimization problem (9). Now this paper mainly focuses on problem analysis and algorithm discussion for problem (9). In the algorithm in this paper, the minimization problem (9) satisfying the first-order necessary condition is equivalent to problem (6).

In his article [7] in 1970, Powell first proposed the trust region method with strong global convergence to solve the unconstrained programming problem. However, the traditional trust region approach [8] requires some criteria to judge the trial step.

The quadratic trust region subproblem of question (9) is as follows:

$$\begin{aligned}&\text{minimize } \Phi_k(d_k)=F_k+\nabla F_k^T d\\&\quad +\tfrac{1}{2}d^T H_k d+\tfrac{\rho_k}{2}\left\|Z_k\left(D_k+\nabla D_k^T d\right)\right\|^2\\&s.t. \qquad \|d\|\leq\Delta_k,\end{aligned}\tag{10}$$

where trust region radius $\Delta_k>0$. H_k are the Hessian matrix of $F(\bar{x}_k)$ or its approximation. In order to test whether the trial step d_k obtained from the above problem (10) is accepted. Define a ratio:

$$r_k=\frac{Ared_k}{Pred_k}\tag{11}$$

where $Ared_k$ the actual reduction of the function value of problem (9). $Pred_k$ represents the predicted reduction of the subproblem (10). Their definitions are as follows:

$$\begin{aligned}Ared_k&=\hat{F}(\bar{x}_k)-\hat{F}(\bar{x}_k+d_k)\\&=F(\bar{x}_k)-F(\bar{x}_{k+1})\\&\quad+\frac{\rho_k}{2}\left[\|Z_kD_k\|^2-\|Z_{k+1}D_{k+1}\|^2\right],\end{aligned}\tag{12}$$

$$\begin{aligned}Pred_k&=\Phi_k(0)-\Phi_k(d_k)\\&=-\nabla F_k^T d_k-\frac{1}{2}d_k^T H_k d_k\\&\quad+\frac{\rho_k}{2}\left[\|Z_kD_k\|^2-\left\|Z_k\left(D_k+\nabla D_k^T d_k\right)\right\|^2\right],\\&=-\left(\nabla F_k+\rho_k\nabla P_k Z_k P_k\right)^T d_k-\frac{1}{2}d_k^T B_k d_k\end{aligned}\tag{13}$$

where $B_k=H_k+\rho_k\nabla D_k Z_k\nabla D_k^T$. Since $\Phi_k(d_k)$ is the minimum value at the current iteration point $\overline{\mathcal{X}}_k$, $Pred_k$ is nonnegative. Therefore, when r_k is less than or close to 0, d_k is rejected, and Δ_k should be reduced. When r_k is close to 1, d_k is accepted and Δ_k should be expanded.

For convenience, we will simplify as follows: $F_k=F(\bar{x}_k), D_k=D(\bar{x}_k), \nabla F_k=\nabla F(\bar{x}_k), \nabla D_k=\nabla D(\bar{x}_k), Z_k=Z(\bar{x}_k)$. $\{k_i\}$ represents the ith trial step in the kth iteration, which is expressed as concisely. All norms take the l_2 norm.

The paper is structured as follows. In Section 2, the algorithm based on nonmonotone technique and adaptive radius is proposed. In Section 3, we discuss the convergence in two cases. In Section 4, the algorithm is verified with numerical examples. Finally, the work done is summarized.

II. Trust region algorithm

A. Evaluation criterion of trial step based on nonmonotone technique

The three key factors that affect the numerical performance of trust region algorithm are initial radius, evaluation criterion of trial step and radius updating strategy. We first consider the unconstrained programming problem under general conditions, and then apply the proposed trial step evaluation criterion r_k to the trust region algorithm of MNBLO.

For the general unconstrained programming problem $\min_{x\in R^n} f(x)$, let $f_k=f(x_k)$, $g_k=\nabla f(x_k)$, and $B_k\in R^{n\times n}$ be $\nabla^2 f(x_k)$ or its approximate Hessian matrix. The quadratic subproblem is $\min_{d\in R^n} m_k(d)=f_k+g_k^T d+\frac{1}{2}d^T B_k d$, where $\|d\|\leq\Delta_k$. Deng et al. [9] applied the nonmonotone technique to the trust region algorithm and modified the ratio. The nonmonotone term $f_{l(k)}$ is expressed as:

$$f_{l(k)}=f\left(x_{l(k)}\right)=\max_{0\leq j\leq m(k)}\{f_{k-j}\}, k=0,1,2,\ldots,\tag{14}$$

where $m(0) = 0, 0 \leq m(k) \leq \min\{N_1, m(k-1)+1\}$, and N_1 is a nonnegative integer. The reference function $f_{l(k)}$ is monotone nonincreasing, and equation (14) is called the maximum nonmonotone rule. After r_k is modified by nonmonotone technology, the general nonmonotone ratio can be obtained as follows:

$$\tilde{r}_k = \frac{f_{l(k)} - f(x_k + d_k)}{m_k(0) - m_k(d_k)}. \quad (15)$$

However, the numerical performance of nonmonotone technique depends heavily on the select of constant N. In 2012, Ahookhosh and Amini [10] proposed a more efficient nonmonotone technology than formula (14) to get rid of the dependence on N. By convex combination, the nonmonotone ratio can be obtained:

$$\hat{r}_k = \frac{R_k - f(x_k + d_k)}{m_k(0) - m_k(d_k)}, \quad (16)$$

where $R_k = \eta_k f_{l(k)} + (1-\eta_k) f(x_k)$, $\eta_k \in [\eta_{\min}, \eta_{\max}]$, $\eta_{\min} \in [0,1), \eta_{\max} \in [\eta_{\min}, 1]$.

$\hat{r}_k$ is applied to solve the MNBLO problem, and the nonmonotone ratio is obtained:

$$r_k = \frac{R_k - \hat{F}(x_k + d_k)}{\Phi_k(0) - \Phi_k(d_k)}, \quad (17)$$

where $R_k = \eta_k \hat{F}_{l(k)} + (1-\eta_k)\hat{F}(x_k)$, $\eta_k \in [\eta_{\min}, \eta_{\max}]$, $\eta_{\min} \in [0,1), \eta_{\max} \in [\eta_{\min}, 1]$. $\hat{F}_{l(k)} = \hat{F}\left(x_{l(k)}\right) = \max\limits_{0 \leq j \leq \Phi(k)} \left\{\hat{F}_{k-j}\right\}, k = 0, 1, 2, \ldots$, $\Phi(0) = 0, 0 \leq \Phi(k) \leq \min\{N_1, \Phi(k-1)+1\}$.

B. Adaptive trust region radius

Numerous scholars consider the adaptive trust region radius to reduce computation. In 1997, Sartenear [11] proposed a method to automatically determine the initial radius by using the initial gradient information. But this approach does not reduce the amount of computation. In 2000, Zhang [12] effectively used the gradient and second derivative information to propose an adaptive algorithm, where $\Delta_k = c^p \frac{\|g_k\|}{a_k}$, p is a positive integer, $0 < c < 1$, $a_k = \max\{\|B_k\|, 1\}$. In 2002, in the algorithm proposed by Zhang et al. [13], $\Delta_k = c^p \|g_k\| \left\|\hat{B}_k^{-1}\right\|$, where $\hat{B}_k = B_k + iI$, i is an integer. However, his method requires an estimate of matrices B_k and $\hat{B}_k^{-1}$ in each iteration. In 2006, Li [14] proposed another effective adaptive trust region algorithm, and let $\Delta_k = \frac{\|d_{k-1}\|}{\|y_{k-1}\|}\|g_k\|$, $y_{k-1} = g_k - g_{k-1}$. This method does not need to compute matrices and contains gradient information.

Inspired by these methods, the following radius update strategy is adopted:

$$\Delta_k = c_k^p \max\left\{1, \frac{\|d_{k-1}\|}{\|y_{k-1}\|}\right\}\|g_k\|, \quad (18)$$

where g_k represents the first order derivative of problem (9) at $\bar{x}$, $g_k = \nabla F_k + \rho_k \nabla D_k Z_k D_k, y_{k-1} = g_k - g_{k-1}$. p is an integer, $0 < c_k < 1, c_k^p$ is an adjustment parameter, and the adjustment of Δ_{k+1} depends on c_k^p, which is defined as:

$$c_{k+1}^p = \begin{cases} \min\left\{\sigma_2 c_k^p, \left(c_k^p\right)_{\max}\right\} & \bar{r}_k \geq \mu_2 \\ c_k^p & \mu_1 \leq \bar{r}_k < \mu_2 \\ \sigma_1 c_k^p & \bar{r}_k < \mu_1 \end{cases}, \quad (19)$$

where $0 < \sigma_1 < 1 < \sigma_2, 0 < \mu_1 < \mu_2 < 1, \left(c_k^p\right)_{\max}$ is the maximum value of c_k^p in k cycles.

C. A nonmonotone adaptive trust region algorithm

Known methods for solving trust region subproblems mainly include the broken line method [15], preconditioned conjugate gradient method [16] and mixed broken line method [17]. We adopt the preconditioned conjugate gradient method to solve d_k.

Algorithm 1. Conjugate gradient method to solve trial step d_k

---

Step 1. Given $\varepsilon_0 > 0$, let $d_0 = 0 \in R^{n_1+n_2+m_2}, w_0 = -(\nabla F_k + \rho_k \nabla D_k Z_k D_k), v_0 = w_0, i := 1, i = 1, 2, \ldots, n_1 + n_2 + m_2$.
Step 2. Calculate $B_k = H_k + \rho_k \nabla D_k Z_k \nabla D_k^T$, $\alpha_i = \frac{w_i^T w_i}{v_i^T B_k v_i}$, calculate γ_i such that $\|d_i + \gamma_i v_i\| = \Delta_k$, if $v_i^T B_k v_i \leq 0$, let $d_k = d_i + \gamma_i v_i$, and terminate the algorithm. Otherwise, let $d_{i+1} = d_i + \alpha_i v_i$, $w_{i+1} = w_i - \alpha_i B_k v_i$, turn to Step 3 .
Step 3. If $\frac{w_{i+1}}{w_0} \leq \varepsilon_0$, let $d_k = d_{i+1}$, and terminate the algorithm. Otherwise, turn to Step 4.
Step 4. Calculate $\theta_i = \frac{w_{i+1}^T w_{i+1}}{w_i^T w_i}$, and a new direction $v_{i+1} = w_{i+1} + \theta_i v_i$, let $i := i+1$, turn to Step 2.

Subsequently, we apply the nonmonotone ratio of formula (17) and the adaptive trust region update strategy of formula (18) and formula (19) to evaluate d_k. See Algorithm 2.

Algorithm 2. Adaptive trust region radius update algorithm

Step 1. Given $0 < \mu_1 < \mu_2 < 1$, $0 < \sigma_1 < 1 < \sigma_2$, $\eta_k \in [\eta_{\min}, \eta_{\max}]$, $\eta_{\min} \in [0,1)$, $\eta_{\max} \in [\eta_{\min}, 1]$, $p := 0$, $0 < c_k < 1$. Let $r_k = \frac{R_k - \hat{F}(x_k + d_k)}{\Phi_k(0) - \Phi_k(d_k)}$. If $\bar{r}_k < \mu_1$, let $\Delta_k = \sigma_1 c_k^p \max\left\{1, \frac{\|d_{k-1}\|}{\|y_{k-1}\|}\right\}\|g_k\|$, $p := p+1$, recalculate d_k.
Step 2. If $\mu_1 \leq \bar{r}_k < \mu_2$, let $\bar{x}_{k+1} = \bar{x}_k + d_k, \Delta_{k+1} = c_k^p \max\left\{1, \frac{\|d_{k-1}\|}{\|y_{k-1}\|}\right\}\|g_k\|$.
Step 3. If $\bar{r}_k \geq \mu_2$, let $\bar{x}_{k+1} = \bar{x}_k + d_k, \Delta_{k+1} = \min\left\{\sigma_2 c_k^p, \left(c_k^p\right)_{\max}\right\} \max\left\{1, \frac{\|d_{k-1}\|}{\|y_{k-1}\|}\right\}\|g_k\|$.

For positive penalty parameter ρ_k, we refer to the method [18] to update ρ_k, see Algorithm 3.

Algorithm 3. Update penalty parameter

Step 1. Calculate $Pred_k = \Phi_k(0) - \Phi_k(d_k)$, if $Pred_k \geq \|\nabla D_k Z_k D_k\| \min\{\|\nabla D_k Z_k D_k\|, \Delta_k\}$, let $\rho_{k+1} = \rho_k$. Otherwise, turn to Step 2 .
Step 2. Let $\rho_{k+1} = 2\rho_k$.

Finally, for $\epsilon_1, \epsilon_2 > 0$, when $\|\nabla F_k\| + \|\nabla D_k Z_k D_k\| \leq \epsilon_1$ or $\|d_k\| \leq \epsilon_2$, the algorithm terminates. See Algorithm 4 for our proposed algorithm.

Algorithm 4. Nonmonotone adaptive trust region algorithm

Step 0. Given an initial point $\bar{x}_0 \in R^{n+n_2+m_1}$, select parameters $\epsilon_1 > 0, \epsilon_2 > 0,\ 0 < \mu_1 < \mu_2 < 1,\quad 0 < \sigma_1 < 1 < \sigma_2,\quad \eta_k \in [\eta_{\min}, \eta_{\max}],\quad \eta_{\min} \in [0,1), \eta_{\max} \in [\eta_{\min}, 1]$, let $\rho_0 = 1, k = 0,\quad 0 < c_k < 1,\quad p := 0, \Delta_0 = \|\nabla F_0\| + \|\nabla D_0 Z_0 D_0\|$
Step 1. If $\|\nabla F_k\| + \|\nabla D_k Z_k D_k\| \leq \epsilon_1$, stop.
Step 2. Use Algorithm 1 to calculate trial step d_k.
Step 3. If $\|d_k\| \leq \epsilon_2$, stop.
Step 4. Let $\bar{x}_{k+1} = \bar{x}_k + d_k$.
Step 5. Calculate Z_{k+1} using formula (7).
Step 6. Algorithm 2 is used to update Δ_k.
Step 7. Algorithm 3 is used to update penalty parameter ρ_k.
Step 8. Calculate $y_k = g_{k+1} - g_k = \|\nabla F_{k+1}\| + \|\nabla D_{k+1} Z_{k+1} D_{k+1}\| - \|\nabla F_k\| - \|\nabla D_k Z_k D_k\|$, see Remark 2 for the update mode of B_{k+1}. Let $k := k+1, p := 0$, turn to Step 1.

Remark 1: The trust region subproblem in Step 2 is as follows:

$$\begin{aligned} \text{minimize} \quad & \Phi_k(d_k) = F_k + \nabla F_k^T d \\ & + \tfrac{1}{2} d^T H_k d + \tfrac{\rho_k}{2} \left\| Z_k \left(D_k + \nabla D_k^T d \right) \right\|^2 \\ s.t. \quad & \|d\| \leq \Delta_k = c_k^p \max\left\{1, \tfrac{\|d_{k-1}\|}{\|y_{k-1}\|}\right\} \|g_k\|, \end{aligned} \tag{20}$$

where c_k^p is formula (19).

Remark 2: BFGS update formula [19] in is as follows:

$$\begin{cases} B_{k+1} = \delta_k I - \delta_k \frac{d_k d_k^T}{d_k^T d_k} + \frac{y_k y_k^T}{d_k^T y_k}, & d_k^T y_k > 0 \\ B_{k+1} = B_k + \frac{y_k{}^* (y_k^*)^T}{d_k^T y_k^*} - \frac{B_k d_k d_k^T B_k}{d_k^T B_k d_k}, d_k^T y_k \leq 0 \end{cases} \tag{21}$$

where the scaling parameter $\delta_k = \frac{d_k^T y_k}{\|d_k\|^2}, y_k^* = y_k + \|g_k\| \left(1 - \frac{d_k^T y_k}{\|d_k\|^2}\right) d_k$.

III. CONVERGENCE ANALYSIS

A. Assumptions

The assumptions required are given below. Let the iteration sequence produced by Algorithm 4 be $\{\bar{x}_k\}_{k \geq 0}$, and Ω is a convex set on $R^{n_1+n_2+m_2}$. For any $k, \bar{x}_k$ and $\bar{x}_k + d_k$ are in Ω.

Assumptions:

(i) For any $\bar{x} \in \Omega$, $F(\bar{x})$ and $D(\bar{x})$ are second-order differentiable functions.

(ii) Let $F(\bar{x}), \nabla F(\bar{x}), \nabla^2 F(\bar{x}), D(\bar{x})$ and $\nabla D(\bar{x})$ be uniformly bounded on Ω.

(iii) The sequences $\{H_k\}$ and $\{B_k\}$ are uniformly bounded.

(iiii) $\nabla F(\bar{x})$ is Lipschitz continuous on Ω.

B. Related lemmas

The lemmas needed for the subsequent proof of convergence are given below.

Lemma 1: [5] Under the conditions of Assumptions (i)-(iiii), $Z(x)D(x)$ is Lipschitz continuous on Ω.

We can conclude that $g(\bar{x}_k) = \nabla F(\bar{x}_k) + \rho_k \nabla D(\bar{x}_k) Z(\bar{x}_k) D(\bar{x}_k)$ is Lipschitz continuous on Ω. That is, there exists a constant L such that the following formula holds:

$$\|g(\bar{x}_k) - g(\bar{y}_k)\| \leq L \|\bar{x}_k - \bar{y}_k\|, \forall \bar{x}_k, \bar{y}_k \in \Omega \tag{22}$$

Similar to the conclusions in literature [20], lemmas 2 and 3 can be obtained.

Lemma 2: Under the conditions of Assumptions (i)-(iiii), for any $k > \bar{k}$, there exists a constant $c_1 > 0$ (All the constants in this article are independent of k), such that the following formula holds:

$$Pred_k \geq c_1 \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\| \min\left\{\Delta_k, \frac{\|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|}{\|B_k\|}\right\}. \tag{23}$$

Proof: See Lemma 3.7 in [21]. ■

Lemma 3: Under the conditions of Assumptions (i)-(iiii), there exists a constant $c_2 > 0$ such that the following formula holds:

$$|\hat{F}_k - \hat{F}(x_k + d_k) - Pred_k| \leq c_2 \rho_k \|d_k\|^2 \tag{24}$$

Proof: See Lemma 3.6 in reference [21]. ■

Lemma 4: Under the conditions of Assumptions (i)-(iiii), d_k is the optimal solution of the problem (10), then

$$Pred_k \geq \frac{c_k^{p_k} \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|^2}{2M_k} \tag{25}$$

where $M_k = \max\{1, \|B_k\|\},\quad p_k$ is the maximum p value obtained by k iterations in Algorithm 2.

Proof: d_k is the optimal solution of the subproblem (10), then $\Delta_k \geq \frac{c_k^{p_k} \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|}{\max\{1, \|B_k\|\}}$, so $d_k' = -\frac{c_k^{p_k} (\nabla F_k + \rho_k \nabla D_k Z_k D_k)}{\max\{1, \|B_k\|\}}$ is a feasible solution of the subproblem (10).

$$\begin{aligned} Pred_k &= -(\nabla F_k + \rho_k \nabla P_k Z_k P_k)^T d_k - \frac{1}{2} d_k^T B_k d_k \\ &\geq -(\nabla F_k + \rho_k \nabla P_k Z_k P_k)^T d_k' - \frac{1}{2} (d_k')^T B_k d_k' \\ &= \frac{c_k^{p_k} \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|^2}{\max\{1, \|B_k\|\}} \\ &\quad - \frac{1}{2} \frac{(c_k^{p_k})^2 (\nabla F_k + \rho_k \nabla D_k Z_k D_k)^T B_k (\nabla F_k + \rho_k \nabla D_k Z_k D_k)}{\max\{1, \|B_k\|\}^2} \\ &\geq \frac{c_k^{p_k} \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|^2}{\max\{1, \|B_k\|\}} - \frac{1}{2} \frac{c_k^{p_k} \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|^2}{\max\{1, \|B_k\|\}} \\ &= \frac{1}{2} \frac{c_k^{p_k} \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|^2}{\max\{1, \|B_k\|\}}. \end{aligned} \tag{26}$$

Proof is complete. ■

Lemma 5: If the iteration sequence generated by Algorithm 4 is $\{\bar{x}_k\}_{k \geq 0}$, then

$$\hat{F}_k \leq R_k, \quad \forall k \in N \tag{27}$$

Proof: From formula (14), we can obtain

$$\begin{aligned} \hat{F}_{k+1} &= \eta_{k+1} \hat{F}_{k+1} + (1 - \eta_{k+1}) \hat{F}_{k+1} \\ &\leq \eta_{k+1} \hat{F}_{l(k+1)} + (1 - \eta_{k+1}) \hat{F}_{k+1} \\ &= R_{k+1}. \end{aligned} \tag{28}$$

Formula (27) is obtained. ■

Lemma 6: If the iteration sequence produced by Algorithm 4 is $\{\bar{x}_k\}_{k\geq 0}$, then the nonmonotone sequence $\left\{\hat{F}_{l(k)}\right\}$ decreases monotonically.

Proof: According to the definitions of R_k and $\hat{F}_{l(k)}$,

$$R_k = \eta_k \hat{F}_{l(k)} + (1-\eta_k)\hat{F}_k \leq \eta_k \hat{F}_{l(k)} + (1-\eta_k)\hat{F}_{l(k)} = \hat{F}_{l(k)}. \tag{29}$$

Assuming that $\bar{x}_{k+1}$ is accepted, it can be known from the adaptive trust region radius update strategy that

$$\frac{\hat{F}_{l(k)} - \hat{F}(x_k + d_k)}{\Phi_k(0) - \Phi_k(d_k)} \geq \frac{R_k - \hat{F}(x_k + d_k)}{\Phi_k(0) - \Phi_k(d_k)} \geq \mu_1, \tag{30}$$

For any $k \in N$,

$$\hat{F}_{l(k)} - \hat{F}(x_k + d_k) \geq \mu_1 (\Phi_k(0) - \Phi_k(d_k)) \geq 0. \tag{31}$$

Hence,

$$\hat{F}_{l(k)} \geq \hat{F}_{k+1}, \quad \forall k \in N. \tag{32}$$

If $k \geq N_1$, and $\Phi(k+1) \leq \Phi(k) + 1$, it can be obtained from formula (32)

$$\begin{aligned} \hat{F}_{l(k+1)} &= \max_{0\leq j\leq \Phi(k+1)} \left\{\hat{F}_{k-j+1}\right\} \leq \max_{0\leq j\leq \Phi(k)+1} \left\{\hat{F}_{k-j+1}\right\} \\ &= \max\left\{\hat{F}_{l(k)}, \hat{F}_{k+1}\right\} \leq \hat{F}_{l(k)} \end{aligned} \tag{33}$$

If $k < N_1$, then $\Phi(k) = k$. For any k, $\hat{F}_k \leq \hat{F}_0$, so $\hat{F}_{l(k)} = \hat{F}_0$. To sum up, $\left\{\hat{F}_{l(k)}\right\}$ decreases monotonically. Lemma 6 is proved. ∎

Lemma 7: In Algorithm 4 inline Algorithm 2, the loop iteration terminates after a finite number of steps.

Proof: Proof by contradiction. Assuming Algorithm 2 iterates indefinitely, then

$$r_k^p < \mu_1, p \to \infty. \tag{34}$$

So, when $p \to \infty$, $\quad c_k^p \to 0$.

Therefore, $\|d_k^p\| \leq \Delta_k \to 0$, d_k^p is the solution of the subproblem with respect to p in k iterations. Notice that $\bar{x}$ is not the optimal solution, then there exists a constant $\epsilon_1 > 0$ that makes $\|\nabla F_k + \rho_k \nabla D_k Z_k D_k\| + \|\nabla D_k Z_k D_k\| > \epsilon_1$ hold, assuming $\|\nabla F_k + \rho_k \nabla D_k Z_k D_k\| > \frac{\epsilon_1}{2}$ and $\|\nabla D_k Z_k D_k\| > \frac{\epsilon_1}{2}$. Combined with Lemmas 3 and 4, then

$$\begin{aligned} \left|\frac{\hat{F}_k - \hat{F}(x_k + d_k^p)}{\Phi_k(0) - \Phi_k(d_k^p)} - 1\right| &= \left|\frac{\hat{F}_k - \hat{F}(x_k + d_k^p) - Pred_k^p}{Pred_k^p}\right| \\ &\leq \frac{2M_k c_3 \rho_k \|d_k^p\|^2}{c_k^p \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|^2} \\ &\leq \frac{8M_k O(\Delta_k)^2}{c_k^p {\epsilon_1}^2}. \end{aligned} \tag{35}$$

As $p \to \infty$, formula (35) tends to 0, giving the formula below

$$\lim_{p\to\infty} \frac{\hat{F}_k - \hat{F}(x_k + d_k^p)}{\Phi_k(0) - \Phi_k(d_k^p)} = 1. \tag{36}$$

In combination with equations (17), (36) and Lemma 5, it can be obtained

$$r_k^p = \frac{R_k - \hat{F}(x_k + d_k^p)}{\Phi_k(0) - \Phi_k(d_k^p)} \geq \frac{\hat{F}_k - \hat{F}(x_k + d_k^p)}{\Phi_k(0) - \Phi_k(d_k^p)}, \tag{37}$$

In this case, $r_k^p \geq \mu_1$ contradicts the hypothesis $r_k^p < \mu_1$, and Lemma 7 is proved. ∎

C. Proof of convergence

1) Convergence as penalty parameter tends to infinity: According to the algorithm in Section 2.3, we know that $k \to \infty$, $\rho_k \to \infty$. Lemmas 8 and 9 indicate that when $\rho_k \to \infty$, $\{\bar{x}_k\}_{k\geq 0}$ satisfies the FJ condition or the infeasible FJ condition [21].

Lemma 8: Under the conditions of Assumptions (i)-(iiii) hold and $\rho_k \to \infty$, for any $k \in \{k_i\}$ and $\lim_{k_i\to\infty} \|Z_{k_i} D_{k_i}\| = 0$, if there is an iterated sequence $\{k_i\}$ of index set satisfying $\|Z_k D_k\| > 0$. Then the iterated sequence of index set satisfying $\{k_i\}$ of $\{\bar{x}_k\}_{k\geq 0}$ satisfies FJ condition under limit case.

Proof: See Lemma 3.9 in [21]. ∎

Lemma 9: Under the conditions of Assumptions (i)-(iiii) hold and $\rho_k \to \infty$, for any $k \in \{k_i\}$, if there is an iterated sequence $\{k_i\}$ of index set satisfying $\|Z_k D_k\| \geq \varepsilon > 0$. Then the iterated sequence of index $\{k_i\}$ satisfies the infeasible FJ condition under limit case.

Proof: See Lemma 3.8 in [21]. ∎

2) Convergence when penalty parameter is bounded: Assuming that ρ_k is bounded, for any $k \geq \bar{k}$, there is $\rho_k = \bar{\rho} < \infty$.

Lemma 10: Under the conditions of Assumptions (i)-(iiii), there is $\|\nabla F_k + \bar{\rho} \nabla D_k Z_k D_k\| + \|\nabla D_k Z_k D_k\| > \epsilon_1$ at any iteration k. There exists a constant $c_3 > 0$, such that the following formula holds.

$$Pred_k \geq c_3 \Delta_k. \tag{38}$$

Proof: See Lemma 3.10 in [21]. ∎

Lemma 11: Under the conditions of Assumptions (i)-(iiii), if for any k, $\|\nabla F_k + \bar{\rho} \nabla D_k Z_k D_k\| + \|\nabla D_k Z_k D_k\| > \epsilon_1$, then Ared $_{k_j} \geq \mu_1$ Pred $_{k_j}$ is satisfied for some finite j. In other words, an admissible trial step is obtained by a finite step computation.

Proof: Since for any k, $\|\nabla F_k + \bar{\rho} \nabla D_k Z_k D_k\| + \|\nabla D_k Z_k D_k\| > \epsilon_1$, combined with Lemmas 3 and 10, then

$$\begin{aligned} \left|\frac{\hat{F}_k - \hat{F}\left(x_k + d_{k_j}\right)}{\Phi_k(0) - \Phi_k\left(d_{k_j}\right)} - 1\right| &= \left|\frac{\hat{F}_k - \hat{F}\left(x_k + d_{k_j}\right) - Pred_{k_j}}{Pred_{k_j}}\right| \\ &\leq \frac{c_2 \rho_k \left\|d_{k_j}\right\|^2}{c_3 \Delta_{k_j}} \\ &\leq \frac{c_2 \rho_k \Delta_{k_j}}{c_3}. \end{aligned} \tag{39}$$

In the iterative process of the algorithms in Section 2.3, let $k \geq \tilde{k}$ and $k_j \geq \tilde{k}$, when $k \to \infty$, $\sum_{k\to\infty} \Delta_{k_j} = 0$. Thus, the right end of formula (39) tends to 0. From equation (17) and Lemma 5, we can obtain

$$r_k = \frac{R_k - \hat{F}\left(x_k + d_{k_j}\right)}{\Phi_k(0) - \Phi_k\left(d_{k_j}\right)} \geq \frac{\hat{F}_k - \hat{F}\left(x_k + d_{k_j}\right)}{\Phi_k(0) - \Phi_k\left(d_{k_j}\right)}. \tag{40}$$

This means that the trial step is accepted, i.e. $r_k \geq \mu_1$, the Lemma 11 is proved. ■

3) Global convergence:

Theorem 1: Under the conditions of Assumptions (i)-(iiii), iteration sequence $\{\bar{x}_k\}_{k\geq 0}$ satisfies the following formula:

$$\liminf_{k\to\infty} [\|\nabla F_k\| + \|\nabla D_k Z_k D_k\|] = 0. \tag{41}$$

Proof: To demonstrate that equation (41) is true, consider proving that $\liminf_{k\to\infty}[\|\nabla F_k + \bar{\rho}\nabla D_k Z_k D_k\| + \|\nabla D_k Z_k D_k\|] = 0$ is true. Use the idea of counterproof. Assume that for any k, there exists a constant $\epsilon_1 > 0$ and an infinite set

$$\Gamma = \{k \mid \|\nabla F_k + \bar{\rho}\nabla D_k Z_k D_k\| + \|\nabla D_k Z_k D_k\| > \epsilon_1\}. \tag{42}$$

It can be known from equation (17) and Lemma 11

$$\mu_1 Pred_k \leq R_k - \hat{F}(x_k + d_k). \tag{43}$$

It can be known from formula (43) and Lemma 5 above

$$\mu_1 Pred_k \leq \hat{F}_{l(k)} - \hat{F}(x_k + d_k). \tag{44}$$

From formula (44) and Lemma 4, we get

$$\begin{aligned}\hat{F}_{l(k)} - \hat{F}_{l(k)+1} &\geq \mu_1 \operatorname{Pred}_k \\ &\geq \frac{\mu_1 c_k^{p_k} \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|^2}{2M_k}.\end{aligned} \tag{45}$$

From Lemma 6 and the assumption (iii), $\left\{\hat{F}_{l(k)}\right\}$ is monotone nonincreasing and $\hat{F}_k$ is bounded, $\left\{\hat{F}_{l(k)}\right\}$ converges. According to the assumption (iii), there exists a positive constant M, such that for any k there exists $M_k \leq M$, then

$$\sum_{k\in\Gamma} \frac{\mu_1 c_k^{p_k} \|\nabla F_k + \rho_k \nabla D_k Z_k D_k\|^2}{2M} < \infty, \tag{46}$$

where p_k is the maximum p value obtained by k iterations in Algorithm 2. In the iteration of Algorithm 2, when $k \in \Gamma$ and $k \to +\infty, p_k \to +\infty$.

According to the definition of $p_k (k \in \Gamma)$, the solution $\bar{d}_k$ of the subproblem

$$\begin{array}{lc} \text{minimize} & \Phi_k(d_k) = F_k + \nabla F_k^T d \\ & +\frac{1}{2} d^T H_k d + \frac{\rho_k}{2}\left\|Z_k\left(D_k + \nabla D_k^T d\right)\right\|^2 \\ s.t. & \|d\| \leq c_k^{p_k - 1} \max\left\{1, \frac{\|d_{k-1}\|}{\|y_{k-1}\|}\right\} \|g_k\| = \frac{\Delta_k}{t}, \end{array} \tag{47}$$

is not acceptable, where $t > 0$ is a constant. So

$$r_k = \frac{R_k - \hat{F}\left(x_k + \bar{d}_k\right)}{\Phi_k(0) - \Phi_k(d_k)} < \mu_1, k \in \Gamma. \tag{48}$$

According to Lemma 7, when $k \in \Gamma$ and $k \to +\infty$, there is $r_k \geq \mu_1$, which contradicts formula (48). Hence, the hypothesis is not true, $\liminf_{k\to\infty}[\|\nabla F_k + \bar{\rho}\nabla D_k Z_k D_k\| + \|\nabla D_k Z_k D_k\|] = 0$. Theorem 1 is proved. ■

IV. Numerical test

An algorithm for solving the MNBLO problem is presented and its global convergence is proved. Algorithm 4 is applied in this section to solve the following MNBLO problems. Examples 1-3 are selected refer to [22–24], and Example 4 is adapted from [21].

Parameter setting: Given the initial point $\bar{x}_0$, select $\varepsilon_0 > 0$. $\epsilon_1 = 10^{-6}, \epsilon_2 = 10^{-8}$, $\mu_1 = 0.25, \mu_2 = 0.75, \sigma_1 = 0.25, \sigma_2 = 1.5, c_0 = 0.2, \eta_0 = 0.2, \eta_{\min} = 0.15, \eta_{\max} = 0.9$. η_k is updated as follows.

$$\eta_k = \begin{cases} \eta_0/2, & k = 1 \\ (\eta_{k-1} + \eta_{k-2})/2, & k \geq 2 \end{cases} \tag{49}$$

The algorithm is compiled and run on MATLAB R2016a.

Example 1:
$$\begin{cases} \min\limits_x \left(y_2 - x_1, -x_2^2 - 2y_1\right) \\ \min\limits_y \left(2x_2 y_1^3, -x_1^2 - y_2^2\right) \\ \text{s.t. } x_1 - 2.5 \leq 0 \\ \quad y_2 - 2x_1 \leq 0 \\ \quad 1 - x_2^2 y_1 \leq 0 \\ \quad x_2 - 4y_1 \leq 0 \\ \quad x_1, x_2, y_1, y_2 \geq 0. \end{cases}$$

Using equations (2), (3) and KKT optimality condition, the following NP problem is obtained.

$$\begin{array}{ll} \min_{x,y} & \left\{\left(\frac{F_1(x,y) - F_1^{\min}}{F_1^{\max} - F_1^{\min}}\right)^2 + \left(\frac{F_2(x,y) - F_2^{\min}}{F_2^{\max} - F_2^{\min}}\right)^2\right\}^{\frac{1}{2}} \\ \text{s.t.} & 6\beta_1 x_2 y_1^2 - \lambda_2 x_2^2 - 4\lambda_3 - \lambda_4 = 0, \\ & -2\beta_2 y_2 + \lambda_1 - \lambda_5 = 0, \\ & \lambda_1 (y_2 - 2x_1) = 0, \\ & \lambda_2 \left(1 - x_2^2 y_1\right) = 0, \\ & \lambda_3 (x_2 - 4y_1) = 0, \\ & \lambda_4 y_1 = 0, \lambda_5 y_2 = 0, \\ & x_1 - 2.5 \leq 0, \\ & y_2 - 2x_1 \leq 0, \\ & 1 - x_2^2 y_1 \leq 0, \\ & x_2 - 4y_1 \leq 0, \\ & (x_1, x_2)^T \geq 0, (y_1, y_2)^T \geq 0, \\ & (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)^T \geq 0, \end{array}$$

where $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)^T \geq 0$ is the Lagrange multiplier vector, β_i is the linear combination of parameter, $i = 1, 2$. $F_k^{\max}$ and $F_k^{\min}$ represent the best and worst optimal solutions of the kth objective function of all constraints, respectively, $k = 1, 2$. By using formulas (7), (8) and (9), we convert the NP problem into an unconstrained problem.

By using Algorithm 4 and MATLAB programming, the optimal solution $(x_1, x_2, y_1, y_2) = (2.5, 3, 1, 2)$ is obtained. The objective functions $(F_1, F_2) = (-0.5, -11)$, $(f_1, f_2) = (6, -10.25)$.

Example 2:
$$\begin{cases} \min_x & \left(\frac{5}{3}x^2, \frac{5}{2}(y - 10)^2\right) \\ \text{s.t.} & -x + y \leq 0, \\ & 0 \leq x \leq 15, \\ \min_y & \left(x + 2y - 30, x + 0.5y^2\right) \\ \text{s.t.} & 0 \leq y \leq 15. \end{cases}$$

Using equations (2), (3) and KKT optimality condition, the following NP problem is obtained.

$$\begin{array}{ll}\min_{x,y} & \left\{\left(\frac{F_1(x,y)-F_1^{\min}}{F_1^{\max}-F_1^{\min}}\right)^2+\left(\frac{F_2(x,y)-F_2^{\min}}{F_2^{\max}-F_2^{\min}}\right)^2\right\} \\ \text{s.t.} & -x+y\leq 0, \\ & 0\leq x\leq 15, \\ & 2\beta_1+\beta_2 y-\lambda_1+\lambda_2=0, \\ & \lambda_1 y=0, \\ & \lambda_2(y-15)=0, \\ & 0\leq y\leq 15, \\ & (\lambda_1,\lambda_2)^T\geq 0.\end{array}$$

As above, Algorithm 4 is applied to obtain the optimal solution $(x,y) = (4.9953, 5.0111)$. The objective functions $(F_1,F_2) = (41.5884, 62.2228)$, $(f_1,f_2) = (-14.9825, 17.5509)$.

Example 3:
$$\begin{cases}\max_x & (xy, x^2+y^2) \\ & x\geq 0 \\ \max_y & (x^3+y, 4e^x+y^2) \\ \text{s.t.} & x^2+y^2\leq 12.5 \\ & x^2y-1\geq 0 \\ & x-y\geq 0 \\ & 4y-x\geq 0 \\ & y\geq 0.\end{cases}$$

Using equations (2), (3) and KKT optimality condition, the following NP problem is obtained.

$$\begin{array}{ll}\min_{x,y} & \left\{\left(\frac{F_1(x,y)-F_1^{\min}}{F_1^{\max}-F_1^{\min}}\right)^2+\left(\frac{F_2(x,y)-F_2^{\min}}{F_2^{\max}-F_2^{\min}}\right)^2\right\}^{\frac{1}{2}} \\ \text{s.t.} & \beta_1+2\beta_2 y+2\lambda_1 y-\lambda_2 x^2+\lambda_3-4\lambda_4-\lambda_5=0, \\ & \lambda_1\left(x^2+y^2-12.5\right)=0, \\ & \lambda_2\left(1-x^2y\right)=0, \\ & \lambda_3(y-x)=0, \\ & \lambda_4(x-4y)=0, \\ & \lambda_5 y=0, \\ & x^2+y^2\leq 12.5, \\ & x^2y-1\geq 0, \\ & x-y\geq 0, \\ & 4y-x\geq 0, \\ & x,y\geq 0, \\ & (\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_5)^T\geq 0.\end{array}$$

As above, Algorithm 4 is applied to obtain the optimal solution $(x,y) = (2.5000, 2.4932)$. The objective functions $(F_1,F_2) = (6.2330, 12.4660)$, $(f_1,f_2) = (18.1182, 54.9460)$.

Example 4:
$$\begin{cases}\min_x & (y_1^2+y_3^2-y_1y_3-4y_2-7x_1+4x_2, \\ & y_1^2+5x_2) \\ \text{s.t.} & x_1+x_2\leq 1 \\ & (x_1,x_2)^T\geq 0 \\ \min_{y_1,y_2} & (y_1^2+\frac{1}{2}y_2^2+\frac{1}{2}y_3^2+y_1y_2+(1-3x_1)\,y_1 \\ & +(1+x_2)\,y_2, \frac{1}{3}y_3^2-4x_1y_1) \\ \text{s.t.} & 2y_1+y_2-y_3+x_1-2x_2+2\leq 0 \\ & (y_1,y_2,y_3)^T\geq 0.\end{cases}$$

Using equations (2), (3) and KKT optimality condition, the following NP problem is obtained.

$$\begin{array}{ll}\min\limits_{x,y_1,y_2,y_3} & \left\{\left(\frac{F_1(x,y)-F_1^{\min}}{F_1^{\max}-F_1^{\min}}\right)^2+\left(\frac{F_2(x,y)-F_2^{\min}}{F_2^{\max}-F_2^{\min}}\right)^2\right\} \\ \text{s.t.} & \beta_1\left(2y_1+y_2+1-3x_1\right)-4\beta_2x_1+2\lambda_1-\lambda_2=0, \\ & \beta_1\left(y_2+y_1+1+x_2\right)+\lambda_1-\lambda_3=0, \\ & \beta_1 y_3+\frac{2}{3}\beta_2 y_3-\lambda_1-\lambda_4=0, \\ & \lambda_1\left(2y_1+y_2-y_3+x_1-2x_2+2\right)=0, \\ & \lambda_2 y_1=0, \lambda_3 y_2=0, \lambda_4 y_3=0, \\ & x_1+x_2\leq 1, \\ & 2y_1+y_2-y_3+x_1-2x_2+2\leq 0, \\ & (x_1,x_2)^T\geq 0, (y_1,y_2,y_3)^T\geq 0, \\ & (\lambda_1,\lambda_2,\lambda_3,\lambda_4)^T\geq 0.\end{array}$$

As above, Algorithm 4 is applied to obtain the optimal solution $(x_1,x_2,y_1,y_2,y_3) = (0.5380, 0.2620, 0.0000, 0.0000, 1.9160)$. The objective functions $(F_1,F_2) = (0.9531, 1.3100), (f_1,f_2) = (1.8355, 1.2237)$.

As can be seen from the solution results, the optimal solutions for examples 1-3 are the same as the results in reference. In Example 4, an objective is added to both levels of objective based on reference [21]. The optimal solution is obtained by using Algorithm 4.

V. Conclusion

There have been few studies on the application of nonmonotone technique and adaptive radius update strategy to the MNBLO problem. For the MNBLO problem, the proposed nonmonotone adaptive algorithm overcomes the shortcoming of slow convergence. Theorem 1 shows that the algorithm can maintain global convergence. The feasibility of the proposed penalty method is further explained by numerical experiments.

References

[1] A. Ou-yassine and S.-d. Krit, "An introduction of a new square quadratic proximal point scalarization method for multiobjective programming," *IAENG International Journal of Applied Mathematics*, vol. 51, no. 2, pp. 439–443, 2021.

[2] J. E. Falk and J. Liu, "On bilevel programming, part i: General nonlinear cases," *Mathematical Programming*, vol. 70, no. 1, pp. 47–72, 1995.

[3] N. V. Thoai, Y. Yamamoto, and A. Yoshise, "Global optimization method for solving mathematical programs with linear complementarity constraints," *Journal of Optimization Theory and Applications*, vol. 124, no. 2, pp. 467–490, 2005.

[4] M. G. P. Fiacco A V, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. New York: John Wiley and Sons, 1968.

[5] J. E. Dennis, M. M. El-Alem, and K. A. Williamson, "A trust-region approach to nonlinear systems of equalities and inequalities," in *Fifth SIAM Conference on Optimization*, 1999.

[6] R. Fletcher, "An l1 penalty method for nonlinear constraints," 1985.

[7] M. Powell, "A new algorithm for unconstrained optimization," *Nonlinear Programming*, pp. 31–65, 1970.

[8] L. Niu, "A new trust region algorithm for nonlinear constrained optimization," *Journal of Computational Mathematics*, vol. 28, pp. 72–86, 2010.

[9] N. Y. Deng, Y. Xiao, and F. J. Zhou, "Nonmonotonic trust region algorithm," *Journal of Optimization Theory and Applications*, vol. 76, no. 2, pp. 259–285, 1993.

[10] M. Ahookhosh and K. Amini, "An efficient nonmonotone trust-region method for unconstrained optimization," *Numerical Algorithms*, vol. 59, no. 4, pp. 523–540, 2012.

[11] Sartenaer and A., "Automatic determination of an initial trust region in nonlinear programming," *SIAM Journal on Scientific Computing*, vol. 18, no. 6, pp. 1788–1803, 1997.

[12] X. S. Zhang, *NN Models for General Nonlinear Programming[M]*. Springer US, 2000.

[13] L. L. Z. Zhang X S, Zhang J L, “An adaptive trust region method and its convergence,” *Science in China,Ser.A*, no. 5, pp. 620–631, 2002.
[14] L. G, “A trust region method with automatic determination of the trust region radius,” *Gongcheng Shuxue Xuebao/Chinese Journal of Engineering Mathematics*, vol. 23, no. 5, pp. 843–848, 2006.
[15] M. O., *Nonlinear Programming*. New York: McGraw-Hill Book Company, 1969.
[16] T. Steihaug, “The conjugate gradient method and trust regions in large scale optimization,” *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 626–637, 1983.
[17] Z. Li, “Linear convergence rate of mixed broken line method and tr algorithm[d],” Master’s thesis, Nanjing Normal University, 2002.
[18] Y. X. Yuan, “On the convergence of a new trust region algorithm,” *Numerische Mathematik*, vol. 70, no. 4, pp. 515–539, 1995.
[19] X. Yanqin, “Based on the research of nonmonotone trust region algorithm,” Master’s thesis.
[20] M. Powell, “Convergence properties of a class of minimization algorithms,” *Optimization Letters*, pp. 1–27, 1975.
[21] B. El-Sobky and Y. Abo-Elnaga, “A penalty method with trust-region mechanism for nonlinear bilevel optimization problem,” *Journal of Computational and Applied Mathematics*, vol. 340, pp. 360–374, 2018.
[22] N. X. Hongsheng Xia, Zhaohan Sheng, “A bilevel nonlinear multi-objective decision making method based on stackelberg master-slave strategy,” *Journal of Southeast University*, vol. 22, no. 3, pp. 125–129, 1992.
[23] T. C. Li Lei, “A kind of global optimization method for bilevel multiobjective decision making,” *Journal of Harbin University of Science and Technology*, vol. 6, no. 6, pp. 25–29, 2001.
[24] X. N. Xia Hongsheng, Sheng Zhaohan, “A bilevel nonlinear multiobjective decision making method,” *Control and Decision*, vol. 7, no. 4, pp. 277–282, 1992.