Journal of Industrial and Systems Engineering
Vol. 14, No. 4, pp. 279-292
Autumn (November) 2022

An algorithm to solve linear fractional bi-level problems based on Taylor approximation

Massoumeh Emami[1], Elnaz Osgooei[1*]

[1]*Faculty of Science, Urmia University of Technology, Urmia, Iran*

m.emami.3606@gmail.com, e.osgooei@uut.ac.ir

Abstract

The linear fractional bi-level problems are strongly NP-hard and non-convex, which results in high computational complexity to find the optimal solution. In this paper, we propose an efficient algorithm for solving a class of non-linear bi-level optimization problems, where the upper and lower objectives are linear fractional. The main idea behind the proposed algorithm is to obtain a single objective optimization problem via Taylor approximation. The proposed algorithm is composed of four steps. In the first, the lower level of the problem is converted into the convex optimization problem by using auxiliary variables and approximation techniques. Next, a single objective optimization problem is obtained by adopting the dual Lagrange method and Karush-Kuhn-Tucker (KKT) conditions. The obtained problem is non-convex with high computational complexity challenging to solve. Hence, the Fischer-Burmeister function is applied to smooth the problem. Finally, the first-order Taylor approximation is adopted to transform the non-linear problem into the linear one. Numerical results confirm the effectiveness of the proposed algorithm in comparison with Estimation of Distribution Algorithm (EDA) in terms of convergence performance.

Keywords: Bi-level programming, linear fractional bi-level problem, Taylor approximation, dual Lagrange method, Fischer-Burmeister

1-Introduction

A bi-level programming problem (BLP) involves two types of decision makers which the constraints region of one is determined implicitly by the solution of the second. The application of BLP includes transportation, management, planning and optimal design. For more information see the literature (Dempe, 2002), (Dempe, 2003), (Colson et al, 2007), (Vicente and Calamai, 1994), (Aghapour and Osgooei, 2022) and (Zhang et al, 2016).

1-1-Related works

In this sub-section, the related works are provided in two categories: 1) Linear/non-linear bi-level problems and 2) Linear fractional bi-level problems.

1) *Linear/non-linear bi-level problems*: In recent decades, bi-level optimization problems have been widely studied at the point of theoretical and computational complexity. In Lv et al., (2007) the linear bi-level problem was transformed into a single objective optimization problem using Karush-Kuhn-Tucker (KKT) optimality conditions. In White and Anandalingam (1993), the authors proposed a duality gap-penalty function to obtain the global solution of linear bi-level programming. An approximation method for solving the non-linear bi-level optimization problems was addressed in Colson (2005).

---

*Corresponding author

In Colson (2005) the upper level and constraints were approximated as linear functions, as well as a quadratic approximation was provided for the lower level. In Marcotte (2001) a trust region approach was proposed to solve the non-linear bi-level problems, where a linear program was implemented at the upper level, and a linear variational inequality function was proposed at the lower level. In Angelo (2013, June), the authors proposed an algorithm to solve the bi-level optimization problems, which employed differential evolution as an efficient candidate. A heuristic algorithm based on the scatter search was considered in Camacho-Vallejo (2015) to solve the bi-level problem of planning the production and distribution in a supply chain. In Li and Wang, (2007, August) a hybrid Genetic Algorithm (GA) was adopted for solving the non-linear bi-level optimization problems in which the simplex method was considered for designing a new crossover operator. In Hejazi et al, (2002) an efficient method based on the GA was proposed to solve the linear bi-level problems. In Wan et al, (2013) a hybrid algorithm involving Particle Swarm Optimization (PSO) and Chaos Searching Technique (CST) was presented for solving the non-linear bi-level optimization problems. In Wang et al, (2011) an effective meta-heuristic algorithm was proposed for obtaining the local solution of the non-linear bi-level problems, where a new fitness function was considered for the upper level. In Watada et al, (2020) an efficient method based on an improved artificial bee colony, Hopfield network, and Boltzmann machine was proposed to solve the non-linear bi-level programming problem. A Stackelberg game framework was proposed in Zhou et al, (2022) to solve the non-linear bi-level problems and coordinate the decision-making process.

2) *Linear fractional bi-level problems*: the linear fractional bi-level problems are strongly non-convex and NP-hard, and thus finding the optimal solution is complicated and computationally complex. In recent years, some algorithms are developed to obtain the optimal solution. In Li (2015) a GA algorithm with global convergence was proposed for solving the linear fractional bi-level problems. In Chen (2019) the authors considered a modified enumerative searching method to solve a class of bi-level programming problems where the lower level is linear fractional. Then, the Charnes-Cooper transformation was used for dealing with non-linearity of the lower level. A new two-level vertex-searching algorithm for solving the continuous linear fractional bi-level problems was proposed in Chen (2020) to find the optimal solution. In Nayak and Ojha (2020) a new method based on interval coefficients of decision variables was developed for determining the optimal solution of the linear fractional bi-level problems. In Toksarı (2010) the authors proposed a solution for linear fractional bi-level problems by adopting Taylor series. In this method, the levels of the problem were classified as upper and lower, and then they were weighted based on their classes. Finally, Taylor series was used to convert the problem into single objective. In Chen et al, (2018, August) a novel approach based on Estimation of Distribution Algorithm (EDA) and heuristic algorithms was proposed to solve the linear fractional bi-level problems. A weighting method was developed in Mishra (2005) where a non-dominated solution set was obtained. In Roghanian et al, (2008) a new method by integrating goal programming, KKT optimality conditions, and penalty functions was presented. In Alessa (2021) an interactive approach was introduced to solve the linear fractional bi-level problems so that the minimal adequate level can be updated at upper level.

1-2-Contributions and organization

In the above-mentioned papers, the main goal is to solve the linear fractional bi-level problems and obtain the optimum solution. However, developing an efficient algorithm with low computational complexity remains scarce in the studied works. To address this issue, in this paper, a new algorithm for solving the linear fractional bi-level problems via the first-order Taylor approximation technique is proposed with low computational complexity. To this end, a four-step alternative method is presented. First, the lower level of the problem is transformed into the convex by adopting the auxiliary variables and other approximation techniques. Next, the lower level, which is converted into the convex, is added to the upper-level problem. With this, the bi-level problem is converted into a single objective optimization problem. Then, the Fischer-Burmeister function is applied to smooth the problem and cope with computational complexity. Finally, the problem is converted into a linear one by adopting the first-order Taylor approximation technique. The CVX optimizer in MATLAB is used to obtain the optimal solution of the linear problem. The performance of the proposed method is evaluated by two numerical examples and compared with EDA method presented in Chen et al, (2018, August). The results illustrate that the proposed method can obtain the optimal solution and provide fast convergence with low computational complexity in comparison to the EDA approach.

The remainder of this paper is organized as follows. In section 2, some Theorems and definitions are presented. Section 3 presents the proposed algorithm for solving linear fractional bi-level problems. In section 4, the numerical evaluations are provided to confirm the effectiveness of the proposed algorithm in comparison with estimation of distribution algorithm in terms of convergence performance. Finally, section 5 concludes the paper.

2-Preliminaries

In this section, some necessary theorems and definitions are provided.

Definition 2.1 Denote $m \geq 0$ and $n \geq 0$ as the two functions (or values) (Hussein and Kamalabadi, 2014). The Fischer-Burmeister function can be defined as

$\phi(m, n, \varepsilon) = m + n - \sqrt{m^2 + n^2 + \varepsilon}$

Where $\varepsilon \ll 1$ and

$m \geq 0, n \geq 0, \phi(m, n, \varepsilon) = 0 \leftrightarrow mn = 0, m \geq 0, n \geq 0$

Theorem 2.1 Assume $s(x) = x_1 x_2$. It is easy to prove that the function $S(x, \varphi) = \frac{\varphi}{2} x_1^2 + \frac{1}{2\varphi} x_2^2$ is convex and provides an upper-bound of $s(x)$ for any value of $\varphi > 0$, namely $S(x, \varphi) \geq s(x), \forall\ \varphi > 0$. Also, by substituting $\varphi = \frac{x_2}{x_1}$ into $S(x, \varphi)$, the following equation is satisfied

$S(x, \varphi) = s(x)$

Proof. Please refer to Tran et al, (2012) and Beck et al, (2010).

3-Proposed method for solving linear fractional bi-level problems

In this section, the mathematical model of the proposed method for solving the linear fractional bi-level problems is provided. Let us consider a linear fractional bi-level problem where the upper level controls the decision variable x and the lower level controls the decision variable y. The optimization problem is formulated as follows.

$$\max_{x} f_1(x, y) = \frac{\alpha_1 + c_{11}x + c_{12}y}{\beta_1 + d_{11}x + d_{12}y}$$
where y solves:
$$\max_{y} f_2(x, y) = \frac{\alpha_2 + c_{21}x + c_{22}y}{\beta_2 + d_{21}x + d_{22}y} \quad (1)$$
s.t.:
$$\text{C1}: g_i(x, y) \leq 0, i = 1, \dots, m$$
$$\text{C2}: x, y \geq 0$$

Where x and y are optimization variables; $f_1(x, y)$ and $f_2(x, y)$ denote the upper and lower levels of the problem (1), respectively, and finally C1 represents the linear constraints of the problem.

The aim of this paper is to propose an efficient solution to solve the linear fractional bi-level problems based on the first-order Taylor approximation. Hence, the following steps are presented to describe the proposed method.

Step 1: Convert the lower level of the problem (1) into convex one

In this step, the lower level of problem (1) is converted into the convex one. To this end, an auxiliary variable η_1 is introduced. Hence, the problem in (1) is transformed into

$$\max_{x} f_1(x, y) = \frac{\alpha_1 + c_{11}x + c_{12}y}{\beta_1 + d_{11}x + d_{12}y} \quad (2)$$

Where y and η_1 solve:
$\max_{y} \eta_1$
s.t.:
C1,
C2: $\frac{\alpha_2 + c_{21}x + c_{22}y}{\beta_2 + d_{21}x + d_{22}y} \geq \eta_1$
C3: $x, y, \eta_1 \geq 0$

With auxiliary variable, the objective function of the lower level is linear with respect to η_1. However, the left side of constraint C2 is non-concave with respect to y. To cope with the non-linearity and non-convexity, the constraint C2 is replaced with the following two constraints.

C4: $\alpha_2 + c_{21}x + c_{22}y \geq \eta_1 \xi_1$
C5: $\xi_1 \geq \beta_2 + d_{21}x + d_{22}y$

Where $\xi_1 > 0$ denotes a new additional auxiliary variable. By inspecting C4, it is observed that it is not a convex constraint due to the term $q(\eta_1, \xi_1)$=$\eta_1 \xi_1$ and this the right side of C4 is approximated with the following function via theorem 2.1.

$$\tilde{q}(\eta_1, \xi_1) = \frac{1}{2\varphi^{(z-1)}}(\eta_1)^2 + \frac{\varphi^{(z-1)}}{2}(\xi_1)^2 \quad (3)$$

Where $\varphi^{(z-1)} = \frac{\eta_1}{\xi_1}$ and z denotes the iterative index. As stated in theorem 2.1, (3) is convex with respect to η_1 and ξ_1. Therefore, the problem in (2) is transformed into
$\max_{x} f_1(x, y) = \frac{\alpha_1 + c_{11}x + c_{12}y}{\beta_1 + d_{11}x + d_{12}y}$
where y, η_1, and ξ_1 solve:

$$\max_{y, \eta_1, \xi_1} \eta_1 \quad (4)$$
s.to:
C1, C3, C5
C6: $\alpha_2 + c_{21}x + c_{22}y \geq \tilde{q}(\eta_1, \xi_1)$
C7: $\xi_1 > 0, \eta_1 \geq 0$

The lower level of the problem (4) is linear concerning y and convex with respect to η_1 and ξ_1. Thus, it is a convex optimization problem.

Step 2: Convert the problem into a single objective problem
Since the lower level of the problem (4) is convex, it can be added to the upper level of the problem by using KKT conditions and the dual Lagrange method. To this end, the Lagrangian function of the problem (4) can be written as follows.

$$\mathcal{L}(x, y, \eta_1, \xi_1) = \eta_1 + \lambda\big(\alpha_2 + c_{21}x + c_{22}y - \tilde{q}(\eta_1, \xi_1)\big) + \mu(\xi_1 - \beta_2 - d_{21}x - d_{22}y) + \sum_{i=1}^{m} \rho_i\big(-g_i(x, y)\big) \quad (5)$$

Where $\rho_i, \forall i = 1, \dots, m$, μ, and λ are the Lagrange multipliers associated with C1, C5, and C6, respectively. Now, the following problem is created by adopting the Lagrangian function and KKT conditions.

$$
\begin{aligned}
&\max_{x,y,\eta_1,\xi_1,\lambda,\mu,\rho} f_1(x,y) = \frac{\alpha_1 + c_{11}x + c_{12}y}{\beta_1 + d_{11}x + d_{12}y} \\
&\text{s.t.:} \\
&\text{C1, C3, C5, C6, C7} \\
&\text{C8: } \nabla_y \mathcal{L}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho) = 0 \\
&\text{C9: } \nabla_{\eta_1} \mathcal{L}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho) = 0 \\
&\text{C10: } \nabla_{\xi_1} \mathcal{L}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho) = 0 \\
&\text{C11: } \lambda\left(\alpha_2 + c_{21}x + c_{22}y - \tilde{q}(\eta_1,\xi_1)\right) = 0 \\
&\text{C12: } \mu(\xi_1 - \beta_2 - d_{21}x - d_{22}y) = 0 \\
&\text{C13: } \rho_i g_i(x,y) = 0, \forall i = 1, \dots, m \\
&\text{C14: } \lambda, \mu, \rho_i \geq 0, \forall i = 1, \dots, m
\end{aligned}
\tag{6}
$$

Where $\nabla_y \mathcal{L}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho) = \lambda c_{22} - \mu d_{22} - \sum_{i=1}^{m} \rho_i \left(\nabla_y g_i(x,y)\right)$, $\nabla_{\eta_1} \mathcal{L}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho) = 1 - \frac{\lambda\eta_1}{\varphi^{(z-1)}}$, and $\nabla_{\xi_1} \mathcal{L}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho) = -\lambda\xi_1\varphi^{(z-1)} + \mu$. It is observed that the constraints C9 and C10 are non-linear since $\lambda\eta_1$ and $\lambda\xi_1$ are non-linear functions. We define $k(\lambda,\eta_1) = \lambda\eta_1$ and $h(\lambda,\xi_1) = \lambda\xi_1$. Now, based on theorem 2.1.

$$
\tilde{k}(\lambda,\eta_1) = \frac{1}{2{\varphi_1}^{(z-1)}}(\lambda)^2 + \frac{{\varphi_1}^{(z-1)}}{2}(\eta_1)^2 \tag{7}
$$

$$
\tilde{h}(\lambda,\xi_1) = \frac{1}{2{\varphi_2}^{(z-1)}}(\lambda)^2 + \frac{{\varphi_2}^{(z-1)}}{2}(\xi_1)^2 \tag{8}
$$

Where ${\varphi_1}^{(z-1)} = \frac{\lambda}{\eta_1}$, ${\varphi_2}^{(z-1)} = \frac{\lambda}{\xi_1}$, and z represents the iterative index.
Then, the optimization problem is converted into

$$
\begin{aligned}
&\max_{x,y,\eta_1,\xi_1,\lambda,\mu,\rho} f_1(x,y) = \frac{\alpha_1 + c_{11}x + c_{12}y}{\beta_1 + d_{11}x + d_{12}y} \\
&\text{s.t.:} \\
&\text{C1, C3, C5, C6, C7, C11} - \text{C14} \\
&\text{C15: } c_{22} - \mu d_{22} - \sum_{i=1}^{m} \rho_i \left(\nabla_y g_i(x,y)\right) = 0 \\
&\text{C16: } \tilde{k}(\lambda,\eta_1) = \varphi^{(z-1)} \\
&\text{C17: } \tilde{h}(\lambda,\xi_1)\varphi^{(z-1)} = \mu
\end{aligned}
\tag{9}
$$

Step 3: Smoothing the problem using Fischer-Burmeister Function.

In this step, the Fischer-Burmeister function presented in definition 2.1 is applied to smooth the problem and deal with computational complexity. In the problem (9), constraints C1, C5, C6, and C11-C13 have high computational complexity, and thus the aforementioned constraints can be replaced with the following constraints.

$$
\text{C18: } \lambda + \alpha_2 + c_{21}x + c_{22}y - \tilde{q}(\eta_1,\xi_1) - \sqrt{\lambda^2 + \left(\alpha_2 + c_{21}x + c_{22}y - \tilde{q}(\eta_1,\xi_1)\right)^2 + \varepsilon} = 0
$$

$$
\text{C19: } \mu + \xi_1 - \beta_2 - d_{21}x - d_{22}y - \sqrt{\mu^2 + (\xi_1 - \beta_2 - d_{21}x - d_{22}y)^2 + \varepsilon} = 0
$$

C20: $\rho_i - g_i(x,y) - \sqrt{{\rho_i}^2 + \left(g_i(x,y)\right)^2 + \varepsilon} = 0, \forall i = 1, \dots, m$

Hence, the optimization problem is transformed into

$$\max_{x,y,\eta_1,\xi_1,\lambda,\mu,\rho} f_1(x,y) = \frac{\alpha_1 + c_{11}x + c_{12}y}{\beta_1 + d_{11}x + d_{12}y}$$
$$\text{s.t.:}$$
$$\text{C1, C7, C15} - \text{C20} \tag{10}$$

The problem in (10) has still non-linear form due to the fractional objective function and non-convex constraints. Hence, similar to step 1, an auxiliary variable η_2 is introduced, where $\frac{\alpha_1+c_{11}x+c_{12}y}{\beta_1+d_{11}x+d_{12}y} \geq \eta_2$. As seen, the aforementioned constraint can be replaced with the following two constraint.

C21: $\alpha_1 + c_{11}x + c_{12}y \geq \eta_2\xi_2$
C22: $\xi_2 \geq \beta_1 + d_{11}x + d_{12}y$

Where $\xi_2 > 0$ is a new additional auxiliary variable. Also, based on theorem 2.1, the right side of constraint C21 is approximated as

$$\tilde{v}(\eta_2, \xi_2) = \frac{1}{2{\varphi_3}^{(z-1)}}(\eta_2)^2 + \frac{{\varphi_3}^{(z-1)}}{2}(\xi_2)^2 \tag{11}$$

Where ${\varphi_3}^{(z-1)} = \frac{\eta_2}{\xi_2}$.
Thus, the optimization problem is

$$\max_{x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2} \eta_2$$
$$\text{s.t.:}$$
$$\text{C15} - \text{C20, C22}$$
$$\text{C23: } \alpha_1 + c_{11}x + c_{12}y \geq \tilde{v}(\eta_2, \xi_2) \tag{12}$$

Step 4: Convert the optimization problem into linear problem via first-order Taylor approximation

To convert the problem into linear, the objective and its constraints must be linear. To this end, the first-order Taylor approximation is applied for non-linear constraints. Let $t = (x, y, \eta_1, \xi_1, \lambda, \mu, \rho, \eta_2, \xi_2)$ be the set of optimization variables. Also, for simplicity, the constraints C16-C20 and C23 are respectively re-written as below.

$$R(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2) = \tilde{k}(\lambda,\eta_1) - 1 \tag{13}$$
$$U(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2) = \tilde{h}(\lambda,\xi_1)\varphi^{(z-1)} - \mu \tag{14}$$
$$H(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2) = \lambda + \alpha_2 + c_{21}x + c_{22}y - \tilde{q}(\eta_1,\xi_1) - \sqrt{\lambda^2 + \left(\alpha_2 + c_{21}x + c_{22}y - \tilde{q}(\eta_1,\xi_1)\right)^2 + \varepsilon} \tag{15}$$
$$F(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2) = \mu + \xi_1 - \beta_2 - d_{21}x - d_{22}y - \sqrt{\mu^2 + (\xi_1 - \beta_2 - d_{21}x - d_{22}y)^2 + \varepsilon} \tag{16}$$
$$A_i(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2) = \rho_i - g_i(x,y) - \sqrt{{\rho_i}^2 + \left(g_i(x,y)\right)^2 + \varepsilon}, \forall i = 1, \dots, m \tag{17}$$
$$B(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2) = \alpha_1 + c_{11}x + c_{12}y - \tilde{v}(\eta_2,\xi_2) \tag{18}$$

Now, based on the first-order Taylor approximation, the equations provided in (13)-(18) can be respectively approximated by (19)-(24).

$$\tilde{R}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=R(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)+\nabla_t R(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)(t-t^{z-1}) \quad (19)$$

$$\tilde{U}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=U(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)+\nabla_t U(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)(t-t^{z-1}) \quad (20)$$

$$\tilde{H}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=H(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)+\nabla_t H(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)(t-t^{z-1}) \quad (21)$$

$$\tilde{F}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=F(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)+\nabla_t F(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)(t-t^{z-1}) \quad (22)$$

$$\tilde{A}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=A(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)+\nabla_t A(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)(t-t^{z-1}) \quad (23)$$

$$\tilde{B}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=B(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)+\nabla_t B(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)(t-t^{z-1}) \quad (24)$$

Finally, the optimization problem is converted into:

$$\begin{aligned}
&\max_{x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2} \quad \eta_2\\
&\text{s.t.:}\\
&\text{C1, C7C15, C22,}\\
&\text{C24: } \tilde{R}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=0\\
&\text{C25: } \tilde{U}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=0\\
&\text{C26: } \tilde{H}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=0\\
&\text{C27: } \tilde{F}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=0\\
&\text{C28: } \tilde{A}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)=0\\
&\text{C29: } \tilde{B}(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)\geq 0
\end{aligned} \quad (25)$$

The optimization problem in (25) is linear that can be optimally solved via CVX optimizer in MATLAB software. In table 1, the procedure of the proposed algorithm is presented, where Z_{max} denotes the maximum number of iterations. In this algorithm, the optimization variables are obtained, iteratively.

Table 1. The procedure of the proposed algorithm

Initialize: $z=1$ and $t^{z-1}=(x,y,\eta_1,\xi_1,\lambda,\mu,\rho,\eta_2,\xi_2)$

Repeat

- Solve the problem (25) by CVX to obtain the optimization variables, namely t
- Update $t^{z-1}=t$
- Update $\varphi^{(z-1)}=\frac{\eta_1}{\xi_1}$, $\varphi_1{}^{(z-1)}=\frac{\lambda}{\eta_1}$, $\varphi_2{}^{(z-1)}=\frac{\lambda}{\xi_1}$, and $\varphi_3{}^{(z-1)}=\frac{\eta_2}{\xi_2}$
- $z=z+1$

Until the convergence occurs or $z=Z_{max}$

4- Numerical results

In this section, the performance of the proposed algorithm presented in table 1 is investigated via numerical optimization problems. To this end, two linear fractional bi-level problems are solved by the proposed method. To verify the results, the proposed method is compared with EDA method presented in Chen et al, (2018, August) and the method in Chen (2020). The final linear problem is solved by CVX optimizer in MATLAB, which is a powerful tool for solving convex and linear optimization problems.

Example 1: consider the following problem.

$$\max_{x_1} f_1(x_1, x_2) = \frac{x_1 + 2x_2}{x_1 + x_2 + 1}$$

where x_2 solves:

$$\max_{x_2} f_2(x_1, x_2) = \frac{2x_1 + x_2}{2x_1 + 3x_2 + 1}$$

s.t.:

$$-x_1 + 2x_2 \leq 3$$
$$2x_1 - x_2 \leq 3$$
$$x_1 + x_2 \geq 3$$
$$x_1 \geq 0, x_2 \geq 0$$

The following steps are followed to obtain the optimal solution.

Step 1:

In step 1, the following problem is obtained by applying the mentioned stages in section 3.

$$\max_{x_1} f_1(x_1, x_2) = \frac{x_1 + 2x_2}{x_1 + x_2 + 1}$$

Where x_2, η_1, and ξ_1 solve:

$$\max_{x_2,\eta_1,\xi_1} \eta_1$$

s.t.:

$$2x_1 + x_2 \geq \tilde{q}(\eta_1, \xi_1) = \frac{1}{2\varphi^{(z-1)}}(\eta_1)^2 + \frac{\varphi^{(z-1)}}{2}(\xi_1)^2$$
$$\xi_1 \geq 2x_1 + 3x_2 + 1$$
$$-x_1 + 2x_2 \leq 3$$
$$2x_1 - x_2 \leq 3$$
$$x_1 + x_2 \geq 3$$
$$x_1 \geq 0, x_2 \geq 0$$

Where η_1 and ξ_1 are new auxiliary variables and $\varphi^{(z-1)} = \frac{\eta_1}{\xi_1}$.

Step 2:

In this step, the following optimization problem can be obtained by using the KKT condition as.

$$\max_{x_1,x_2,\eta_1,\xi_1,\lambda,\mu,\rho_1,\rho_2,\rho_3} f_1(x_1, x_2) = \frac{x_1 + 2x_2}{x_1 + x_2 + 1}$$

s.t.:

$$\lambda - 3\mu - 2\rho_1 + \rho_2 + \rho_3 = 0$$
$$\tilde{k}(\lambda, \eta_1) = \varphi^{(z-1)}$$
$$\tilde{h}(\lambda, \xi_1) = \mu$$
$$\lambda\big(2x_1 + x_2 - \tilde{q}(\eta_1, \xi_1)\big) = 0$$
$$\mu(\xi_1 - 2x_1 - 3x_2 - 1) = 0$$
$$\rho_1(x_1 - 2x_2 + 3) = 0$$
$$\rho_2(-2x_1 + x_2 + 3) = 0$$
$$\rho_3(x_1 + x_2 - 3) = 0$$

$$2x_1 + x_2 \geq \tilde{q}(\eta_1, \xi_1) = \frac{1}{2\varphi^{(z-1)}}(\eta_1)^2 + \frac{\varphi^{(z-1)}}{2}(\xi_1)^2$$
$$\xi_1 \geq 2x_1 + 3x_2 + 1$$
$$-x_1 + 2x_2 \leq 3$$
$$2x_1 - x_2 \leq 3$$
$$x_1 + x_2 \geq 3$$
$$x_1 \geq 0, x_2 \geq 0$$

Where λ, μ, ρ_1, ρ_2, and ρ_3 are Lagrange multipliers associated with constraints of the optimization problem;

$\tilde{k}(\lambda, \eta_1) = \frac{1}{2\varphi_1{}^{(z-1)}}(\lambda)^2 + \frac{\varphi_1{}^{(z-1)}}{2}(\eta_1)^2$, $\tilde{h}(\lambda, \xi_1) = \left(\frac{1}{2\varphi_2{}^{(z-1)}}(\lambda)^2 + \frac{\varphi_2{}^{(z-1)}}{2}(\xi_1)^2\right)\varphi^{(z-1)}$, $\varphi_1{}^{(z-1)} = \frac{\lambda}{\eta_1}$, and $\varphi_2{}^{(z-1)} = \frac{\lambda}{\xi_1}$.

Step 3:
In this step, the following optimization problem is obtained by using Fischer-Burmeister function.

$$\max_{x_1,x_2,\eta_1,\xi_1,\lambda,\mu,\rho_1,\rho_2,\rho_3,\eta_2} \eta_2$$
s.t.:
$$x_1 + 2x_2 \geq \tilde{v}(\eta_2, \xi_2) = \frac{1}{2\varphi_3{}^{(z-1)}}(\eta_2)^2 + \frac{\varphi_3{}^{(z-1)}}{2}(\xi_2)^2$$
$$\xi_2 \geq x_1 + x_2 + 1$$
$$\lambda - 3\mu - 2\rho_1 + \rho_2 + \rho_3 = 0$$
$$\tilde{k}(\lambda, \eta_1) = \varphi^{(z-1)}$$
$$\tilde{h}(\lambda, \xi_1) = \mu$$
$$\lambda + 2x_1 + x_2 - \tilde{q}(\eta_1, \xi_1) - \sqrt{\lambda^2 + \left(2x_1 + x_2 - \tilde{q}(\eta_1, \xi_1)\right)^2 + \varepsilon} = 0$$
$$\mu + \xi_1 - 2x_1 - 3x_2 - 1 - \sqrt{\mu^2 + (\xi_1 - 2x_1 - 3x_2 - 1)^2 + \varepsilon} = 0$$
$$\rho_1 + x_1 - 2x_2 + 3 - \sqrt{\rho_1^2 + (x_1 - 2x_2 + 3)^2 + \varepsilon} = 0$$
$$\rho_2 - 2x_1 + x_2 + 3 - \sqrt{\rho_2^2 + (-2x_1 + x_2 + 3)^2 + \varepsilon} = 0$$
$$\rho_3 + x_1 + x_2 - 3 - \sqrt{\rho_3^2 + (x_1 + x_2 - 3)^2 + \varepsilon} = 0$$
$$x_1 \geq 0, x_2 \geq 0$$

Step 4:
In this step, the non-linear optimization problem is converted into the linear one via the first-order Taylor approximation. Hence, we have:

$$A_1(x_1, x_2, \eta_1, \xi_1, \lambda, \mu, \rho_1, \rho_2, \rho_3, \eta_2, \xi_2) = x_1 + 2x_2 - \tilde{v}(\eta_2, \xi_2)$$
$$A_2(x_1, x_2, \eta_1, \xi_1, \lambda, \mu, \rho_1, \rho_2, \rho_3, \eta_2, \xi_2) = \tilde{k}(\lambda, \eta_1) - \varphi^{(z-1)}$$
$$A_3(x_1, x_2, \eta_1, \xi_1, \lambda, \mu, \rho_1, \rho_2, \rho_3, \eta_2, \xi_2) = \tilde{h}(\lambda, \xi_1) - \mu$$
$$A_4(x_1, x_2, \eta_1, \xi_1, \lambda, \mu, \rho_1, \rho_2, \rho_3, \eta_2, \xi_2) = \lambda + 2x_1 + x_2 - \tilde{q}(\eta_1, \xi_1) - \sqrt{\lambda^2 + \left(2x_1 + x_2 - \tilde{q}(\eta_1, \xi_1)\right)^2 + \varepsilon}$$
$$A_5(x_1, x_2, \eta_1, \xi_1, \lambda, \mu, \rho_1, \rho_2, \rho_3, \eta_2, \xi_2) = \mu + \xi_1 - 2x_1 - 3x_2 - 1 - \sqrt{\mu^2 + (\xi_1 - 2x_1 - 3x_2 - 1)^2 + \varepsilon}$$

$$A_6(x_1, x_2, \eta_1, \xi_1, \lambda, \mu, \rho_1, \rho_2, \rho_3, \eta_2, \xi_2) = \rho_1 + x_1 - 2x_2 + 3 - \sqrt{\rho_1^2 + (x_1 - 2x_2 + 3)^2 + \varepsilon}$$

$$A_7(x_1, x_2, \eta_1, \xi_1, \lambda, \mu, \rho_1, \rho_2, \rho_3, \eta_2, \xi_2) = \rho_2 - 2x_1 + x_2 + 3 - \sqrt{\rho_2^2 + (-2x_1 + x_2 + 3)^2 + \varepsilon}$$

$$A_8(x_1, x_2, \eta_1, \xi_1, \lambda, \mu, \rho_1, \rho_2, \rho_3, \eta_2, \xi_2) = \rho_3 + x_1 + x_2 - 3 - \sqrt{\rho_3^2 + (x_1 + x_2 - 3)^2 + \varepsilon}$$

Where $t = (x_1, x_2, \eta_1, \xi_1, \lambda, \mu, \rho_1, \rho_2, \rho_3, \eta_2, \xi_2)$ is the set of the optimization variables.

Now, we can obtain a linear optimization problem by applying the first-order Taylor approximation. This procedure is performed in MATLAB. Then, the linear problem is solved by CVX optimizer. The results are obtained by Core i5 CPU 4200M and 8.00 GB RAM. The results of the proposed method, EDA Chen et al, (2018, August) and the method in Chen (2020) are provided in table 2. These results indicate that the proposed method obtains the optimal solution and provides low computational complexity compared to the EDA and the method in Chen (2020).

Also, in figure 1, we investigate the convergence behavior of the proposed method and EDA method versus the number of iterations. As can be seen, the proposed method needs six iterations to coverage, which provides fast convergence in comparison of EDA method.

Table 2. The obtained results of example 1

method	Obtained solution	$f_1(x_1, x_2)$	$f_2(x_1, x_2)$	Run time
Proposed method	$x_1 = 3$ $x_2 = 3$	1.28571	0.20673	5.2 sec
EDA method (Chen et al, 2018 August)	$x_1 = 3$ $x_2 = 3$	1.28571	0.20673	30.3 sec
The method in (Chen, 2020)	$x_1 = 3$ $x_2 = 3$	1.28571	0.20673	9.4 sec

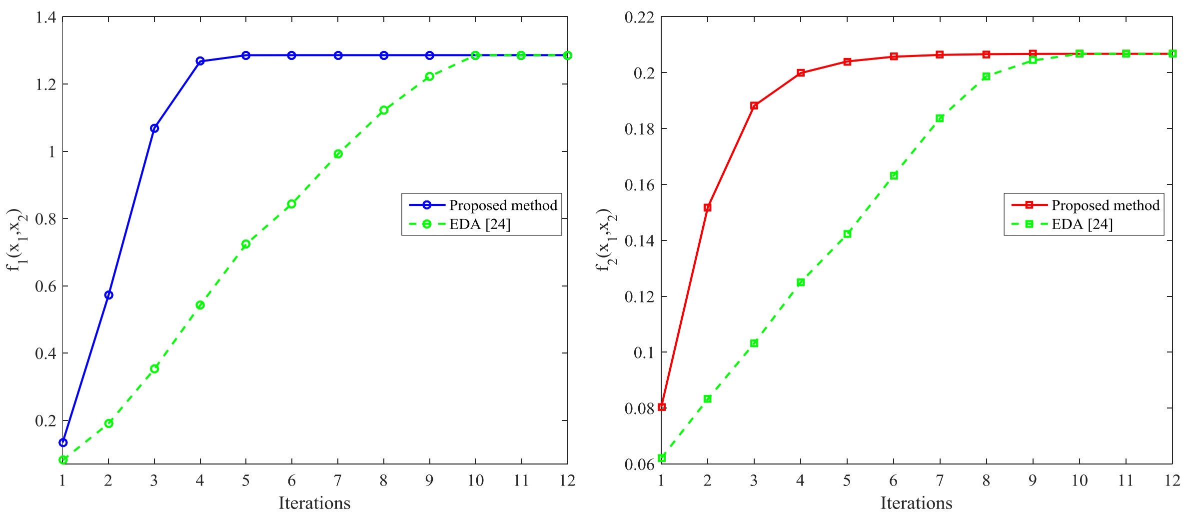

Fig. 1. The convergence behavior of the proposed method and EDA method for example 1

Example 2: Consider the following problem

$$\max_{0\leq x\leq 8} f_1(x,y) = \frac{x}{y_1 + y_2 + 1}$$

where y_1 and y_2 solve:

$$\max_{y\geq 0} f_2(x,y) = \frac{y_1}{y_2 + 1}$$

s.t.:

$$y_1 + y_2 + x \leq 10$$

$$y_1 \leq 9$$

$$y_2 \leq 7$$

Similar to example 1, the mentioned steps presented in section 3 are followed and the optimal solution is obtained by CVX optimizer.

The optimal solutions of the proposed method, EDA, and the method in Chen (2020), are provided in table 3. The results show that the proposed method obtains the optimal solution and provides low computational complexity in comparison to the benchmark methods.

Table 3. The obtained results of example 2

method	Obtained solution	$f_1(x_1, x_2)$	$f_2(x_1, x_2)$	Run time
Proposed method	$x = 8$ $y_1 = 2, y_2 = 0$	2.66667	2	4.8 sec
EDA method (Chen et al, 2018 August)	$x = 8$ $y_1 = 2, y_2 = 0$	2.66667	2	28.3 sec
The method in (Chen, 2020)	$x = 8$ $y_1 = 2, y_2 = 0$	2.66667	2	8.6 sec

In figure 2, the convergence behavior of the proposed method and EDA method is plotted over the number of iterations. We observe that the proposed method converges after five iterations, which is more effective than EDA method in terms of computational complexity.

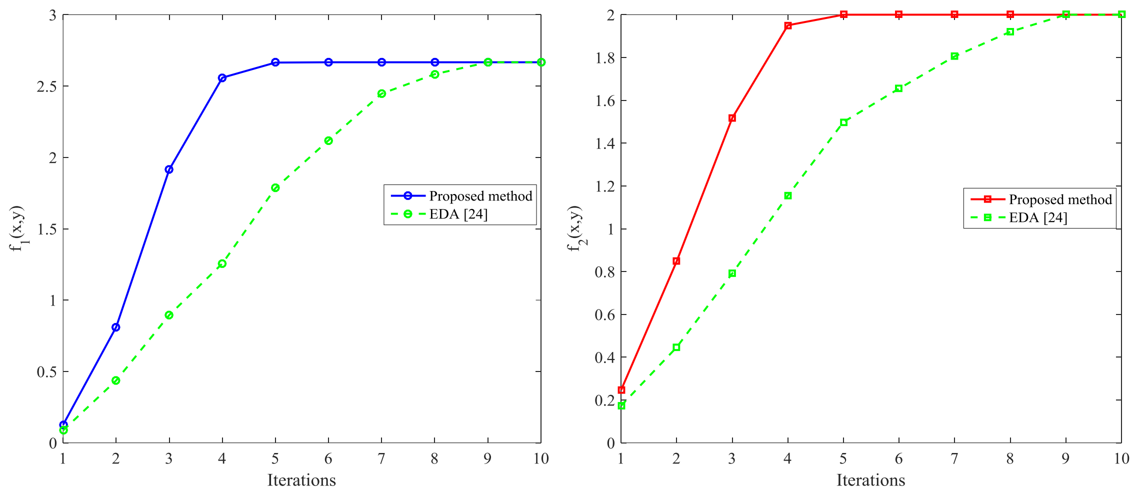

Fig. 2. The convergence behavior of the proposed method for example 2

Example 3: consider the following problem:

$$\max_{y_1,y_2} f_1(y) = \frac{-y_1 + y_2 - 2y_4 - 1}{8 - y_1 - 2y_3 + y_4 + 5y_5}$$

Where $\{y_3, \dots, y_8\}$ solves:

$$\max_{y_3,\dots,y_8} f_1(y) = \frac{-y_1 + y_2 - 2y_4 - 1}{8 - y_1 - 2y_3 + y_4 + 5y_5}$$

s.t.:

$-y_3 + y_4 + y_5 + y_6 = 1,$

$2y_1 - y_3 + 2y_4 - 0.5y_5 + y_7 = 1,$

$2y_2 + 2y_3 - y_4 - 0.5y_5 + y_8 = 1,$

$y_i \geq 0, i = 1, \dots, 8$

Similar to examples 1 and 2, the mentioned steps presented in section 3 are followed and the optimal solution is obtained by CVX optimizer.

The optimal solutions of the proposed method, EDA, and the method in Chen (2020) are provided in table 4. The results show that the proposed method obtains the optimal solution and provides low computational complexity in comparison to the benchmark methods.

Table 4. The obtained results of example 3

Method	Obtained solution $(y_1, \dots, y_8)$	$f_1(y)$	$f_2(y)$	Run time
Proposed method	(0.75,0.75,0,0,1,0,0,0)	-0.0816	-0.7778	5.4 sec
EDA method (Chen et al, 2018 August)	(0.75,0.75,0,0,1,0,0,0)	-0.0816	-0.7778	35.3 sec
The method in (Chen, 2020)	(0.75,0.75,0,0,1,0,0,0)	-0.0816	-0.7778	9.5 sec

5-Conclusion

This paper proposed a new method for solving the linear fractional bi-level problems based on the first-order Taylor approximation. The proposed method was formed in four steps. First, the lower level of the original problem was converted into convex by using the auxiliary variables and approximation techniques. Second, the convex lower level was added to the upper level by the dual Lagrange method and KKT conditions, and thus a single optimization problem was obtained. Third, the problem was smoothed by applying Fischer-Burmeister. Finally, the first-order Taylor approximation was applied to convert the problem into a linear optimization. The CVX optimizer was used to obtain the optimal solution. Numerical examples verified the effectiveness of the proposed method in comparison with EDA method in terms of convergence performance. For future work, we will develop an algorithm to solve the non-linear fractional bi-level problems via the Taylor approach.

References

Aghapour, H., & Osgooei, E. (2022). A novel approach for solving the fully fuzzy bi-level linear programming problems. *Journal of Industrial and Systems Engineering*, *14*(1), 221-237.

Alessa, N. A. (2021). Bi-Level linear programming of intuitionistic fuzzy. *Soft Computing*, *25*(13), 8635-8641.

Angelo, J. S., Krempser, E., & Barbosa, H. J. (2013, June). Differential evolution for bilevel programming. In *2013 IEEE Congress on Evolutionary Computation* (pp. 470-477). IEEE.

Beck, A., Ben-Tal, A., & Tetruashvili, L. (2010). A sequential parametric convex approximation method with applications to nonconvex truss topology design problems. *Journal of Global Optimization*, *47*, 29-51.

Camacho-Vallejo, J. F., Muñoz-Sánchez, R., & González-Velarde, J. L. (2015). A heuristic algorithm for a supply chainʼ s production-distribution planning. *Computers & Operations Research*, *61*, 110-121.

Chen, H. J. (2020). A two-level vertex-searching global algorithm framework for bilevel linear fractional programming problems. *Systems Science & Control Engineering*, *8*(1), 488-499.

Chen, H. J. (2019). A new vertex enumeration-based approach for bilevel linear-linear fractional programming problems. *Journal of Information and Optimization Sciences*, *40*(7), 1413-1427.

Chen, H., Li, H., & Huang, J. (2018, August). An EDA for Solving Linear Fractional Bilevel Programming Problems. In *2018 International Conference on Information Technology and Management Engineering (ICITME 2018)* (pp. 196-200). Atlantis Press.

Colson, B., Marcotte, P., & Savard, G. (2007). An overview of bilevel optimization. *Annals of operations research*, *153*, 235-256.

Colson, B., Marcotte, P., & Savard, G. (2005). A trust-region method for nonlinear bilevel programming: algorithm and computational experience. *Computational Optimization and Applications*, *30*, 211-227.

Dempe, S. (2003). Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints.

Dempe, S. (2002). *Foundations of bilevel programming*. Springer Science & Business Media.

Hejazi, S. R., Memariani, A., Jahanshahloo, G., & Sepehri, M. M. (2002). Linear bilevel programming solution by genetic algorithm. *Computers & Operations Research*, *29*(13), 1913-1925.

Hussein, E., & Kamalabadi, I. (2014). Taylor approach for solving nonlinear bilevel programming problem. *Adv. Comput. Sci., Int. J*, *3*(5), 91-97.

Li, H. (2015). A genetic algorithm using a finite search space for solving nonlinear/linear fractional bilevel programming problems. *Annals of Operations Research*, *235*(1), 543-558.

Li, H., & Wang, Y. (2007, August). A hybrid genetic algorithm for solving nonlinear bilevel programming problems based on the simplex method. In *Third International Conference on Natural Computation (ICNC 2007)* (Vol. 4, pp. 91-95). IEEE.

Lv, Y., Hu, T., Wang, G., & Wan, Z. (2007). A penalty function method based on Kuhn–Tucker condition for solving linear bilevel programming. *Applied mathematics and computation*, *188*(1), 808-813.

Marcotte, P., Savard, G., & Zhu, D. L. (2001). A trust region algorithm for nonlinear bilevel programming. *Operations research letters*, *29*(4), 171-179.

Mishra, S. (2007). Weighting method for bi-level linear fractional programming problems. *European Journal of Operational Research*, *183*(1), 296-302.

Nayak, S., & Ojha, A. K. (2020). Solving Bi-Level Linear Fractional Programming Problem with Interval Coefficients. In *Numerical Optimization in Engineering and Sciences: Select Proceedings of NOIEAS 2019* (pp. 265-273). Springer Singapore.

Roghanian, E., Aryanezhad, M. B., & Sadjadi, S. J. (2008). Integrating goal programming, Kuhn–Tucker conditions, and penalty function approaches to solve linear bi-level programming problems. *Applied Mathematics and Computation*, *195*(2), 585-590.

Toksarı, M. D. (2010). Taylor series approach for bi-level linear fractional programming problem. *Selçuk journal of applied mathematics*, *11*(1), 63-69.

Tran, L. N., Hanif, M. F., Tolli, A., & Juntti, M. (2012). Fast converging algorithm for weighted sum rate maximization in multicell MISO downlink. *IEEE Signal Processing Letters*, *19*(12), 872-875.

Vicente, L. N., & Calamai, P. H. (1994). Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization*, *5*(3), 291-306.

Wan, Z., Wang, G., & Sun, B. (2013). A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems. *Swarm and Evolutionary Computation*, *8*, 26-32.

Wang, Y., Li, H., & Dang, C. (2011). A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence. *INFORMS Journal on Computing*, *23*(4), 618-629.

Watada, J., Roy, A., Wang, B., Tan, S. C., & Xu, B. (2020). An artificial bee colony-based double layered neural network approach for solving quadratic bi-level programming problems. *IEEE Access*, *8*, 21549-21564.

White, D. J., & Anandalingam, G. (1993). A penalty function approach for solving bi-level linear programs. *Journal of Global Optimization*, *3*, 397-419.

Zhang, G., Han, J., & Lu, J. (2016). Fuzzy bi-level decision-making techniques: a survey. *International Journal of Computational Intelligence Systems*, *9*(sup1), 25-34.

Zhou, X., Tian, J., Wang, Z., Yang, C., Huang, T., & Xu, X. (2022). Nonlinear bilevel programming approach for decentralized supply chain using a hybrid state transition algorithm. *Knowledge-Based Systems*, *240*, 108119.