AIMS Mathematics

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

AIMS Mathematics, 7(4): 5534–5562.
DOI:10.3934/math.2022307
Received: 22 September 2021
Revised: 16 December 2021
Accepted: 29 December 2021
Published: 10 January 2022

Research article

An interior-point trust-region algorithm to solve a nonlinear bilevel programming problem

B. El-Sobky*and G. Ashry

Department of Mathematics and Computer Science, Alexandria University, Faculty of Science, Egypt

* **Correspondence:** Email: bothinaelsobky@yahoo.com.

Abstract: In this paper, a nonlinear bilevel programming (NBLP) problem is transformed into an equivalent smooth single objective nonlinear programming (SONP) problem utilized slack variable with a Karush-Kuhn-Tucker (KKT) condition. To solve the equivalent smooth SONP problem effectively, an interior-point Newton's method with Das scaling matrix is used. This method is locally method and to guarantee convergence from any starting point, a trust-region strategy is used. The proposed algorithm is proved to be stable and capable of generating approximal optimal solution to the nonlinear bilevel programming problem.
A global convergence theory of the proposed algorithm is introduced and applications to mathematical programs with equilibrium constraints are given to clarify the effectiveness of the proposed approach.

Keywords: nonlinear bilevel programming problem; Newton's method; Das scaling matrix; trust-region technique; global convergence
Mathematics Subject Classification: 93D52, 49N35, 93D22, 49N10, 65K05

1. Introduction

Bilevel programming problem has increasingly been addressed in the literature, both from the theoretical and computational points of view [14]. This model has been widely applied to decentralized planning problems involving a decision progress with a hierarchical structure. It is characterized by the existence of two optimization problems in which the constraint region of the first-level problem is implicitly determined by another optimization problem. The NBLP problem is hard to solve. In fact, the problem has been proved to be NP-hard [8]. However, the NBLP problem is used so extensively in resource allocation, finance budget, price control, transaction network etc. [1, 7, 28, 29, 39] that many researches have been devoted to this field, which leads to a rapid development in theories and algorithms. For the detailed expositions, the reader may consult [21, 33].

In this paper we will consider the following NBLP problem

$$\begin{aligned} \min_t \quad & f_u(t, y) \\ s.t. \quad & g_u(t, y) \le 0, \\ & \min_y \quad f_l(t, y), \\ & s.t. \quad g_l(t, y) \le 0, \end{aligned} \tag{1.1}$$

where $t \in \Re^{n_1}$ and $y \in \Re^{n_2}$. The functions $f_u : \Re^{n_1+n_2} \to \Re$, $f_l : \Re^{n_1+n_2} \to \Re$, $g_u : \Re^{n_1+n_2} \to \Re^{m_1}$, and $g_l : \Re^{n_1+n_2} \to \Re^{m_2}$ are assumed to be at least twice continuously differentiable function.

There are several approaches have proposed to solve problem 1.1, see [2,3,25,35,40]. One of these approaches and used in this paper, is converted the original two level problems to a single level one by replacing the lower level optimization problem with its Karush-Kuhn-Tucker (KKT) conditions, see [24,41]. By KKT optimality conditions for the lower-level problem, then we can reduce the NBLP problem 1.1 to one-level programming problem. This problem is non-convex and non-differentiable, moreover the regularity assumptions which are needed to successfully handle smooth optimization problems are never satisfied and it is not good to use our approach. So, we add slack variables for inequalities constraints in problem 1.1.

By adding slack variables $s_u \in \Re^{m_1}$ and $s_l \in \Re^{m_2}$ to the upper inequality constraint $g_u(t, y)$ and the lower inequality constraint $g_l(t, y)$ respectively, then NBLP problem 1.1 can be written as follows

$$\begin{aligned} \min_t \quad & f_u(t, y) \\ s.t. \quad & g_u(t, y) + s_u = 0, \\ & \min_y \quad f_l(t, y), \\ & s.t. \quad g_l(t, y) + s_l = 0, \\ & \qquad s_u \ge 0, \qquad s_l \ge 0. \end{aligned}$$

The above NBLP problem can be simplified as follows

$$\begin{aligned} \min_t \quad & f_u(t, y) \\ s.t. \quad & \tilde{g}_u(t, y, s_u) = 0, \\ & \min_y \quad f_l(t, y), \\ & s.t. \quad \tilde{g}_l(t, y, s_l) = 0, \\ & \qquad s \ge 0, \end{aligned} \tag{1.2}$$

where $\tilde{g}_u(t, y, s_u) = g_u(t, y) + s_u \in \Re^{m_1}$, $\tilde{g}_l(t, y, s_l) = g_l(t, y) + s_l \in \Re^{m_2}$, and $s = (s_u, s_l)^T \in \Re^{m_1+m_2}$.

Applying KKT conditions only on the lower-level problem without the constraint $s \ge 0$, then we can reduce the NBLP problem 1.2 to the following smooth SONP problem:

$$\begin{aligned} \min_t \quad & f_u(t, y) \\ s.t. \quad & \tilde{g}_u(t, y, s_u) = 0, \\ & \nabla_y f_l(t, y) + \nabla_y \tilde{g}_l(t, y, s_l)\mu_l = 0, \\ & \tilde{g}_l(t, y, s_l) = 0, \\ & s \ge 0, \end{aligned} \tag{1.3}$$

where $\mu_l \in \Re^{m_2}$ is a Lagrange multiplier vector associated with equality constraint $\tilde{g}_l(t, y, s_l)$, see [5].

Using problem 1.3, to overcome the difficulty that problem 1.1 does not satisfy any regularity assumptions, which are needed for successfully handling smooth optimization problems, and pave the way for using the proposed approach to solve problem 1.1. To simply our discussion, we introduce the following notations. $x = (t, y, s)^T \in \Re^n$, $n = n_1 + n_2 + m_1 + m_2$, $h(x) \in \Re^m$ represents the vector of equality constraints such that $m = m_1 + m_2 + n_2$. Then problem 1.3 can be written as follows

$$\begin{array}{ll} minimize & f_u(x) \\ subject\ to & h(x) = 0, \\ & v \le x \le w, \end{array} \tag{1.4}$$

where $v \in \{\Re \bigcup \{-\infty\}\}^n$, $w \in \{\Re \bigcup \{+\infty\}\}^n$, and $v < w$.

Various approaches have been proposed to solve the SONP problem 1.4, see [5, 9–11, 15–19]. In this paper, we use Newton's interior point method with Das scaling matrix [12] to solve problem 1.4. Newton's method converges quadratically to a stationary point under reasonable assumptions if the starting point sufficiently closed to the stationary point. It may not converge if the starting point is far away from the stationary point. To guarantee convergence from any starting point, a trust-region strategy is used. The trust-region strategy can induce strongly global convergence, which is very important method for solving SONP problem and is more robust when it deals with rounding errors. It does not require the objective function of the model be convex or the Hessian of the objective function must be positive definite. Also, some criteria are used to test the trial step is acceptable or no. If it is not acceptable, then the subproblem must be resolved with a reduced the trust-region radius. For the detailed expositions, the reader may consult [4, 17, 20–23, 30, 32, 36, 42, 43, 45, 46].

A reduced hessian technique is used in this paper to overcome some difficulties in trust-region subproblem. This technique was suggested by [6, 37] and used by [19, 20].

In this paper, we use the symbol, $f_{u_k} \equiv f_u(x_k)$, $h_k \equiv h(x_k)$, $P_k \equiv P(x_k)$ $\ell_k \equiv \ell(x_k, \lambda_k)$, $\nabla_x \ell_k \equiv \nabla_x \ell(x_k, \lambda_k)$, and so on to denote the function value at a particular point. Finally, all norms are l_2-norms.

The rest of the paper is organized as follows. In section 2, we introduce detailed description for the proposed method to solve problem 1.4. Section 3 is devoted to analysis of the global convergence of the proposed algorithm. Section 4 contains implementation of the proposed algorithm and the results of test problems. Section 5 contains concluding remarks.

2. An interior-point method with trust-region algorithm

In this section, firstly, we will consider the detailed description for the Newton's interior-point method with Das scaling matrix to solve SONP problem 1.4. Secondly, to guarantee convergence from any starting point, we will introduce the detailed description for trust-region strategy. Finally, we clarify main steps for general algorithm to solve NBLP 1.1.

2.1. Newton's method with scaling matrix

Motivated by the impressive computational performance of Newton's interior-point method for solving SONP problem 1.4, let

$$\ell(x, \lambda) = f_u(x) + \lambda^T h(x), \tag{2.1}$$

be a Lagrangian function associated with problem 1.4 without the constraints $v \le x \le w$, and let

$$L(x, \lambda, \mu^v, \mu^w) = \ell(x, \lambda) - \mu^{v^T}(x - v) - \mu^{w^T}(w - x), \tag{2.2}$$

be a Lagrangian function associated with problem 1.4 with the constraints $v \leq x \leq w$. The vectors $\lambda \in \Re^m$, $\mu^v \in \Re^n$, and $\mu^w \in \Re^n$ represent Lagrange multiplier vectors associated with the constraints $h(x) = 0$, $0 \leq (x-v)$, and $0 \leq (w-x)$ respectively. Let $\hat{\boldsymbol{G}} = \{x : v \leq x \leq w\}$ and $int(\hat{\boldsymbol{G}}) = \{x : v < x < w\}$.

The first-order necessary conditions for the point x_* to be a local minimizer of problem 1.4 are the existence of multipliers $\lambda_* \in \Re^m$, $\mu_*^v \in \Re_+^n$, and $\mu_*^w \in \Re_+^n$, such that $(x_*, \lambda_*, \mu_*^v, \mu_*^w)$ satisfies

$$\nabla_x \ell(x_*, \lambda_*) - \mu_*^v + \mu_*^w = 0, \tag{2.3}$$

$$h(x_*) = 0, \tag{2.4}$$

$$v \leq x_* \leq w, \tag{2.5}$$

and for all i corresponding to $x^{(i)}$ with finite bound, we have

$$(\mu_*^v)^{(i)}(x_*^{(i)} - v^{(i)}) = 0, \tag{2.6}$$

$$(\mu_*^w)^{(i)}(w^{(i)} - x_*^{(i)}) = 0, \tag{2.7}$$

where $\nabla_x \ell(x_*, \lambda_*) = \nabla f_u(x_*) + \nabla h(x_*)\lambda_*$.

The proposed algorithm here, like its predecessors in [12, 18, 19], starts at a point strictly feasible with respect to the bounds on the variables and produces iterates that are strictly feasible with respect to the bounds (i.e. 'in the interior'). Define a diagonal scaling matrix $P(x) = diag(p(x)$ whose diagonal elements $p(x)$ are given by

$$p^{(i)}(x) = \begin{cases} \sqrt{(x^{(i)} - v^{(i)})}, & \text{if } v^{(i)} > -\infty \text{ and } (\nabla_x \ell(x,\lambda))^{(i)} \geq 0, \\ \sqrt{(w^{(i)} - x^{(i)})}, & \text{if } w^{(i)} < +\infty \text{ and } (\nabla_x \ell(x,\lambda))^{(i)} < 0\,, \\ 1, & \text{otherwise.} \end{cases} \tag{2.8}$$

Using the matrix P(x), then $(x_*, \lambda_*, \mu_*^v, \mu_*^w)$ satisfy the KKT conditions [2.3-2.7] if and only if

$$P^2(x)\nabla_x \ell(x, \lambda) = 0, \tag{2.9}$$

$$h(x) = 0. \tag{2.10}$$

For more details about the proof, see [12].

Applying Newton's method on the nonlinear system [2.9-2.10], then we have

$$[P^2(x)\nabla_x^2 \ell(x,\lambda) + diag(\nabla_x \ell(x,\lambda))diag(\theta(x))]\Delta x + P^2(x)\nabla h(x)\Delta\lambda = -P^2(x)\nabla_x \ell(x,\lambda), \tag{2.11}$$

$$\nabla h(x)^T \Delta x = -h(x). \tag{2.12}$$

where $\theta(x)$ is a vector whose components are given by

$$\theta^{(i)}(x) = \begin{cases} 1, & \text{if } v^{(i)} > -\infty \text{ and } (\nabla_x \ell(x,\lambda))^{(i)} \geq 0, \\ -1, & \text{if } w^{(i)} < +\infty \text{ and } (\nabla_x \ell(x,\lambda))^{(i)} < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

For more details see [18].

In our method, the matrix $P(x)$ must be nonsingular, so we restrict the point $x \in int(\hat{\boldsymbol{G}})$. Multiplying both sides of equation 2.11 by $P^{-1}(x)$, then we have

$$[P(x)\nabla_x^2 \ell(x,\lambda) + P^{-1}(x)diag(\nabla_x \ell(x,\lambda))diag(\theta(x))]\Delta x + P(x)\nabla h(x)\Delta\lambda = -P(x)\nabla_x \ell(x,\lambda),$$

$$\nabla h(x)^T \Delta x \;=\; -h(x).$$

Substituting $\Delta x = P(x)\, d$ in the above two system, then we have

$$[P(x)H(x,\lambda)P(x) + diag(\nabla_x \ell(x,\lambda))diag(\theta(x))]\, d \;+\; P(x)\nabla h(x)\Delta\lambda = -P(x)\nabla_x \ell(x,\lambda), \tag{2.14}$$
$$(P(x)\nabla h(x))^T d \;=\; -h(x), \tag{2.15}$$

where $H(x,\lambda) = \nabla_x^2 \ell(x,\lambda)$ represents the Hessian of the Lagrange function 2.1 or an approximation to it. It is easy to see that the step generated by the above system coincides with the solution of the following quadratic programming subproblem

$$\begin{array}{ll} minimize & \ell(x,\lambda) + (P(x)\nabla_x \ell(x,\lambda))^T d + \frac{1}{2} d^T B d \\ subject\ to & h(x) + (P(x)\nabla h(x))^T d = 0, \end{array} \tag{2.16}$$

where $B = P(x)H(x,\lambda)P(x) + diag(\nabla_x \ell(x,\lambda))diag(\theta(x))$. This means that, the point (x_*, λ_*) that satisfies the KKT conditions for subproblem 2.16 will satisfy the KKT conditions for problem 1.4.

Although Newton's method converges quadratically to a stationary point under reasonable assumptions, it may not converge to a stationary point if the starting point is far away from the solution. To overcome this disadvantage and to guarantee convergence from any starting point, we use the trust-region technique.

2.2. *Trust-region technique*

Trust-region methods can induce strongly global convergence, which are very important methods for solving a smooth nonlinear programming problem and are more robust when they deal with rounding errors. It does not require the objective function of the model be convex. Also, it does not demand the Hessian of the objective function must positive definite.

The trust-region subproblem associated with problem 2.16 is

$$\begin{array}{ll} minimize & q_k(P_k d_k) = \ell_k + (P_k \nabla_x \ell_k)^T d + \frac{1}{2} d^T B_k d \\ subject\ to & h_k + (P_k \nabla h_k)^T d = 0, \\ & \|d\| \le \delta_k, \end{array} \tag{2.17}$$

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

Subproblem 2.17 may be infeasible, because there may be no intersecting points between the constraint $\|d\| \le \delta_k$ and $h_k + (P_k \nabla h_k)^T d = 0$ constraints. Even if they intersect, there is no warranty that this will continue true if δ_k is decreased. For more details see [13]. To overcome this difficulty, we use a reduced hessian technique. This technique was suggested by [6, 37] and used by [19, 20]. In this technique, the trial step d is decomposed into two orthogonal components: the normal component d^n to improve feasibility and the tangential component d_k^t to improve optimality. Each of components is computed by solving unconstrained trust-region subproblem.

- How to estimate the normal component d_k^n

The normal component d_k^n is computed by solving the following trust-region subproblem

$$\begin{array}{ll} minimize & \|h_k + (P_k \nabla h_k)^T d^n\|^2 \\ subject\ to & \|d^n\| \le \zeta \delta_k, \end{array} \tag{2.18}$$

for some $0 < \zeta < 1$. To solve the subproblem 2.18, we use a conjugate gradient method which is introduced by [38] and used by [21], see algorithm 2.1 in [21]. It is very cheap if the problem is large-scale and the Hessian is indefinite. By using the conjugate gradient method, the normal predicted decrease obtained by d_k^n is greater than or equal to a fraction of the normal predicted decrease obtained by the Cauchy step d_k^{ncp}. This means that

$$\|h_k\|^2 - \|h_k + (P_k \nabla h_k)^T d_k^n\|^2 \geq \vartheta_1 \{\|h_k\|^2 - \|h_k + (P_k \nabla h_k)^T d_k^{ncp}\|^2\}, \tag{2.19}$$

such that d_k^{ncp} is defined as follows

$$d^{ncp} = -\varphi_k^{ncp} P_k \nabla h_k h_k, \tag{2.20}$$

where the parameter φ_k^{ncp} is given by

$$\varphi_k^{ncp} = \begin{cases} \frac{\|P_k \nabla h_k h_k\|^2}{\|(P_k \nabla h_k)^T P_k \nabla h_k h_k\|^2} & \text{if } \frac{\|P_k \nabla h_k h_k\|^3}{\|(P_k \nabla h_k)^T P_k \nabla h_k h_k\|^2} \leq \delta_k \text{ ,} \\ & \text{and } \|(P_k \nabla h_k)^T P_k \nabla h_k h_k\| > 0, \\ \frac{\delta_k}{\|P_k \nabla h_k h_k\|} & \text{otherwise.} \end{cases} \tag{2.21}$$

Once d_k^n is obtained, we will evaluate $d_k^t = Z_k \bar{d}_k^t$ where Z_k is the matrix whose columns form a basis for the null space of $(P_k \nabla h_k)^T$.

- How to estimate the tangential component d_k^t

To obtain the tangential component d_k^t, we use the conjugate gradient method [21] to solve the following trust-region subproblem

$$\begin{array}{ll} minimize & [Z_k^T \nabla q_k(P_k d_k^n)]^T \bar{d}^t + \frac{1}{2} \bar{d}^{t^T} Z_k^T B_k Z_k \bar{d}^t \\ subject\ to & \|Z_k \bar{d}^t\| \leq \Delta_k, \end{array} \tag{2.22}$$

where $\nabla q_k(P_k d_k^n) = P_k \nabla_x \ell_k + B_k d_k^n$ and $\Delta_k = \sqrt{\delta_k^2 - \|d_k^n\|^2}$.

By using the conjugate gradient method, the tangential predicted decrease which is obtained by tangential step $\bar{d}_k^t$ is greater than or equal to a fraction of the tangential predicted decrease obtained by a tangential Cauchy step $\bar{d}_k^{tcp}$. This means that

$$q_k(P_k d_k^n) - q_k(P_k(d_k^n + Z_k \bar{d}_k^t)) \geq \vartheta_2 [q_k(P_k d_k^n) - q_k(P_k(d_k^n + Z_k \bar{d}_k^{tcp}))], \tag{2.23}$$

for some $0 < \vartheta_2 \leq 1$ and $\bar{d}_k^{tcp}$ is defined as follows

$$\bar{d}^{tcp} = -\varphi_k^{tcp} Z_k^T \nabla q_k(P_k d_k^n), \tag{2.24}$$

where the parameter φ_k^{tcp} is given by

$$\varphi_k^{tcp} = \begin{cases} \frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|^2}{(Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n)} & \text{if } \frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|^3}{(Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n)} \leq \Delta_k \text{ ,} \\ & \text{and } (Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n) > 0, \\ \frac{\Delta_k}{\|Z_k^T \nabla q_k(P_k d_k^n)\|} & \text{otherwise,} \end{cases} \tag{2.25}$$

where $\bar{B}_k = Z_k^T B_k Z_k$.

- How to estimate a parameter γ_k

Once obtaining d_k^t, we set $d_k = d_k^n + d_k^t$ and $x_{k+1} = x_k + P_k d_k$. To ensure $x_{k+1} \in int(\hat{\boldsymbol{G}})$, we need to evaluate the parameter γ_k. To do this, evaluate

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

and

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

Compute

$$\gamma_k = \min\{1, \min_i \{a_k^{(i)}, b_k^{(i)}\}\}. \tag{2.26}$$

Once the trial step $\gamma_k P_k d_k$ is evaluated, it needs to be tested to decide whether it will be accepted or not. To do this, we need to a merit function which is ties the objective function and the constraints in such a way that progress in the merit function means progress in solving problem. In our method, we use the following merit function which is introduced by [26] and known as an augmented Lagrange function

$$\Phi(x, \lambda; \rho) = \ell(x, \lambda) + \rho \|h(x)\|^2, \tag{2.27}$$

where $\ell(x, \lambda)$ is defined in 2.1 and $\rho > 0$ represents the penalty parameter.

- How to estimate λ_{k+1}

The Lagrange multiplier vector λ_{k+1} will be estimated as follows

$$minimize \ \ \|\nabla f_{u_{k+1}} + \nabla h_{k+1} \lambda\|^2. \tag{2.28}$$

To test whether the point (x_{k+1}, λ_{k+1}), will be accepted in the next iterate or no we need to define the following actual reduction $Ared_k$ and the predicted reduction $Pred_k$.

The actual reduction $Ared_k$ in the merit function 2.27 in moving from (x_k, λ_k) to $(x_k + \gamma_k P_k d_k, \lambda_{k+1})$ is defined as follows

$$Ared_k = \Phi(x_k, \lambda_k; \rho_k) - \Phi(x_k + \gamma_k P_k d_k, \lambda_{k+1}; \rho_k).$$

Also we can write the actual reduction $Ared_k$ as follows,

$$\begin{aligned} Ared_k &= \Phi(x_k, \lambda_k; \rho_k) - \Phi(x_k + \gamma_k P_k d_k, \lambda_{k+1}; \rho_k), \\ &= \ell(x_k, \lambda_k) - \ell(x_{k+1}, \lambda_k) - \Delta\lambda_k^T h_{k+1} + \rho_k[\|h_k\|^2 - \|h_{k+1}\|^2], \end{aligned} \tag{2.29}$$

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

The predicted reduction $Pred_k$ is defined as follows

$$\begin{aligned} Pred_k &= -(P_k \nabla_x \ell(x_k, \lambda_k))^T \gamma_k d_k - \frac{1}{2}\gamma_k^2 d_k^T B_k d_k - \Delta\lambda_k^T (h_k + (P_k \nabla h_k)^T \gamma_k d_k) \\ &\quad + \rho_k[\|h_k\|^2 - \|h_k + (P_k \nabla h_k)^T \gamma_k d_k\|^2]. \end{aligned} \tag{2.30}$$

Since $q_k(\gamma_k P_k d_k) = \ell_k + (P_k \nabla_x \ell_k)^T \gamma_k d_k + \frac{1}{2}\gamma_k^2 d_k^T B_k d_k$, then $Pred_k$ can be written as follows,

$$Pred_k = q_k(0) - q_k(\gamma_k P_k d_k) - \Delta\lambda_k^T (h_k + (P_k \nabla h_k)^T \gamma_k d_k) + \rho_k[\|h_k\|^2 - \|h_k + (P_k \nabla h_k)^T \gamma_k d_k\|^2]. \tag{2.31}$$

- How to update the penalty parameter ρ_k

To ensure $Pred_k$ is strictly positive, we use the following scheme to update the positive penalty parameter ρ_k

Algorithm 2.1. *: (Updating the penalty parameter ρ_k)*
Set $\rho_{k+1} = \rho_k$.
If

$$Pred_k \geq \frac{\rho_k}{2}[\|h_k\|^2 - \|h_k + (P_k \nabla h_k)^T \gamma_k d_k\|^2], \tag{2.32}$$

then set

$$\rho_k = \frac{2[q_k(\gamma_k P_k d_k) - q_k(0) + \Delta\lambda_k^T(h_k + (P_k \nabla h_k)^T \gamma_k d_k)]}{\|h_k\|^2 - \|h_k + (P_k \nabla h_k)^T \gamma_k d_k\|^2} + c_0. \tag{2.33}$$

End if.

- How to test the step $\gamma_k P_k d_k$ and update δ_k

To decide the trial step $\gamma_k P_k d_k$ will be accepted in the next iteration or no, we use the following algorithm.

Algorithm 2.2. *: (Testing the step $\gamma_k P_k d_k$ and updating δ_k)*
Step 0. *Choose $0 < \tau_1 < \tau_2 < 1$, $0 < \beta_1 < 1 < \beta_2$, and $\delta_{min} \leq \delta_0 \leq \delta_{max}$.*
Step 1. *While $\frac{Ared_k}{Pred_k} < \tau_1$ or $Pred_k \leq 0$.*
Do not accept the step and set $\delta_k = \beta_1 \|d_k\|$.
Compute a new trial step.
End while.
Step 2. *If $\tau_1 \leq \frac{Ared_k}{Pred_k} < \tau_2$.*
Accept the step: $x_{k+1} = x_k + \gamma_k P_k d_k$.
Set $\delta_{k+1} = \max(\delta_k, \delta_{min})$.
End if.
Step 3. *If $\frac{Ared_k}{Pred_k} \geq \tau_2$.*
Accept the step: $x_{k+1} = x_k + \gamma_k P_k d_k$.
Set $\delta_{k+1} = \min\{\delta_{max}, \max\{\delta_{min}, \beta_2 \delta_k\}\}$.
End if.

Finally, the algorithm is stopped when either $\|Z_k^T P_k \nabla_x \ell_k\| + \|h_k\| \leq \varepsilon_1$, for some $\varepsilon_1 > 0$ or $\|d_k\| \leq \varepsilon_2$ for some $\varepsilon_2 > 0$.

Main steps of the trust-region algorithm for solving subproblem 2.17 are summarized in the following algorithm.

Algorithm 2.3. *(Trust-region algorithm)*
Step 0. *Starting with $x_0 \in int(\hat{\boldsymbol{G}})$. Evaluate λ_0, P_0, and β_0. Set $\rho_0 = 1$ and $c_0 = 0.1$.*
Choose $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, and set $k = 0$.
Step 1. *If $\|Z_k^T P_k \nabla_x \ell_k\| + \|h_k\| \leq \varepsilon_1$, then stop.*
Step 2. *(To compute d_k)*

a) Compute d_k^n by solving trust-region subproblem 2.18.
b) Compute $\bar{d}_k^t$ by solving trust-region subproblem 2.22.
c) Set $d_k = d_k^n + Z_k \bar{d}_k^t$.

Step 3. *If $||d_k|| \leq \varepsilon_2$, then stop.*
Step 4. *Compute γ_k using equation 2.26.*
Step 5. *Update λ_{k+1} using subproblem 2.28.*
Step 6. *Update the penalty parameter using scheme 2.1.*
Step 7. *Test the step $\gamma_k P_k d_k$ and update δ_k by using algorithm 2.2.*
Step 8. *Compute P_{k+1} and α_{k+1} using definitions 2.8 and 2.13 respectively.*
Step 9. *Set $k = k + 1$ and go to Step 1.*

Main steps for solving NBLP problem1.1 are summarized in the following algorithm.

Algorithm 2.4. *(Interior-point trust-region (IPTR) algorithm)*
Step 1. *Adding slack variables to inequalities in NBLP problem1.1 and convert it to problem 1.2.*
Step 2. *By KKT optimality conditions for the lower-level problem, NBLP problem 1.2 is equivalent to the one level problem 1.3 which can be written in the form 1.4.*
Step 3. *Using Newton's method and Das strategy to transform problem 1.4 to subproblem 2.16.*
Step 4. *Using trust-region algorithm 2.3 to solve subproblem 2.16.*

The following section is devoted to global convergence analysis for IPTR algorithm 2.4.

3. Global convergence theory

We state the general assumption under which the global convergence theory for IPTR algorithm 2.4 is proved.

3.1. A general assumptions

Let Ω be a convex subset of $\Re^n$ that contains all points $x_k \in int(\hat{\boldsymbol{G}})$ and $(x_k + \gamma_k P_k d_k) \in int(\hat{\boldsymbol{G}})$. On the set Ω we state the following general assumptions under which the global convergence theory of IPTR algorithm is proved

[GS_1.] The functions $f_u(x), h(x) \in C^2$ for all $x \in \Omega$.
[GS_2.] The matrix $P_k \nabla h_k$ has full column rank.
[GS_3.] All of $f_u(x)$, $\nabla f_u(x)$, $\nabla^2 f_u(x)$, $h(x)$, $\nabla h(x)$, $\nabla^2 h_i(x)$ for $i = 1, ..., m$ and $(P_k \nabla H_k)((P_k \nabla h_k)^T (P_k \nabla h_k))^{-1}$ are uniformly bounded in Ω.
[GS_4.] The sequence of Lagrange multiplier vectors $\{\lambda_k\}$ is bounded.
[GS_5.] The sequence of approximate Hessian matrices $\{H_k\}$ is bounded.

An immediate consequence of the above general assumptions is that the existence of positive constant b_1, such that

$$||Z_k^T B_k|| \leq b_1, \qquad ||Z_k^T B_k Z_k|| \leq b_1. \tag{3.1}$$

3.2. Technical lemmas

In this section, we introduce some important results which are needed in the subsequent proof.

The following lemma shows how accurate the definition of $Pred_k$ is as an approximation to $Ared_k$.

Lemma 3.1. *Under assumptions GS_1-GS_5, there exists a positive constant K_1, such that*

$$|Ared_k - Pred_k| \leq K_1 \rho_k \gamma_k ||d_k||^2. \tag{3.2}$$

Proof. From Equations 2.29, 2.30, and using the inequality of Cauchy-Schwarz, we have

$$\begin{aligned}|Ared_k - Pred_k| \leq\ & \frac{1}{2}|\gamma_k^2 P_k d_k^T[H_k - \nabla^2\ell(x_k + \xi_1\gamma_k P_k d_k)]P_k d_k| \\ & + \frac{1}{2}|\Delta\lambda_k\gamma_k^2 P_k d_k^T\nabla^2 h(x_k + \xi_2\gamma_k P_k d_k)P_k d_k| \\ & + \frac{1}{2}|\gamma_k^2 d_k^T diag(\nabla_x\ell_k)diag(\theta_k)d_k| \\ & + |\Delta\lambda_k P_k[\nabla h_k - \nabla h(x_k + \xi_2\gamma_k P_k d_k)]^T\gamma_k d_k| \\ & + 2\rho_k|P_k[(\nabla h_k - \nabla h(x_k + \xi_2\gamma_k P_k d_k))h_k]^T\gamma_k d_k| \\ & + \rho_k|\gamma_k^2 P_k d_k^T[\nabla h_k\nabla h_k^T - \nabla h(x_k + \xi_2\gamma_k P_k d_k)\nabla h(x_k + \xi_2\gamma_k P_k d_k)^T]P_k d_k|,\end{aligned}$$

for some ξ_1 and $\xi_2 \in (0, 1)$. Using the general assumptions $GS_1 - GS_5$ and $0 < \gamma_k \leq 1$, we have

$$|Ared_k - Pred_k| \leq \gamma_k[\kappa_1\|d_k\|^2 + \kappa_2\rho_k\|d_k\|^3 + \kappa_3\rho_k\|d_k\|^2\|h_k\|], \tag{3.3}$$

where κ_1, κ_2, and κ_3 are positive constants. Since $\rho_k \geq 0$, $\|d_k\| \leq \delta_{max}$, and $\|h_k\|$ is uniformly bounded, then inequality 3.2 hold. □

The following lemma obviously that the normal predicted reduction at any iteration k, is at least equal to the decrease in the 2-norm of the linearized constrained by the Cauchy step

Lemma 3.2. *Under assumptions GS_1-GS_5, there exists a constant $K_2 > 0$, such that*

$$Pred_k \geq \frac{K_2\gamma_k\rho_k}{2}\|h_k\|\min\{\|h_k\|, \delta_k\}. \tag{3.4}$$

Proof. Since d_k^n is obtained by approximating the solution of subproblem 2.18 using the conjugate gradient method [21], then the fraction of Cauchy decrease condition 2.19 is hold. We will consider two cases:
Firstly, if $d^{ncp} = -\frac{\delta_k}{\|P_k\nabla h_k h_k\|}(P_k\nabla h_k h_k)$ and $\|\delta_k\|(P_k\nabla h_k)^T P_k\nabla h_k h_k\|^2 \leq \|P_k\nabla h_k h_k\|^3$ then

$$\begin{aligned}\|h_k\|^2 - \|h_k + (P_k\nabla h_k)^T d_k^{ncp}\|^2 &= -2(P_k\nabla h_k h_k)^T d_k^{ncp} - d_k^{ncp^T}(P_k\nabla h_k)(P_k\nabla h_k)^T d_k^{ncp} \\ &= 2\delta_k\|P_k\nabla h_k h_k\| - \frac{\delta_k^2\|(P_k\nabla h_k)^T P_k\nabla h_k h_k\|^2}{\|P_k\nabla h_k h_k\|^2} \\ &\geq 2\delta_k\|P_k\nabla h_k h_k\| - \delta_k\|P_k\nabla h_k h_k\| \\ &\geq \delta_k\|P_k\nabla h_k h_k\|. \end{aligned} \tag{3.5}$$

Secondly, if $d^{ncp} = -\frac{\|P_k\nabla h_k h_k\|^2}{\|(P_k\nabla h_k)^T P_k\nabla h_k h_k\|^2}(P_k\nabla h_k h_k)$ and $\delta_k\|(P_k\nabla h_k)^T P_k\nabla h_k h_k\|^2 \geq \|P_k\nabla h_k h_k\|^3$, then

$$\begin{aligned}\|h_k\|^2 - \|h_k + (P_k\nabla h_k)^T d_k^{ncp}\|^2 &= -2(P_k\nabla h_k h_k)^T d_k^{ncp} - d_k^{ncp^T}(P_k\nabla h_k)(P_k\nabla h_k)^T d_k^{ncp} \\ &= \frac{2\|P_k\nabla h_k h_k\|^4}{\|(P_k\nabla h_k)^T P_k\nabla h_k h_k\|^2} \\ &\quad - \frac{\|P_k\nabla h_k h_k\|^4}{\|(P_k\nabla h_k)^T P_k\nabla h_k h_k\|^2} \\ &= \frac{\|P_k\nabla h_k h_k\|^4}{\|(P_k\nabla h_k)^T P_k\nabla h_k h_k\|^2}\end{aligned}$$

$$\geq \frac{||P_k \nabla h_k h_k||^2}{||P_k \nabla h_k (P_k \nabla h_k)^T||}. \tag{3.6}$$

Using assumption GS_2, we have

$$||P_k \nabla h_k h_k|| \geq \frac{||h_k||}{||((P_k \nabla h_k)^T P_k \nabla h_k)^{-1} (P_k \nabla h_k)^T||}.$$

Then, from the above inequality, inequalities 3.5, 3.6, and using assumption GS_3, we have

$$||h_k||^2 - ||h_k + (P_k \nabla h_k)^T d_k^{ncp}||^2 \geq K_2 ||h_k|| \min\{||h_k||, \delta_k\}.$$

From the above inequality and 2.19, we have

$$||h_k||^2 - ||h_k + (P_k \nabla h_k)^T d_k^{n}||^2 \geq K_2 ||h_k|| \min\{||h_k||, \delta_k\}. \tag{3.7}$$

Since $0 < \gamma_k \leq 1$, then we have

$$||h_k||^2 - ||h_k + (P_k \nabla h_k)^T \gamma_k d_k^{n}||^2 \geq \gamma_k [||h_k||^2 - ||h_k + (P_k \nabla h_k)^T d_k^{n}||^2].$$

From inequality 3.7 and the above inequality, we have

$$||h_k||^2 - ||h_k + (P_k \nabla h_k)^T \gamma_k d_k^{n}||^2 \geq K_2 \gamma_k ||h_k|| \min\{||h_k||, \delta_k\}. \tag{3.8}$$

From inequalities 2.32 and 3.8 we have

$$Pred_k \geq \frac{K_2 \gamma_k \rho_k}{2} ||h_k|| \min\{||h_k||, \delta_k\}.$$

Lemma 3.3. *Under assumptions GS_1-GS_5, there exists a positive constant K_3, such that*

$$[q_k(\gamma_k P_k d_k^n) - q_k(\gamma_k P_k d_k)] \geq K_3 \gamma_k ||Z_k^T \nabla q_k(P_k d_k^n)|| \min\{\frac{||Z_k^T \nabla q_k(P_k d_k^n)||}{||\bar{B}||}, \Delta_k\}. \tag{3.9}$$

Proof. Since the conjugate gradient method is used to solve subproblem 2.22 to obtain an approximate solution for $\bar{d}_k^t$, then the fraction of Cauchy decrease condition 2.23 is hold and we will consider two cases:

Firstly, if $\bar{d}_k^{tcp} = -\frac{\Delta_k}{||Z_k^T \nabla q_k(P_k d_k^n)||}(Z_k^T \nabla q_k(P_k d_k^n))$ and $\Delta_k (Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n) \leq ||Z_k^T \nabla q_k(P_k d_k^n)||^3$, then

$$\begin{aligned} q_k(P_k d_k^n) - q_k(P_k(d_k^n + Z_k \bar{d}_k^{tcp})) &= q_k(P_k d_k^n) - q_k(P_k(d_k^n + Z_k \bar{d}_k^{tcp})) \\ &= -(Z_k^T \nabla q_k(P_k d_k^n))^T \bar{d}_k^{tcp} - \frac{1}{2} \bar{d}_k^{tcp^T} \bar{B}_k \bar{d}_k^{tcp} \\ &= \Delta_k ||Z_k^T \nabla q_k(P_k d_k^n)|| \\ &\quad - \frac{\Delta_k^2}{2||Z_k^T \nabla q_k(P_k d_k^n)||^2} [(Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n)] \end{aligned}$$

$$
\begin{aligned}
&\geq \Delta_k \|Z_k^T \nabla q_k(P_k d_k^n)\| - \frac{1}{2}\Delta_k \|Z_k^T \nabla q_k(P_k d_k^n)\| \\
&\geq \frac{1}{2}\Delta_k \|Z_k^T \nabla q_k(P_k d_k^n)\|. \qquad (3.10)
\end{aligned}
$$

Secondly, if $\bar{d}_k^{tcp} = -\frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|^2}{Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n)} Z_k^T \nabla q_k(P_k d_k^n)$ and $\Delta_k (Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n) \geq \|Z_k^T \nabla q_k(P_k d_k^n)\|^3$, then

$$
\begin{aligned}
q_k(P_k d_k^n) - q_k(P_k(d_k^n + Z_k \bar{d}_k^{tcp})) &= q_k(P_k d_k^n) - q_k(P_k(d_k^n + Z_k \bar{d}_k^{tcp})) \\
&= -(Z_k^T \nabla q_k(P_k d_k^n))^T \bar{d}_k^{tcp} - \frac{1}{2} \bar{d}_k^{tcp^T} \bar{B}_k \bar{d}_k^{tcp} \\
&= \frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|^4}{(Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n)} \\
&\quad - \frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|^4}{2(Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n)} \\
&= \frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|^4}{2(Z_k^T \nabla q_k(P_k d_k^n))^T \bar{B}_k Z_k^T \nabla q_k(P_k d_k^n)} \\
&\geq \frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|^2}{2\|\bar{B}_k\|}. \qquad (3.11)
\end{aligned}
$$

From inequalities 3.10, 3.11, and using necessary assumptions, we have

$$
q_k(P_k d_k^n) - q_k(P_k(d_k^n + Z_k \bar{d}_k^{tcp})) \geq K_3 \|Z_k^T \nabla q_k(P_k d_k^n)\| \min\{\frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|}{\|\bar{B}\|}, \Delta_k\}.
$$

From condition 2.23 and the above inequality, we have

$$
q_k(P_k d_k^n) - q_k(P_k(d_k^n + Z_k \bar{d}_k^t)) \geq K_3 \|Z_k^T \nabla q_k(P_k d_k^n)\| \min\{\frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|}{\|\bar{B}\|}, \Delta_k\}. \qquad (3.12)
$$

Since $0 < \gamma_k \leq 1$, then we have

$$
q_k(\gamma_k P_k d_k^n) - q_k(\gamma_k P_k(d_k^n + Z_k \bar{d}_k^t)) \geq \gamma_k [q_k(P_k d_k^n) - q_k(P_k(d_k^n + Z_k \bar{d}_k^t))].
$$

From the above inequality and inequality 3.12, we have

$$
q_k(\gamma_k P_k d_k^n) - q_k(\gamma_k P_k d_k) \geq K_3 \gamma_k \|Z_k^T \nabla q_k(P_k d_k^n)\| \min\{\frac{\|Z_k^T \nabla q_k(P_k d_k^n)\|}{\|\bar{B}\|}, \Delta_k\}.
$$

This completes the proof.

The following lemma is needed in many forthcoming lemmas. In what follows, we use implicitly that $\nabla h_k d_k^n = \nabla h_k d_k$.

Lemma 3.4. *Under assumptions GS_1-GS_5, there exists a positive constant K_4, such that*

$$
q_k(0) - q_k(\gamma_k P_k d_k^n) - \Delta\lambda_k^T (h_k + (P_k \nabla h_k)^T \gamma_k d_k) \geq -K_4 \gamma_k \|h_k\|. \qquad (3.13)
$$

Proof. Since d_k^n is normal to the tangent space, then we have

$$\begin{aligned} \|d_k^n\| &= \|(P_k\nabla h_k)[(P_k\nabla h_k)^T(P_k\nabla h_k)]^{-1}(P_k\nabla h_k)^T d_k\| \\ &= \|(P_k\nabla h_k)[(P_k\nabla h_k)^T(P_k\nabla h_k)]^{-1}[h_k + (P_k\nabla h_k)^T d_k - h_k]\| \\ &\le \|(P_k\nabla h_k)[(P_k\nabla h_k)^T(P_k\nabla h_k)]^{-1}\|[\|h_k + (P_k\nabla h_k)^T d_k\| + \|h_k\|]. \end{aligned}$$

Using the fact that $\|h_k + (P_k\nabla h_k)^T d_k\| \le \|h_k\|$, we have

$$\|d_k^n\| \le 2\|(P_k\nabla h_k)[(P_k\nabla h_k)^T(P_k\nabla h_k)]^{-1}\|\|h_k\|.$$

From above inequality and necessary assumptions, we have

$$\|d_k^n\| \le \kappa_4\|h_k\|. \tag{3.14}$$

Since

$$\begin{aligned} q_k(0) - q_k(\gamma_k P_k d_k^n) - \Delta\lambda_k^T(h_k + (P_k\nabla h_k)^T\gamma_k d_k) &= -(P_k\nabla_x\ell_k)^T\gamma_k d_k^n - \frac{1}{2}\gamma_k^2 d_k^{nT} B_k d_k^n \\ &\quad -\Delta\lambda_k^T(h_k + (P_k\nabla h_k)^T\gamma_k d_k) \\ &\ge -\gamma_k\|P_k\nabla_x\ell_k\|\|d_k^n\| - \frac{1}{2}\gamma_k^2\|B_k\|\|d_k^n\|^2 \\ &\quad -\|\Delta\lambda_k\|\|(h_k + (P_k\nabla h_k)^T\gamma_k d_k^n)\| \\ &\ge -\gamma_k[\|P_k\nabla_x\ell_k\| + \frac{1}{2}\gamma_k\|B_k\|\|d_k^n\|]\|d_k^n\| - \|\Delta\lambda_k^T\|\|h_k\|. \end{aligned}$$

From the above inequality and inequality 3.14 and using the fact that $\|d_k^n\| \le \delta_{max}$, we have

$$q_k(0) - q_k(\gamma_k P_k d_k^n) - \Delta\lambda_k^T(h_k + (P_k\nabla h_k)^T\gamma_k d_k) \ge -K_4\gamma_k\|h_k\|.$$

This completes the proof.

Lemma 3.5. *Under assumptions GS_1-GS_5,*

$$\begin{aligned} Pred_k \ge\ & K_3\gamma_k\|Z_k^T\nabla q_k(P_k d_k^n)\| \min\{\frac{\|Z_k^T\nabla q_k(P_k d_k^n)\|}{\|\bar{B}\|},\ \Delta_k\} \\ & -K_4\gamma_k\|h_k\| + \rho_k[\|h_k\|^2 - \|h_k + (P_k\nabla h_k)^T\gamma_k d_k\|^2]. \end{aligned} \tag{3.15}$$

Proof. From Equation 2.31, we have

$$\begin{aligned} Pred_k &= q_k(0) - q_k(\gamma_k P_k d_k) - \Delta\lambda_k^T(h_k + (P_k\nabla h_k)^T\gamma_k d_k) + \rho_k[\|h_k\|^2 - \|h_k + (P_k\nabla h_k)^T\gamma_k d_k\|^2] \\ &= [q_k(\gamma_k P_k d_k^n) - q_k(\gamma_k P_k d_k)] + [q_k(0) - q_k(\gamma_k P_k d_k^n) - \Delta\lambda_k^T(h_k + (P_k\nabla h_k)^T\gamma_k d_k)] \\ &\quad +\rho_k[\|h_k\|^2 - \|h_k + (P_k\nabla h_k)^T\gamma_k d_k\|^2]. \end{aligned}$$

Using inequalities 3.9 and 3.13, we obtain the desired result.

The following lemma shows that, if $\|Z_k^T P_k\nabla_x\ell_k\| \ge \varepsilon_1$ and $\|h_k\| \le \eta\delta_{k^i}$ at any trial iteration k^i, then the penalty parameter ρ_k is not increased.

Lemma 3.6. *Under assumptions $GS_1 - GS_5$, if $\|Z_k^T P_k \nabla_x \ell_k\| \geq \varepsilon_1$ and $\|h_k\| \leq \eta\delta_{k^i}$ at any trial iteration k^i, then there exists a positive constant K_5, such that*

$$Pred_{k^i} \geq K_5 \gamma_{k^i} \delta_{k^i} + \rho_{k^i}\{\|h_k\|^2 - \|h_k + (P_k \nabla h_k)^T \gamma_{k^i} d_{k^i}\|^2\}, \tag{3.16}$$

where η is given by

$$0 < \eta \leq \min\left\{\frac{\sqrt{3}}{2\kappa_4}, \frac{\varepsilon_1}{2b_1\kappa_4\delta_{\max}}, \frac{K_3\varepsilon_1}{8K_4}\min\{\frac{\varepsilon_1}{b_1\delta_{\max}}, 1\}\right\}.$$

Proof. Since $\|Z_k^T P_k \nabla_x \ell_k\| \geq \varepsilon_1$ and $\|h_k\| \leq \eta\delta_{k^i}$, and using inequalities 3.1 and 3.14, we have

$$\begin{aligned} \|Z_k^T \nabla q_k(P_k d^n_{k^i})\| &= \|Z_k^T (P_k \nabla_x \ell_k + B_k d^n_{k^i})\| \\ &\geq \|Z_k^T P_k \nabla_x \ell_k\| - \|Z_k^T B_k d^n_{k^i}\| \\ &\geq \varepsilon_1 - b_1\kappa_4\|h_{k^i}\| \geq \varepsilon_1 - b_1\kappa_4\eta\delta_{k^i}. \end{aligned}$$

But $\eta \leq \frac{\varepsilon_1}{2b_1\kappa_4\delta_{\max}}$, hence

$$\|Z_k^T \nabla q_k(P_k d^n_{k^i})\| \geq \frac{1}{2}\varepsilon_1. \tag{3.17}$$

From inequality 3.14, assumption $\|h_k\| \leq \eta\delta_{k^i}$, and $\eta \leq \frac{\sqrt{3}}{2\kappa_4}$, then we have $\|d^n_{k^i}\| \leq \kappa_4\eta\delta_{k^i} \leq \kappa_4\frac{\sqrt{3}}{2\kappa_4}\delta_{k^i} = \frac{\sqrt{3}}{2}\delta_{k^i}$. Since $\Delta_{k^i} = \sqrt{\delta^2_{k^i} - \|d^n_{k^i}\|^2}$, then $\Delta_{k^i} \geq \frac{1}{2}\delta_{k^i}$. Hence, from inequalities 3.15 and 3.17, we have

$$Pred_{k^i} \geq \frac{K_3\gamma_{k^i}\varepsilon_1}{4}\min\{\frac{\varepsilon_1}{b_1\delta_{\max}}, 1\}\delta_{k^i} - K_4\gamma_{k^i}\eta\delta_{k^i} + \rho_{k^i}[\|h_k\|^2 - \|h_k + (P_k\nabla h_k)^T\gamma_{k^i}d_{k^i}\|^2].$$

But $\eta \leq \frac{K_3\varepsilon_1}{8K_4}\min\{\frac{\varepsilon_1}{b_1\delta_{\max}}, 1\}$, hence

$$Pred_{k^i} \geq \frac{K_3\gamma_{k^i}\varepsilon_1}{8}\min\{\frac{\varepsilon_1}{b_1\delta_{\max}}, 1\}\delta_{k^i} + \rho_{k^i}[\|h_k\|^2 - \|h_k + (P_k\nabla h_k)^T\gamma_{k^i}d_{k^i}\|^2].$$

The result follows if we take $K_5 = \frac{K_3\varepsilon_1}{8}\min\{\frac{\varepsilon_1}{b_1\delta_{\max}}, 1\}$.

The following lemma shows that, at any iteration k, we can find an acceptable step after finite number of trials, or equivalently, the condition $Ared_{k^j}/Pred_{k^j} \geq \tau_1$ will be satisfied for some finite j.

Lemma 3.7. *Under assumptions $GS_1 - GS_5$, if $\|h_k\| > \varepsilon_1$, where $\varepsilon_1 > 0$, then $Ared_{k^j}/Pred_{k^j} \geq \tau_1$ will be satisfied for some finite j.*

Proof. From inequalities 3.2, 3.4, and assumption $\|h_k\| > \varepsilon_1$, we have

$$\left|\frac{Ared_k}{Pred_k} - 1\right| = \frac{|Ared_k - Pred_k|}{Pred_k} \leq \frac{2K_1\gamma_k\delta_k^2}{K_2\gamma_k\varepsilon_1\min\{\varepsilon_1, \delta_k\}}.$$

If the trial step d_{k^j} gets rejected, then δ_{k^j} becomes small and hence we have

$$\left|\frac{Ared_{k^j}}{Pred_{k^j}} - 1\right| \leq \frac{2K_1\delta_{k^j}}{K_2\varepsilon_1}.$$

That is the acceptance rule will be met after finite number of trials (i.e.,for finite j) and this completes the proof.

Lemma 3.8. *Under assumptions $GS_1 - GS_5$ and at any iteration k, if*

$$||d_{k^j}|| \leq \min\{\frac{(1-\tau_1)K_2}{4K_1}, 1\}||h_k||, \tag{3.18}$$

at the j^{th} trial step, then the step must be accepted.

Proof. Assume that inequality 3.18 holds and the step d_{k^j} is rejected. From the way of updating trust-region radius which is clarified in Algorithm 2.2 we have

$$(1-\tau_1) < \frac{|Ared_{k^j} - Pred_{k^j}|}{Pred_{k^j}}.$$

From the above inequality and using inequalities 3.2, 3.4, and 3.18 we have

$$(1-\tau_1) < \frac{|Ared_{k^j} - Pred_{k^j}|}{Pred_{k^j}} < \frac{2K_1||d_{k^j}||^2}{K_2||h_k||||d_{k^j}||} \leq \frac{1}{2}(1-\tau_1).$$

This is a contradiction with the assumption d_{k^j} was rejected. Hence the step must be accepted.

Lemma 3.9. *Under assumptions $GS_1 - GS_5$ and for all trail iterates j of any iteration k we have*

$$\delta_{k^j} \geq \min\{\frac{\delta_{\min}}{b_2}, \beta_1\frac{(1-\tau_1)K_2}{4K_1}, \beta_1\}||h_k||, \tag{3.19}$$

where $b_2 = \sup_{x\in\Omega} ||h_k||$.

Proof. Consider any trial iterate k^j, if $j = 1$, then the step is accepted and hence

$$\delta_{k^j} = \delta_{k^1} \geq \delta_{\min} \geq \frac{\delta_{\min}}{b_2}||H_k||, \tag{3.20}$$

such that $b_2 = \sup_{x\in\Omega} ||h_k||$.

Now, if $j > 1$, then there exists at least one rejected trial step. For all rejected trial steps, we have from Lemma 3.8,

$$||d_{k^i}|| > \min\{\frac{(1-\tau_1)K_2}{4K_1}, 1\}||h_k||,$$

for all $i = 1, 2, ...j-1$. Since d_k^i is rejected trial step, then from the way of updating the radius of trust-region, we have

$$\delta_{k^j} = \beta_1||d_{k^{j-1}}|| > \beta_1 \min\{\frac{(1-\tau_1)K_2}{4K_1}, 1\}||h_k||.$$

From the above inequality and inequality 3.20, we obtain the desired results.

The following lemma show that the sequence of trust-region radii $\{\delta_{k^j}\}$ is bounded away from zero if $\{||h_k||\}$ is bounded away from zero.

Lemma 3.10. *Under assumptions $GS_1 - GS_5$, if $||h_k|| \geq \varepsilon_1$ where $\varepsilon_1 > 0$, then there exists a constant $K_6 > 0$ such that*

$$\delta_{k^j} > K_6, \tag{3.21}$$

for all trial iterates j of any iteration k.

Proof. From Lemma 3.9 and the condition $\|h_k\| \geq \varepsilon_1$, the proof follows directly by taking $K_6 = \min\{\frac{\delta_{\min}}{b_2}, \beta_1 \frac{(1-\tau_1)K_2}{4K_1}, \beta_1\}\varepsilon_1$.

Lemma 3.11. *Under assumptions $GS_1 - GS_5$, there exists a subsequence $\{k_i\}$ of the iteration sequence at which ρ_k is increased such that at any trial steps j of any iteration $k \in \{k_i\}$, we have*

$$\rho_{k^j}\|h_k\| \leq K_7. \tag{3.22}$$

where K_7 is a positive constant.

Proof. At any trial steps j of any iteration k, if ρ_{k^j} is increased, then from equation 2.33, we have

$$\begin{aligned}\frac{\rho_{k^j}}{2}[\|h_k\|^2 - \|h_k \quad + \quad & (P_k \nabla h_k)^T \gamma_{k^j} d_{k^j}\|^2] = [q_k(P_k \gamma_{k^j} d_{k^j}) - q_k(P_k \gamma_{k^j} d^n_{k^j})] \\ & + [q_k(P_k \gamma_{k^j} d^n_{k^j}) - q_k(0) + \Delta\lambda^T_{k^j}(h_k + (P_k \nabla h_k)^T \gamma_{k^j} d^n_{k^j})] \\ & + \frac{c_0}{2}[\|h_k\|^2 - \|h_k + (P_k \nabla h_k)^T \gamma_{k^j} d_{k^j}\|^2].\end{aligned}$$

Applying inequality 3.8 on the left hand side and inequalities 3.9, 3.13, and 3.14 on the right hand side, we have

$$\begin{aligned}\frac{K_2 \rho_{k^j} \gamma_{k^j}}{2}\|h_k\| \min\{\|h_k\|, \delta_{k^j}\} \quad \leq \quad & -K_3 \gamma_{k^j} \|Z_k{}^T \nabla q_k(P_k d^n_{k^j})\| \min\{\frac{\|Z_k{}^T \nabla q_k(P_k s^n_{k^j})\|}{\|\bar{B}\|}, \Delta_{k^j}\} \\ & + K_4 \gamma_{k^j}\|h_k\| + c_0 \gamma_{k^j} \|P_k \nabla h_k h_k\| \|d^n_{k^j}\| \\ & + \frac{c_0 \gamma^2_{k^j}}{2}\|(P_k \nabla h_k)^T\|^2 \|d^n_{k^j}\|^2, \\ \leq \quad & [K_4 + c_0 \kappa_4 \|P_k \nabla h_k h_k\| + \frac{c_0 \kappa_4 \gamma_{k^j}}{2}\|(P_k \nabla h_k)^T\|^2 \|d^n_{k^j}\|] \gamma_{k^j}\|h_k\|.\end{aligned}$$

From assumptions GS_2, GS_3, and using the fact that $\|d^n_{k^j}\| \leq \delta_{k^j} \leq \delta_{max}$, we have

$$\rho_{k^j}\|h_k\| \min\{\|h_k\|, \delta_{k^j}\} \leq \tilde{K}_7 \|h_k\|. \tag{3.23}$$

From inequalities 3.19 and 3.23, there exists a constant $K_7 > 0$ such that

$$\rho_{k^j}\|h_k\| \leq K_7,$$

at any trial steps j for any iteration $k \in \{k_i\}$.

In the following lemma we will prove that the sequence $\{\|h_k\|\}$ is not bounded away from zero when $\{\rho_k\}$ unbounded sequence.

Lemma 3.12. *Under assumptions $GS_1 - AS_6$, there exists a subsequence $\{k_i\}$ of the iteration sequence at which ρ_k is increased such that*

$$\lim_{k_i \to \infty} \|h_{k_i}\| = 0. \tag{3.24}$$

Proof: From Lemma 3.11 and the assumption ρ_k is increased, we obtain the desired result.

In the following section, we prove the main global convergence results for IPTR algorithm 2.4.

3.3. Fundamental convergence theorem

In the following theorem we prove that the sequence of the iterates generated by algorithm 2.4 converges to the feasible set.

Theorem 3.1. *Under assumptions $GS_1 - GS_5$, the sequence of iterates generated by IPTR algorithm satisfies*

$$\lim_{k\to\infty} \|h_k\| = 0.$$

Proof. The proof of this theorem is by contradiction, so we assume that $\limsup_{k\to\infty} \|h_k\| \geq \varepsilon_1$ where $\varepsilon_1 > 0$. This implies the existence an infinite subsequence of indices $\{k_j\}$ indexing iterates that satisfy $\|h_k\| \geq \frac{\varepsilon_1}{2}$, for all $k \in \{k_j\}$. From Lemma 3.7, there exists a finite sequence of acceptable steps. Without lose of generality, we assume all members of the sequence $\{k_j\}$ are acceptable iterates. Now we will consider two cases:

Firstly, if the sequence of the penalty parameter $\{\rho_k\}$ is unbounded, then there exists a subsequence $\{k_i\}$ of the iteration sequence at which ρ_k is increased. Using Lemma 3.12, we have $\lim_{k_i\to\infty} \|h_{k_i}\| = 0$. Therefore, there are no common elements between $\{k_i\}$ and $\{k_j\}$ at iteration k which is sufficiently large. From inequality 3.4 and Lemma 3.10, we have

$$\frac{Ared_k}{\rho_k} \geq \tau_1 \frac{Pred_k}{\rho_k} \geq \tau_1 \frac{\varepsilon_1 K_2 \gamma_k}{4} \min[\frac{\varepsilon_1}{2}, \delta_k] \geq \tau_1 \frac{\varepsilon_1 K_2 \gamma_k}{4} \min[\frac{\varepsilon_1}{2}, \bar{K}_6], \tag{3.25}$$

for all $k \in \{k_j\}$, such that $\bar{K}_6 = \frac{\varepsilon_1}{2} \min\{\frac{\delta_{\min}}{b_2}, \beta_1 \frac{(1-\tau_1)K_2}{2K_1}, \beta_1\}$. Since

$$\begin{aligned} Ared_k &= \Phi(x_k, \lambda_k; \rho_k) - \Phi(x_k + \gamma_k P_k d_k, \lambda_{k+1}; \rho_k), \\ &= \ell(x_k, \lambda_k) - \ell(x_{k+1}, \lambda_{k+1}) + \rho_k[\|h_k\|^2 - \|h_{k+1}\|^2], \end{aligned}$$

then from 3.25 we have

$$\frac{Ared_k}{\rho_k} = \frac{\ell_k - \ell_{k+1}}{\rho_k} + \|h_k\|^2 - \|h_{k+1}\|^2 \geq \tau_1 \frac{\varepsilon_1 K_2 \gamma_k}{4} \min[\frac{\varepsilon_1}{2}, \bar{K}_6] > 0. \tag{3.26}$$

Hence

$$\frac{\ell_k - \ell_{k+1}}{\rho_k} + \|h_k\|^2 - \|h_{k+1}\|^2 \geq 0, \tag{3.27}$$

for all acceptable steps which are generated by IPTR algorithm 2.4. Let $k \in \{k_j\}$ be an element between the two elements $k_{\hat{i}}$ and $k_{\hat{i}+1}$ which are consecutive elements of the sequence $\{k_i\}$. From inequality 3.26, we have

$$\sum_{k=k_{\hat{i}}}^{k_{\hat{i}+1}-1} \frac{\{\ell_k - \ell_{k+1}\}}{\rho_k} + \|h_{k_{\hat{i}}}\|^2 - \|h_{k_{\hat{i}+1}}\|^2 \geq \tau_1 \frac{\varepsilon_1 K_2 \gamma_k}{4} \min[\frac{\varepsilon_1}{2}, \bar{K}_6] > 0.$$

Since the value of ρ_k is the same for all iterates $k_{\hat{i}}, \ldots, k_{\hat{i}+1} - 1$, we have

$$\frac{\ell_{k_{\hat{i}}} - \ell_{k_{\hat{i}+1}}}{\rho_{k_{\hat{i}}}} + \|h_{k_{\hat{i}}}\|^2 - \|h_{k_{\hat{i}+1}}\|^2 \geq \tau_1 \frac{\varepsilon_1 K_2 \gamma_k}{4} \min[\frac{\varepsilon_1}{2}, \bar{K}_6].$$

Since $\rho_k \to \infty$ as $k \to \infty$, and $|\ell_k|$ is bounded, we can write

$$\|h_{k_{\hat{i}}}\|^2 - \|h_{k_{\hat{i}+1}}\|^2 \geq \tau_1 \frac{\varepsilon_1 K_2 \gamma_k}{8} \min[\frac{\varepsilon_1}{2}, \bar{K}_6] > 0,$$

for $k_{\hat{i}}$ sufficiently large. But this leads to a contradiction with Lemma 3.12.

Secondly, if the sequence of the penalty parameters $\{\rho_k\}$ is bounded, then there exists an integer $\bar{k}$ such that for all $k \geq \bar{k}$, we have $\rho_k = \bar{\rho}$. Since all the iterates of $\{k_j\}$ are acceptable, then for any $\bar{k} \in \{k_j\}$ we have

$$\Phi_{\tilde{k}} - \Phi_{\tilde{k}+1} = Ared_{\tilde{k}} \geq \tau_1 Pred_{\tilde{k}}. \tag{3.28}$$

From Lemma 3.10, inequality 3.4, we have for any $\tilde{k} \in \{k_j\}$ and $\tilde{k} \geq \bar{k}$

$$\begin{aligned} Pred_{\tilde{k}} &\geq \frac{K_2 \tau_{\tilde{k}} \bar{\rho}}{2} \|h_{\tilde{k}}\| \min\{\|h_{\tilde{k}}\|, \delta_{\tilde{k}}\} \\ &\geq \frac{K_2 \tau_{\tilde{k}} \bar{\rho} \varepsilon_1}{4} \min\{\| \frac{\varepsilon_1}{2\delta_{max}}, 1\} \delta_{\tilde{k}} \\ &\geq K_8 \delta_{\tilde{k}} \geq K_6 K_8, \end{aligned} \tag{3.29}$$

such that $K_8 = \frac{K_2 \tau_{\tilde{k}} \bar{\rho} \varepsilon_1}{4} \min\{\| \frac{\varepsilon_1}{2\delta_{max}}, 1\}$. From inequalities 3.28 and 3.29, we have

$$\Phi_{\tilde{k}} - \Phi_{\tilde{k}+1} \geq \tau_1 K_6 K_8 > 0.$$

This gives a contradiction with the fact that $\{\Phi_k\}$ is bounded below when $\{\rho_k\}$ is bounded. Hence in both cases, we have a contradiction. Thus, the supposition is not correct and the theorem is proved.

Theorem 3.2. *Under assumptions $GS_1 - GS_5$, the algorithm is terminated because*

$$\lim_{k \to \infty} [\|Z_k^T P_k \nabla \ell_k\| + \|h_k\|] = 0$$

Proof. Assume that IPTR algorithm 2.4 does not terminate and that some subsequences of $\{\|Z_k^T P_k \nabla \ell_k\|\}$ convergence to zero, then the nontermination is immediately contradicted by Theorem 3.1.

Now assume that for $\bar{k}$ sufficiently large, there exists an index $\tilde{k} > \bar{k}$ such that $\|Z_k^T P_k \nabla \ell_k\| \geq \varepsilon_1$. Let $\{k_j\}$ be a subsequence of iterates that satisfy $\|h_{k_j}\| > \eta \delta_{k_j}$, then $\lim_{k_j \to \infty} \delta_{k_j} = 0$ such that $\lim_{k_j \to \infty} \|h_{k_j}\| = 0$. This implies the existence of an infinite sequence $\{k_j\}$ of rejected trial steps. But this leads to contradiction. To show this, we consider two cases:

Firstly, if the sequence of the penalty parameter $\{\rho_k\}$ is unbounded, then from inequalities 3.3 and 3.4, we have

$$\begin{aligned} \frac{|Ared_{k_j} - Pred_{k_j}|}{Pred_{k_j}} &\leq \frac{[\kappa_1 \|d_{k_j}\|^2 + \kappa_2 \rho_{k_j} \|d_{k_j}\|^3 + \kappa_3 \rho_{k_j} \|d_{k_j}\|^2 \|h_{k_j}\|]}{\frac{K_2}{2} \rho_{k_j} \|h_{k_j}\| \min\{\|h_{k_j}\|, \delta_{k_j}\}} \\ &\leq \frac{[\kappa_1 \|d_{k_j}\|^2 + \kappa_2 \rho_{k_j} \|d_{k_j}\|^3 + \kappa_3 \rho_{k_j} \|d_{k_j}\|^2 \|h_{k_j}\|]}{\frac{K_2}{2} \rho_{k_j} \|h_{k_j}\| \|d_{k_j}\| \min\{\eta, 1\}} \\ &\leq \frac{2\kappa_1}{K_2 \rho_{k_j} \eta \min\{\eta, 1\}} + [\frac{2\kappa_2}{K_2} + \frac{2\kappa_3}{K_2 \eta}] \frac{\delta_{k_j}}{\min\{\eta, 1\}}. \end{aligned}$$

As $\rho_{k_j} \to \infty$ and $\delta_{k_j} \to 0$, then $\frac{|Ared_{k_j} - Pred_{k_j}|}{Pred_{k_j}} \to 0$. This means that for k_j large enough, all trial steps $\|d_{k_j}\|$ must be accepted. This leads to a contradiction, so δ_{k_j} must be bounded away from zero in this case.

Secondly, if the sequence of the penalty parameter $\{\rho_k\}$ is bounded, then there exists an integer $\bar{k}$ such that for all $k \geq \bar{k}$, $\rho_k = \bar{\rho}$. Now, we discuss three cases:

1] If the previous step is accepted ($j = 1$), then from the way of updating the trust-region radius in algorithm 2.2, we have $\delta_{k_j} \geq \delta_{min}$. That is δ_{k_j} is bounded away from zero in this case.

2] If $j > 1$ and $\|h_{k_r}\| > \eta\delta_{k_r}$ for all $r = 1, \cdots, j-1$. Then

$$(1-\tau_1) < \frac{|Ared_{k_r} - Pred_{k_r}|}{Pred_{k_r}}$$

such that all the trial steps on $\{k_j\}$ are rejected. From above inequality and inequalities 3.2 and 3.4 we have

$$(1-\tau_1) < \frac{|Ared_{k_r} - Pred_{k_r}|}{Pred_{k_r}} \leq \frac{2K_1\|d_{k_r}\|}{K_2\|h_{k_r}\|\min\{1,\eta\}}.$$

Hence

$$\|d_{k_r}\| > \frac{K_2(1-\tau_1)\min\{1,\eta\}}{2K_1}\|h_{k_k}\|.$$

But from the way of updating the radius of trust-region in algorithm 2.2, all the rejected trial steps satisfy $\delta_{k_r} = \beta_1\|d_{k_r}\|$, hence

$$\begin{aligned} \delta_{k_r} = \beta_1\|d_{k_{r-1}}\| \;&\geq\; \frac{K_2\beta_1\eta(1-\tau_1)\min\{1,\eta\}}{2K_1}\delta_{k_r} \\ &\geq\; \frac{K_2\beta_1\eta(1-\tau_1)\min\{1,\eta\}}{2K_1}\delta_{min}. \end{aligned}$$

This means that δ_{k_r} is bounded away from zero in this case.

3] If $j > 1$ and $\|h_{k_r}\| > \eta\delta_{k_r}$ does not hold for all r. Hence, there exists an integer i such that $\|h_{k_r}\| \leq \eta\delta_{k_r}$ for all $r = 1, \cdots, i$, and $\|h_{k_r}\| > \eta\delta_{k_r}$ for all $r = i+1, \cdots, j-1$. Since $\|h_{k_r}\| > \eta\delta_{k_r}$ for all $r = i+1, \cdots, j-1$, then as the above case we can prove δ_{k_r} is bounded away from zero.

The case when $\|h_{k_r}\| \leq \eta\delta_{k_r}$ for all $r = 1, \cdots, i$, then for all rejected trial steps, we have

$$(1-\tau_1) < \frac{|Ared_{k_r} - Pred_{k_r}|}{Pred_{k_r}}.$$

From inequality 3.2, Lemma 3.6, and the above inequality, we have

$$(1-\tau_1) < \frac{|Ared_{k_r} - Pred_{k_r}|}{Pred_{k_r}} \leq \frac{K_1\bar{\rho}\|d_{k_r}\|}{K_5}.$$

Hence

$$\|d_{k_r}\| > \frac{K_5(1-\tau_1)}{K_1\bar{\rho}}.$$

From the way of updating the radius of trust-region, we have for all rejected trial step

$$\delta_{k_r} = \beta_1\|d_{k_{r-1}}\| > \frac{\beta_1 K_5(1-\tau_1)}{K_1\bar{\rho}}.$$

Hence, δ_{k_r} is bounded away from zeros. This leads to a contradiction and then for k_j sufficiently large, all the iterates satisfy $\|h_k\| \leq \eta\delta_{k_j}$.

For all successful steps and from the way of updating the radius of trust-region and Lemma 3.6, we have for all $k \in \{k_j\}$ and $k \geq \bar{k}$

$$\Phi_k - \Phi_{k+1} = Ared_k \geq \tau_1 Pred_k \geq \tau_1 K_5 \gamma_k \delta_k, \quad for \;\; all \;\; k \geq \bar{k}.$$

We proved in the above cases, that δ_{k_j} is bounded away from zeros. Then $\Phi_k - \Phi_{k+1} > 0$. This leads to a contradiction with the fact that $\{\Phi_k\}$ is bounded below when $\{\rho_k\}$ is bounded. Hence in both cases, we have a contradiction. Thus, the supposition is not correct and the theorem is proved.

4. Application

In this section, firstly the proposed algorithm IPTR is applied to the engineering application which is called **two-echelon supply chain system** with one manufacturer and one retailer.

The manufacturer purchases raw materials from the supplier first, then after the manufacturer's production and processing, the end products are sold to the retailer, this problem is formulated as bilevel models for joint pricing and lot-sizing decisions, see [34].

$$\begin{array}{ll}
\max_{t_1,t_2} & f_u = (t_2 - \tilde{P}_s - \tilde{T}_c - \tilde{M}_c) t_1 t_3 y_1 - 0.5 \tilde{c}_m \tilde{T} \tilde{P}_s t_3 (y_1 - 1) - \tilde{O}_m t_1 \\
s.t. & \tilde{P}_s + \tilde{T}_c + \tilde{M}_c \leq t_2 \leq 10, \\
 & t_1 \geq 0, \\
\quad \max_{y_1,y_2} & f_l = t_1 t_2 t_3 y_1 (y_2 - 1) - 0.5 \tilde{c}_r \tilde{T} t_2 t_3 - \tilde{O}_r t_1 y_1 \\
\quad s.t. & 1 \leq y_2 \leq 5, \\
 & y_1 \geq 0.
\end{array}$$

where $\tilde{T} = 52$; $\tilde{P}_s = 4$; $\tilde{T}_c = 0.5$; $\tilde{M}_c = 1$; $\tilde{c}_m = \tilde{c}_r = 0.001$; $\tilde{O}_m = 400$; $\tilde{O}_r = 200$. For more details about the above application and its notations, see [34].

We solve this model in case of the manufacturer is the leader, who makes the first decision, and the retailer is the follower. Our results, when applying Algorithm (2.4) is $t_1 = 5.8778$, $t_2 = 6.002$, $t_3 = 19710.195$, $y_1 = 7.691$, $y_2 = 2.6007$, $f_u = 431230$, and $f_l = 8548300$, which is closed to whose reported in [34].

Secondly, we introduce an extensive variety of possible numeric bilevel nonlinear programming problems to clarify the effectiveness of our IPTR algorithm, since, Problems 1,2,6,7,13, and 14 have quadratic functions in both levels. Problems 3,4,5,8,9 all the inner level functions are convex and Problem 10 [27], at fixed x, the inner problem is convex. These problems are solved numerically with the help of algorithm (2.4) to clarify the effectiveness of that approach. For each test example, 10 independent runs with different initial starting point are performed to observe the consistency of the outcome. Statistical results of all examples are summarized in Table 1 which shows that the results found by the IPTR algorithm (2.4) are approximate or equal to those by the compared algorithms in the literature.

Table 1 also including the mean number of iterations (iter),the mean number of function evaluations (nfunc), the mean value of CPU time (CPUs) in seconds.

For comparison, we have included the corresponding results of the mean value of CPU time (CPUs) obtained by Method in [31](Table 2), [27](Table 3), and [44](Table 4) respectively. It is clear from the results that our approach is capable for treating nonlinear bilevel programming problems even the

Problem 1 [31]:

$$\begin{array}{rl} \min_t & f_u = y_1^2 + y_2^2 + t^2 - 4t \\ s.t. & 0 \le t \le 2, \\ \min_y & f_l = y_1^2 + 0.5y_2^2 + y_1y_2 + \\ & (1-3t)y_1 + (1+t)y_2, \\ s.t. & 2y_1 + y_2 - 2t \le 1, \\ & y_1 \ge 0, \quad y_2 \ge 0. \end{array}$$

Problem 2 [31]:

$$\begin{array}{rl} \min_t & f_u = y_1^2 + y_3^2 - y_1y_3 - 4y_2 - 7t_1 + 4t_2 \\ s.t. & t_1 + t_2 \le 1, \\ & t_1 \ge 0, \quad t_2 \ge 0 \\ \min_y & f_l = y_1^2 + 0.5y_2^2 + 0.5y_3^2 + y_1y_2 + \\ & (1-3t_1)y_1 + (1+t_2)y_2, \\ s.t. & 2y_1 + y_2 - y_3 + t_1 - 2t_2 + 2 \le 0, \\ & y_1 \ge 0; \quad y_2 \ge 0 \quad y_3 \ge 0. \end{array}$$

Problem 3 [31]:

$$\begin{array}{rl} \min_t & f_u = 0.1(t_1^2 + t_2^2) - 3y_1 - 4y_2 + 0.5(y_1^2 + y_2^2) \\ s.t. & \\ \min_y & f_l = 0.5(y_1^2 + 5y_2^2) - 2y_1y_2 - t_1y_1 - t_2y_2, \\ s.t. & -0.333y_1 + y_2 - 2 \le 0, \\ & y_1 - 0.333y_2 - 2 \le 0, \\ , & y_1 \ge 0, \quad y_2 \ge 0, \end{array}$$

Problem 4 [31]:

$$\begin{array}{rl} \min_t & f_u = t_1^2 - 2t_1 + t_2^2 - 2t_2 + y_1^2 + y_2^2 \\ s.t. & t_1 \ge 0, \quad t_2 \ge 0 \\ \min_y & f_l = (y_1 - t_1)^2 + (y_2 - t_2)^2, \\ s.t. & 0.5 \le y_1 \le 1.5, \\ & 0.5 \le y_2 \le 1.5, \end{array}$$

upper and the lower levels are convex or not and the computed results converge to the optimal solution which is similarly or approximate to the optimal that reported in literature. Finally, it is clear from the comparison between the solutions obtained using IPTR algorithm with literature, that IPTR is able to find the optimal solution of all problems by a small number of iterations, small number of function evaluations, and less time.

We offered the numerical results of our algorithm using MATLAB (R2013a)(8.2.0.701)64-bit(win64) and a starting point $x_0 \in int(\hat{\boldsymbol{G}})$. The following parameter setting is used: $\delta_{min} = 10^{-3}$, $\delta_0 = max(\|s_0^{cp}\|, \delta_{min})$, $\delta_{max} = 10^3\delta_0$, $\tau_1 = 10^{-4}$, $\tau_2 = 0.75$, $\beta_1 = 0.5$, $\beta_2 = 2$, $\hat{\varepsilon} = 0.01$, $\varepsilon_1 = 10^{-8}$, and $\varepsilon_2 = 10^{-10}$.

Problem 5 [31]:

$$\begin{array}{rl} \min_t & f_u = t^2 + (y - 10)^2 \\ s.t. & -t + y \le 0, \\ & 0 \le t \le 15, \\ \min_y & f_l = (t + 2y - 30)^2, \\ s.t. & t + y \le 20, \\ & 0 \le y \le 20, \end{array}$$

Problem 6 [31]:

$$\begin{array}{rl} \min_t & f_u = (t-1)^2 + 2y_1^2 - 2t \\ s.t. & t \ge 0, \\ \min_y & f_l = (2y_1 - 4)^2 + (2y_2 - 1)^2 + ty_1, \\ s.t. & 4t + 5y_1 + 4y_2 \le 12, \\ & -4t - 5y_1 + 4y_2 \le -4, \\ & 4t - 4y_1 + 5y_2 \le 4, \\ & -4t + 4y_1 + 5y_2 \le 4, \\ & y_1 \ge 0, \quad y_2 \ge 0, \end{array}$$

Problem 7 [31]:

$$\begin{aligned}\min_t \quad & f_u = (t-5)^2 + (2y+1)^2\\ s.t. \quad & t \geq 0,\\ \min_y \quad & f_l = (2y-1)^2 - 1.5ty,\\ s.t. \quad & -3t + y \leq -3,\\ & t - 0.5y \leq 4,\\ & t + y \leq 7,\\ & y \geq 0.\end{aligned}$$

Problem 8 [31]:

$$\begin{aligned}\min_t \quad & f_u = t_1^2 - 3t_1 + t_2^2 - 3t_2 + y_1^2 + y_2^2\\ s.t. \quad & t_1 \geq 0, \quad t_2 \geq 0,\\ \min_y \quad & f_l = (y_1 - t_1)^2 + (y_2 - t_2)^2,\\ s.t. \quad & 0.5 \leq y_1 \leq 1.5,\\ & 0.5 \leq y_2 \leq 1.5,\end{aligned}$$

Problem 9 [27]:

$$\begin{aligned}\min_t \quad & f_u = 16t^2 + 9y^2\\ s.t. \quad & -4t + y \leq 0,\\ & t \geq 0,\\ \min_y \quad & f_l = (t + y - 20)^4,\\ s.t. \quad & 4t + y - 50 \leq 0,\\ & y \geq 0.\end{aligned}$$

Problem 10 [27]:

$$\begin{aligned}\min_t \quad & f_u = t^3 y_1 + y_2\\ s.t. \quad & 0 \leq t \leq 1,\\ \min_y \quad & f_l = -y_2\\ s.t. \quad & t y_1 \leq 10,\\ & y_1^2 + t y_2 \leq 1,\\ & y_2 \geq 0.\end{aligned}$$

Problem 11 [44]:

$$\begin{aligned}\min_t \quad & f_u = 2t_1 + 2t_2 - 3y_1 - 3y_2 - 60\\ s.t. \quad & t_1 + t_2 + y_1 - 2y_2 \leq 40,\\ & 0 \leq t_1 \leq 50,\\ & 0 \leq t_2 \leq 50,\\ \min_y \quad & f_l = (y_1 - t_1 + 20)^2 + (y_2 - t_2 + 20)^2,\\ s.t. \quad & t_1 - 2y_1 \geq 10,\\ & t_2 - 2y_2 \geq 10,\\ & -10 \leq y_1 \leq 20,\\ & -10 \leq y_2 \leq 20.\end{aligned}$$

Problem 12 [27]:

$$\begin{aligned}\min_t \quad & f_u = (t-3)^2 + (y-2)^2\\ s.t. \quad & -2t + y - 1 \leq 0,\\ & t - 2y + 2 \leq 0,\\ & t + 2y - 14 \leq 0,\\ & 0 \leq t \leq 8,\\ \min_y \quad & f_l = (y-5)^2\\ s.t. \quad & y \geq 0.\end{aligned}$$

Problem 13 [44]:

$$\begin{aligned}\min_t \quad & f_u = -t_1^2 - 3t_2^2 - 4y_1 + y_2^2\\ s.t. \quad & t_1^2 + 2t_2 \leq 4,\\ & t_1 \geq 0, \quad t_2 \geq 0,\\ \min_y \quad & f_l = 2t_1^2 + y_1^2 - 5y_2,\\ s.t. \quad & t_1^2 - 2t_1 + 2t_2^2 - 2y_1 + y_2 \geq -3,\\ & t_2 + 3y_1 - 4y_2 \geq 4,\\ & y_1 \geq 0, \quad y_2 \geq 0.\end{aligned}$$

Problem 14 [44]:

$$\begin{aligned}\min_t \quad & f_u = (t-1)^2 + (y-1)^2\\ s.t. \quad & t \geq 0,\\ \min_y \quad & f_l = 0.5y^2 + 500y - 50ty\\ s.t. \quad & y \geq 0.\end{aligned}$$

Problem 15 [44]:

$$\begin{array}{lll} \min_t & f_u = -8t_1 - 4t_2 + 4y_1 - 40y_2 - 4y_3 \\ s.t. & t_1 \geq 0, \quad t_2 \geq 0 \\ & \min_y \ f_l = t_1 + 2t_2 + y_1 + y_2 + 2y_3, \\ & s.t. \ y_2 + y_3 - y_1 \leq 1, \\ & 2t_1 - y_1 + 2y_2 - 0.5y_3 \leq 1, \\ & 2t_2 + 2y_1 - y_2 - 0.5y_3 \leq 1, \\ & y_i \geq 0, \quad i = 1, 2, 3. \end{array}$$

Problem 16 [44]:

$$\begin{array}{lll} \min_t & f_u = -8t_1 - 4t_2 + 4y_1 - 40y_2 - 4y_3 \\ s.t. & t_1 \geq 0, \quad t_2 \geq 0 \\ & \min_y \ f_l = \frac{1+t_1+t_2+2y_1-y_2+y_3}{6+2t_1+y_1+y_2-3y_3}, \\ & s.t. \ -y_1 + y_2 + y_3 + y_4 = 1, \\ & 2t_1 - y_1 + 2y_2 - 0.5y_3 + y_5 = 1, \\ & 2t_2 + 2y_1 - y_2 - 0.5y_3 + y_6 = 1, \\ & y_i \geq 0, \quad i = 1, ..., 6. \end{array}$$

Table 1. Comparisons of the results by IPTR algorithm 2.4 and methods in reference.

Problem name	(t_*, y_*) IPTR	f_u^* f_l^* IPTR	iter nfunc IPTR	CPUs time IPTR	(t_*, y_*) Ref.	f_u^* f_l^* Ref.
prob(1)	(0.8503, 0.0227, 0.03589)	-2.6764 0.0332	11 12	1.43	(0.8438, 0.7657, 0)	-2.0769 -0.5863
prob(2)	(0.609, 0.391, 0, 0,1.828)	0.6086 1.6713	10 14	1.987	(0.609, 0.391, 0, 0, 1.828)	0.6426 1.6708
prob(3)	(0.97, 3.14, 2.6, 1.8)	-8.92 -6.05	6 8	2.9	(0.97, 3.14, 2.6, 1.8)	-8.92 -6.05
prob(4)	(.5,.5,.5,.5)	-1 0	10 14	1.68	(0.5, 0.5, 0.5, 0.5)	-1 0
prob(5)	(9.839,10.059)	96.809 0.0019	6 9	1.635	(10.03, 9.969)	100.58 0.001
prob(6)	(1.6879, 0.8805,0)	-1.3519 7.4991	6 11	4.1	NA	3.57 2.4
prob(7)	(1, 0)	17 1	12 13	1.9	(1, 0)	17 1
prob(8)	(0.75,0.75, 0.75, 0.75)	-2.25 0	10 11	1.002	($\sqrt{3}/2$, $\sqrt{3}/2$, $\sqrt{3}/2$, $\sqrt{3}/2$)	-2.1962 0
prob(9)	(11.138,5)	2209.8 222.52	10 13	1.95	(11.25,5)	2250 197.753
prob(10)	(1,0,6.6387e-06)	6.6387e-06 -6.6387e-06	5 7	2.987	(1,0,1)	1 -1
prob(11)	(24.972, 29.653, 5.0238,9.7565)	4.9101 0.01332	9 12	3.742	(25,30,5,10)	5 0
prob(12)	(3,5)	9 0	8 9	1.23	(3,5)	9 0
prob(13)	(0,1.7405, 1.8497,0.9692)	-15.548 -1.4247	5 7	2.1	(0,2,1.875,0.9063)	-12.68 -1.016
prob(14)	(10.016,0.81967)	81.328 -0.3359	6 8	2.12	(10.04,0.1429)	82.44 0.271
prob(15)	(0,0.9,0,0.6,0.4)	-29.2 3.2	5 6	20.512	(0,0.9,0,0.6,0.4)	-29.2 3.2
prob(16)	(0,0.9,0,0.6,0.4,0,0,0)	-29.2 0.3148	5 7	40.319	(0,0.9,0,0.6,0.4,0,0,0)	-29.2 0.3148

Table 2. Comparisons of the results by IPTR (2.4) and method [31].

Problem name	(t_*, y_*) IPTR	f_u^* f_l^* IPTR	CPUs IPTR	(t_*, y_*) method [31]	f_u^* f_l^* method [31]	CPUs method [31].
prob(1)	(0.8503, 0.0227, 0.03589)	-2.6764 0.0332	1.43	(0.8462,0.769 2,0)	-2.0769 -0.5917	1.734
prob(2)	(0.609, 0.391,0, 0,1.828)	0.6086 1.6713	1.987	(0.6111, 0.3889,0, 0, 1.8333)	0.6389 1.6806	2.375
prob(3)	(0.97, 3.14, 2.6, 1.8)	-8.92 -6.05	2.9	(1.031 6, 3.097 8, 2.597 0, 1.792 9)	-8.917 2 -6.137 0	3.315
prob(4)	(0.5,0.5,0.5,0.5)	-1 0	1.68	(0.5,0.5,0.5,0.5)	-1 0	1.576
prob(5)	(9.839,10.059)	96.809 0.0019	1.635	(10, 10)	100 0	1.825
prob(6)	(1.6879, 0.8805,0)	-1.3519 7.4991	4.1	(1.8889, 0.8889,0)	-1.2099 7.6173	4.689
prob(7)	(1, 0)	17 1	1.9	(1,0)	17 1	1.769
prob(8)	(0.75,0.75, 0.75, 0.75)	-2.25 0	1.002	(0.75,0.75, 0.75, 0.75)	-2.25 0	1.124

Table 3. Comparisons of the results by IPTR (2.4) and method [27].

Problem name	(t_*, y_*) IPTR	f_u^* f_l^* IPTR	CPUs IPTR	(t_*, y_*) method [27]	f_u^* f_l^* method [27]	CPUs method [27].
prob(9)	(11.138,5)	2209.8 222.52	1.95	(11.25, 5)	2250 197.753	2.21
prob(10)	(1,0,6.6387e-06)	6.6387e-06 -6.6387e-06	1.9	(1,0,-1)	-1 1	3.38
prob(12)	(3,5)	9 0	1.23	(3,5)	9 0	-

Table 4. Comparisons of the results by IPTR (2.4) and method [44].

Problem name	(t_*, y_*) IPTR	f_u^* f_l^* IPTR	CPUs IPTR	(t_*, y_*) method [44]	f_u^* f_l^* method [44]	CPUs method [44].
prob(3)	(0.97, 3.14, 2.6, 1.8)	-8.92 -6.05	2.9	(1.03, 3.097, 2.59,1.79	-8.92 -6.14	11.854
prob(5)	(9.839,10.059)	96.809 0.0019	1.635	(10,10)	100.014 4.93e-7	5.888
prob(6)	(1.6879, 0.8805,0)	-1.3519 7.4991	4.1	(1.8888,0.888)	-1.2091 7.6145	25.332
prob(11)	(24.972, 29.653 5.0238,9.7565)	4.9101 0.01332	3.742	(0,30,-10,10)	0 100	37.308
prob(13)	(0,1.7405, 1.8497,0.9692)	-15.548 -1.4247	2.1	(4.4e-7,2, 1.875,0.9063)	-12.65 -1.021	14.42
prob(14)	(10.016,0.81967)	81.328 -0.3359	2.12	(10.0164,0.8197)	18.3279 -0.3359	4.218
prob(15)	(0,0.9,0,0.6,0.4)	-29.2 3.2	20.512	(0,0.9,0,0.6,0.4)	-29.2 3.2	45.39
prob(16)	(0,0.9,0,0.6,0.4,0,0,0)	-29.2 0.3148	40.319	(0,0.9,0,0.6,0.4,0,0,0)	-29.2 0.3148	107.55

5. Concluding remarks

This paper presented a new technique for solving a nonlinear bilevel optimization problem based on using the slack variable with KKT condition to transform NBLP problem into an equivalent smooth SONP problem. A Newton's interior-point method with Das scaling matrix is utilized to solve the equivalent smooth SONP problem effectively. Newton's method is locally method, so a trust region technique is utilized to ensure global convergence from any starting point. On applying this methodology we overcome some known difficulties on treating such problems, as

- A trust-region technique can induce strongly global convergence, which is very important technique for solving a smooth optimization problems and is more robust when they deal with rounding errors
- Our approach used to transform Problem 1.3 which is not smooth to smooth problem
- Using the interior-point method guarantees the converges quadratically to a stationary point.

On the other hand, the global convergence theorems for the IPTR algorithm is presented and numerical results reflect the good behavior of our algorithm and computed results converge to the optimal solutions. Finally, it is clear from the comparison between the solutions obtained using IPTR algorithm with literature, that IPTR is able to find the optimal solution of all problems by a small number of iterations.

Acknowledgments

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

Conflict of interest

The authors declare that there is no conflict of interest in this paper.

References

1. D. Aksen, S. Akca, N. Aras, A bilevel partial interdiction problem with capacitated facilities and demand outsourcing, *Comput. Oper. Res.*, **41** (2014), 346–358. https://doi.org/10.1016/j.cor.2012.08.013
2. Y. Abo-Elnaga, M. El-Shorbagy, Multi-Sine Cosine Algorithm for Solving Nonlinear Bilevel Programming Problems, *Int. J. Comput. Int. Sys.*, **13** (2020), 421–432. https://doi.org/10.2991/ijcis.d.200411.001
3. Y. Abo-Elnaga, S. Nasr, Modified Evolutionary Algorithm and Chaotic Search for Bilevel Programming Problems, *Symmetry*, **12** (2020), 1–29. https://doi.org/10.3390/sym12050767
4. Y. Abo-Elnag, S. Nasr, K-means cluster interactive algorithm-basedevolutionary approach for solving bilevel multi-objective programming problems, Alexandria Engineering Journal, **61** (2022), 811–827. https://doi.org/10.1016/j.aej.2021.04.098
5. M. Bazaraa, H. Sherali, C. Shetty, *Nonlinear programming theory and algorithms*, John Wiley and Sons, 2006. https://doi.org/10.1002/0471787779
6. R. Byrd, Omojokun, Robust trust-region methods for nonlinearly constrained optimization, A talk presented at the Second SIAM Conference on Optimization, Houston, TX, 1987.
7. A. Burgard, P. Pharkya, C. Maranas, Optknock: a bilevel programming framework for identifying gene knockout strategies formicrobial strain optimization, *Biotechnol. Bioeng.*, **84** (2003), 647–657. https://doi.org/10.1002/bit.10803
8. O. Ben-Ayed, O. Blair, Computational difficulty of bilevel linear programming, *Oper. Res.*, **38** (1990), 556–560. https://doi.org/10.1287/opre.38.3.556
9. R. Byrd, M. Hribar, J. Nocedal, An interior point algorithm for largescale nonlinear programming, *SIAM J. Optim.*, **9** (1999), 877–900. https://doi.org/10.1137/S1052623497325107
10. R. Byrd, J. Gilbert, J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, *Math. Program.*, **89** (2000), 149–185. https://doi.org/10.1007/PL00011391
11. F. E. Curtis, O. Schenk, A. Wachter, An interior-point algorithm for large-scale nonlinear optimization with inexact step computations, *SIAM J. Sci. Comput.*, **32** (2010), 3447–3475. https://doi.org/10.1137/090747634
12. I. Das, An interior point algorithm for the general nonlinear programming problem with trust region globlization, Technical Report 96-61, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center Hampton, VA, USA, 1996.
13. J. Dennis, M. Heinkenschloss, L. Vicente, Trust-region interior-point SQP algorithms for a class of nonlinear programming problems, *SIAM J. Control Optim.*, **36** (1998), 1750–1794. https://doi.org/10.1137/S036012995279031

14. S. Dempe, Foundations of Bilevel Programming, Kluwer Academic Publishers, London, 2002.

15. H. Esmaeili, M. Kimiaei, An efficient implementation of a trust-region method for box constrained optimization, *J. Appl. Math. Comput.*, **48** (2015), 495–517. https://doi.org/10.1007/s12190-014-0815-0

16. B. El-Sobky, A global convergence theory for an active trust region algorithm for solving the general nonlinear programming problem, *Appl. Math. Comput.*, **144** (2003), 127–157. https://doi.org/10.1016/S0096-3003(02)00397-1

17. B. El-Sobky, A Multiplier active-set trust-region algorithm for solving constrained optimization problem, *Appl. Math. Comput.*, **219** (2012), 928–946. https://doi.org/10.1016/j.amc.2012.06.072

18. B. El-Sobky, An interior-point penalty active-set trust-region algorithm, *Journal of the Egyptian Mathematical Society*, **24** (2016), 672–680. https://doi.org/10.1016/j.joems.2016.04.003

19. B. El-Sobky, An active-set interior-point trust-region algorithm, *Pac. J. Optim.*, **14** (2018), 125–159.

20. B. El-Sobky, Y. Abouel-Naga, Multi-objective optimal load flow problem with interior-point trust-region strategy, *Electr. Pow. Syst. Res.*, **148** (2017), 127–135. https://doi.org/10.1016/j.epsr.2017.03.014

21. 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

22. B. El-Sobky, A. Abotahoun, An active-set algorithm and a trust-region approach in constrained minimax problem, *Comput. Appl. Math.*, **37** (2018), 2605–2631. https://doi.org/10.1007/s40314-017-0468-3

23. B. El-Sobky, A. Abotahoun, A Trust-Region Algorithm for Solving Mini-Max Problem, *J. Comput. Math.*, **36** (2018), 881–902. https://doi.org/10.4208/jcm.1705-m2016-0735

24. T. Edmunds, J. Bard, Algorithms for nonlinear bilevel mathematical programs, *IEEE transactions on Systems, Man, and Cybernetics*, **21** (1991), 83–89. https://doi.org/10.1109/21.101139

25. J. Falk, J. Liu, On bilevel programming, Part I: general nonlinear cases, *Math. Program.*, **70** (1995), 47–72. https://doi.org/10.1007/BF01585928

26. M. Hestenes, Muliplier and gradient methods, *J. Optimiz. Theory App.*, **4** (1969), 303–320. https://doi.org/10.1007/BF00927673

27. Z. H. Gumus, C. A. Flouda, Global Optimization of Nonlinear Bilevel Programming Problems, *J. Global Optim.*, **20** (2001), 1–31.

28. V. Gonzlez, J. Vallejo, G. Serrano, A scatter search algorithm for solving a bilevel optimization model for determining highway tolls, *Comput. Syst.*, **19** (2015), 3529–3549. https://doi.org/10.13053/cys-19-1-1916

29. G. Hibino, M. Kainuma, Y. Matsuoka, Two-level mathematical programming for analyzing subsidy options to reduce greenhouse-gas emissions, Tech. Report WP-96-129, IIASA, Laxenburg, Austria, 1996.

30. D. Kouri, M. Heinkenschloss, D. Ridzal, B. van Bloemen Waanders, A Trust-Region Algorithm with Adaptive Stochastic Collocation for PDE Optimization under Uncertainty, *SIAM J. Sci. Comput.*, **35** (2020), 1847–1879. https://doi.org/10.1137/120892362

31. H. Li, Y. Jiao, L. Zhang, Orthogonal genetic algorithm for solving quadratic bilevel programming problems, *J. Syst. Eng. Electron.*, **21** (2010), 763–770. https://doi.org/10.3969/j.issn.1004-4132.2010.05.008

32. N. Li, D. Xue, W. Sun, J. Wang, A stochastic trust-region method for unconstrained optimization problems, *Math. Probl. Eng.*, (2019). https://doi.org/10.1155/2019/8095054

33. Y. Lva, T. Hua, G. Wanga, Z. Wanb, A neural network approach for solving nonlinear bilevel programming problem, *Comput. Math. Appl.*, **55** (2008), 2823–2829. https://doi.org/10.1016/j.camwa.2007.09.010

34. W. Ma, M. Wang, X. Zhu, Improved particle swarm optimization based approach for bilevel programming problem-an application on supply chain model, *Int. J. Mach. Learn. Cyber*, **5** (2014), 281–290. https://doi.org/10.1007/s13042-013-0167-3

35. L. Ma, G. Wang, A Solving Algorithm for Nonlinear Bilevel Programing Problems Based on Human Evolutionary Model, *Algorithms*, **13** (2020), 1–12. https://doi.org/10.3390/a13100260

36. L. F. Niu, Y. Yuan, A new trust region algorithm for nonlinear constrained optimization, *J. Comput. Math.*, **28** (2010), 72–86. https://doi.org/10.4208/jcm.2009.09-m2924

37. E. Omojokun, Trust-region strategies for optimization with nonlinear equality and inequality constraints, PhD thesis, Department of Computer Science, University of Colorado, Boulder, Colorado, 1989.

38. T. Steihaug, The conjugate gradient method and trust-region in large scale optimization, *Siam J. Numer. Anal.*, **20** (1983), 626–637. https://doi.org/10.1137/0720042

39. S. Sadatrasou, M. Gholamian, K. Shahanaghi, An application of data mining classification and bi-level programming for optimal credit allocation, *Decis. Sci. Lett.*, **4** (2015), 35–50. https://doi.org/10.5267/j.dsl.2014.9.005

40. G. Savard, J. Gauvin, The steepest descent direction for the nonlinear bilevel programming problem, *Oper. Res. Lett.*, **15** (1994), 265–272. https://doi.org/10.1016/0167-6377(94)90086-8

41. N. Thoai, Y. Yamamoto, A. Yoshise, Global optimization method for solving mathematical programs with linear complementarity constraints, Discussion Paper No. 987, Institute of Policy and Planning Sciences, University of Tsukuba, Japan, 2002.

42. X. Wang, Y. Yuan, A trust region method based on a new affine scaling technique for simple bounded optimization, *Optimization Methods and Software*, **28** (2013), 871–888. https://doi.org/10.1080/10556788.2011.622378

43. X. Wang, Y. Yuan, An augmented Lagrangian trust region method for equality constrained optimization, *Optimization Methods and Software*, **30** (2015), 559–582. https://doi.org/10.1080/10556788.2014.940947

44. Y. Wang, Y. Jiao, H. Li, An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-Handling scheme, *IEEE T. Syst. Man Cy. C*, **35** (2005), 221–232. https://doi.org/10.1109/TSMCC.2004.841908

45. Y. Yuan, Recent advances in trust region algorithms, *Math. Program. Ser. B*, **151** (2015), 249–281. https://doi.org/10.1007/s10107-015-0893-2

46. M. Zeng, Q. Ni, A new trust region method for nonlinear equations involving fractional mode, *Pac. J. Optim.*, **15** (2019), 317–329.