Original Research Article

Two Approachesfor SolvingNon-Linear Bi-level Programmingproblem

ABSTRACT

In the recent years, the bi-level programming problem (BLPP) is interested by many researchers and it is known as an tool to solve the real problems in several areas such as economic, traffic, finance, management, and so on. Also, it has been proven that the general BLPP is an NP-hard problem. In this paper, we attempt to develop two effective approaches, one based on approximate approach and the other based on the hybrid algorithm by combining the penalty function and the line search algorithm for solving thenon-linear BLPP. In these approaches, by using the Karush-Kuhn-Tucker conditions the BLPP is converted to a non-smooth single problem, and then it is smoothed by Fischer-Burmeister functions.Finally, the smoothed problem is solved using both of the proposed approaches. The presented approaches achieve an efficient and feasible solution in an appropriate time which has been evaluated by comparing to references and test problems.

*Keywords:*Non-linear bi-level programming problem, Approximate Method, Karush-Kuhn-Tucker conditions, Line search method.

1. Introduction

It has been proven that the bi-level programming problem (BLPP) is an NP-Hard problem [1, 2].Several algorithms have been proposed to solve BLPP [3, 4, 11, 12, 13, 21, 25, 31]. These algorithms are divided into the following classes: global techniques, enumeration methods, transformation methods, meta heuristic approaches, fuzzy methods, primal-dual interior methods. In the following, these techniques are shortly introduced.

1.1. Global techniques

All optimization methods can be divided into two distinctive classes: local and global algorithms. Local ones depend on initial point and characteristics such as continuity and differentiability of the objective function. These algorithms search only a local solution, a point at which the objective function is smaller than at all other feasible points in vicinity. They do not always find the best minima, that is, the global solution. On the other hand, global methods can achieve global optimal solution. These methods are independent of initial point as well as continuity and differentiability of the objective function [9, 10, 11, 12, 33].

1.2. Enumeration methods

Branch and bound is an optimization algorithm that uses the basic enumeration. But in these methods we employ clever techniques for calculating upper bounds and lower bounds on the objective function by reducing the number of search steps.

In these methods, the main idea is that the vertex points of achievable domain for BLPP are basic feasible solutions of the problem and the optimal solution is among them [14].

1.3. Transformation methods

An important class of methods for constrained optimization seeks the solution by replacing the original constrained problem with a sequence of unconstrained sub-problems or a problem with simple constraints. These methods are interested by some researchers for solving BLPP, so that they transform the follower problem by methods such as penalty functions, barrier functions, Lagrangian relaxation method or KKT conditions. In fact, these techniques convert the BLPP into a single problem and then it is solved by other methods [3, 4, 22, 23, 32, 34, 35].

1.4. Meta heuristic approaches

Meta heuristic approaches are proposed by many researchers to solve complex combinatorial optimization. Whereas these methodsare too fast and known as suitable techniques for solving optimization problems, however, they can only propose a solution near to optimal. These approaches are generally appropriate to search global optimal solutions in very large space whenever convex or non-convex feasible domain is allowed. In these approaches, BLPP is transformed to a single level problem by using transformation methods and then meta heuristic methods are utilized to find out the optimal solution [15, 16, 17, 18, 19, 25, 36, 37, 38, 39].

1.5. Fuzzy methods

Sometimes crisp values to the variables are not appropriate. Therefore,the fuzzy approach is asuitable tool to describe them. In this category,membership functions can be leader, follower or both of objective functions.Also it can be define with constraints and variables. There are so many researchers using this method [5, 6, 7, 8, 24, 40].

1.6. Interior pointmethods

The interior point methods formulate many large linear programs as nonlinear problems and solve them with various modifications of nonlinear algorithms. These methods require all iterates to satisfy the inequality constraints in the problem strictly. The primal-dual method is a class of these methods which is the most efficient practical approach. The interior point methods can be strong competitors to the simplex method on large problems [13].

The remainder of the paper is structured as follows: in Section 2, basic concepts of the linear BLPP are introduced. We provide a smooth method to BLPP in Section 3. The first presented algorithm is proposed in Section 4. We will present the second proposed algorithm in Section 5 and computational results are presented for both approaches in Section 6.Finally, the paper is finished in Section 7 by presenting the concluding remarks.

2.The Non-Linear BLPP and Smoothing Method

The BLPP is used frequently by problems with decentralized planning structure. It is defined as [20]:

$$
\begin{aligned}
&\min_{x} F(x,y) \\
&s.t \min_{y} f(x,y) \qquad (1) \\
&s.t\ g(x,y) \leq 0,
\end{aligned}
$$

$x, y \geq 0.$

Where

$F: R^{n\times m} \dot{\to} R^1, f: R^{n\times m} \dot{\to} R^1,$

$g: R^{n\times m} \dot{\to} R^q, x \in R^n, y \in R^m.$

Also F and f are objective functions of the leader and follower respectively.

The feasible region of the non-linear BLP problem is

$$S = \{(x, y) | g(x, y) \leq 0, x, y \geq 0\} \tag{2}$$

On using KKT conditions the problem (1) can be converted into the following problem:

$$\begin{aligned} &\min_{x,y\mu} F(x, y, \mu) \\ &s.t\ \nabla_y L(x, y, \mu) = 0, \\ &\mu g(x, y) = 0, \\ &g(x, y) \leq 0, \\ &\mu \geq 0. \end{aligned} \tag{3}$$

Where L is the Lagrange function and $L(x, y, \mu) = f(x, y) + \mu g(x, y)$.

Because problem (3) has a complementary constraint, it is not convex and it is not differentiable. Fortunately Facchinei et al, 1999 proposed smooth method for solving problem with complementary constraints and we use this method to smooth problem (3).

In general the BLPP is a non-convex optimization problem therefore there is no general algorithm to solve it. This problem can be non-convex even when all functions and constraints are bounded and continuous.

A summary of important properties for convex problem as follows, which $f: S \dot{\to} R^n$ and S is a nonempty convex set in R^n.

(1) The convex function f is continuous on the interior of S.

(2) Every local optimal solution of f over a convex set $X \subseteq S$ is the unique global optimal solution.

(3) If $\nabla f(\bar{x}) = 0$, then $\bar{x}$ is unique global optimal solution of f over S.

Since in problem (3), most of the equality constraints are not linear then it concerns that the above problem is a non-convex programming problem, which indicates there are local optimal solutions that are not global solutions. Therefore solving the problem (3) will be complicated.

Definition 2.1:

Fischer – Burmeister is the following function,

$\phi: R^2 \to R$, $\phi(a,b) = a + b - \sqrt{a^2 + b^2}$ or $\phi: R^3 \to R$, $\phi(a,b,\mathcal{E}) = a + b - \sqrt{a^2 + b^2 + \mathcal{E}}$, where $a \geq 0$, $b \geq 0$, then $ab = 0 \leftrightarrow \phi(a,b,\mathcal{E}) = 0$.

Using Fischer–Burmeister function$\phi(a,b,\mathcal{E}) = a + b - \sqrt{a^2 + b^2 + \mathcal{E}}$ in problem (3) we obtain the followingproblem:

$$\begin{aligned} &\min F(x,y,\mu) \\ &s.t\ \nabla_y L(x,y,\mu) = 0, \\ &\mu_i - g_i(x,y) - \sqrt{\mu_i^2 + g_i^2(x,y) + \varepsilon} = 0, i = 1,2,\dots,m, \\ &x, y, \mu_i \geq 0, i = 1,\dots,m. \end{aligned} \tag{4}$$

Which $g_i(x,y) = a^i x + b^i y - r$, and a^i, b^i are i-th row of A, B respectively, and$a = \mu_i \geq 0, b = -g_i(x,y) \geq 0$.

Let:

$$G(x,y,\mu) = \begin{bmatrix} \mu_1 - g_1(x,y) - \sqrt{\mu_1^2 + g_1^2(x,y) + \mathcal{E}} \\ \mu_2 - g_2(x,y) - \sqrt{\mu_2^2 + g_2^2(x,y) + \mathcal{E}} \\ \vdots \\ \mu_m - g_m(x,y) - \sqrt{\mu_m^2 + g_m^2(x,y) + \mathcal{E}} \end{bmatrix}, \quad H(x,y,\mu) = \nabla_y L(x,y,\mu). \tag{5}$$

Problem (4) can be written as follows,

$$\begin{aligned} &\min F(x,y,\mu) \\ &s.t\ H(x,y,\mu) = 0, \\ &G(x,y,\mu) = 0, \\ &x,y,\mu \geq 0. \end{aligned} \tag{6}$$

Where $t = (x, y, \mu)$

3. Hybrid algorithm (HA)

Penalty functions transform a constrained problem into a single unconstrained problem or into a sequence of unconstrained problems. The constraints are appended into the objective function via a penalty parameter in a way that penalizes any violation of the constraints. In general, a suitable function must incur a positive penalty for infeasible points and no penalty for feasible points. Also, the penalty function method is a common approach to solve the bi-level programming problems. In this kind of approach, the lower level problem is appended to the upper level objective function with a penalty. We use a penalty function to convert problem (6) to an unconstraint problem.

Consider problem (6); we append all constraints to the upper level objective function with a penalty for each constraint. Then, we obtain the following penalized problem.

$$\min F(x,y,\mu) + \mu_1 H(x,y,\mu) + \sum_i \mu_i (G_i(t))^2 \tag{7}$$

Which$G_j(t)$ is ith row of matrix $G(t)$ and μ_i is taken as the penalty coefficient.

Now we solve problem (7) using our line search method. The line search method is proposed as follows:

Given a vector x, a suitable direction d is first determined, and then f is minimized from x in the direction d. Our method searches along the directions $(d_1, d_2, \ldots, d_{n-1})$ where $d_j, j = 1,2, \ldots, n-1$ is a vector of zeros except at the jth position which is 1 and$d_n = \left(\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}},, \ldots, \frac{1}{\sqrt{n}}\right)$.

Clearly, all directions have a norm equal to 1 and they are linearly independent search directions. In fact, the proposed line search method uses the following directions as the search directions:

$$d_1 = (1,0, \ldots,0), d_2 = (0,1, \ldots,0), \ldots, d_{n-1} = (0, \ldots,1,0), d_n = \left(\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}},, \ldots, \frac{1}{\sqrt{n}}\right) \quad (8)$$

Therefore, along the search direction$d_j, j = 1,2, \ldots, n-1$, the variable x_j is changed while all other variables are kept fixed. We summarize below the proposed line search method for minimizing a function of several variables. Then, we show that, if the function is differentiable then the proposed method converges to a stationary point.

Step 1: Initial step

Choose a scalar $\mathcal{E} > 0$to be used for terminating the algorithm, and let $d_1, d_2, \ldots, d_{n-1}$ be the coordinate directions and d_n be a vector of $\frac{1}{\sqrt{n}}$. Choose an initial point x_1 let $x_1 = y_1. k = j = 1$, and go to the next step.

Step 2: Main step

Let μ_j be an optimal solution to the problem to minimize$(y_j + \mu d_j)$, and let $y_{j+1} = y_j + \mu_j d_j$

If $j < n$ replace $j$by $j + 1$, and repeat step1. Otherwise, if $j = n$, go to the next step.

Step 3: Termination

Let $x_{k+1} = y_{n+1}$ if $\|x_{k+1} - x_k\| < \varepsilon$ then stop, otherwise, let $y_1 = x_{k+1}$and $j = 1$, replace k by $k + 1$, and repeat step 2.

We now propose a theorem which establishes the convergence of algorithms for solving a problem of the form: minimize $f(x)$ subject to $x \in R^n$. We show that an algorithm that generates n linearly independent search directions, and obtains a new point by sequentially minimizing f along these directions, converges to a stationary point. The theorem also establishes the convergence of algorithms using linearly independent and orthogonal search directions.
same optimal solution according to the following theorem.

Theorem 3.1:

Consider the following problem:

$$\begin{aligned} &\min_x f(x) \\ &s.t\ g_i(x) \le 0,\ \text{i=1,2,…,m}, \\ &h_j(x) = 0,\ \text{j=1,2,…,l}, \end{aligned} \quad (9)$$

where$f, g_1, \ldots, g_m,\ h_1, \ldots, h_l$ are continuous functions on R^n and X is a nonempty set in R^n. Suppose that the problem has a feasible solution, and α is a continuous function as follows:

$$\alpha(\mathrm{x}) = \sum_{i=1}^{m} \emptyset[g_i(x)] + \sum_{i=1}^{l} \emptyset[h_i(x)] \quad (10)$$

where

$$\emptyset(y) = 0 \text{ if } y \le 0, \qquad \emptyset(y) > 0 \text{ } if \text{ } y > 0. \quad (11)$$

$$\emptyset(y) = 0 \text{ if } y = 0, \qquad \emptyset(y) > 0 \text{ } if \text{ } y \ne 0. \quad (12)$$

Then,

$$\begin{aligned} &\inf\{f(x): g(x) \le 0, \ h(x) = 0, x \in X\} \\ &\qquad = \inf\{f(x) + \mu\alpha(x): x \in X\} \end{aligned} \quad (13)$$

whereμ is a large positive constant ($\mu \to \infty$).

4. Taylor method (TA)

Because functions G, H in (6) is always continuous everywhere and it is possible to use, Taylor Theorem for them in (6) and F should be continuous too.

Theorem 4.1 (TaylorTheorem)[30]:Suppose that f has $n+1$ continuous derivatives on an open interval containing a. Then for each$x$ in the interval,

$$\mathrm{f(x)} = \left[\sum_{k=0}^{n} \frac{\mathrm{f^k(a)}}{\mathrm{k!}}(\mathrm{x}-\mathrm{a})^{\mathrm{k}}\right] + \mathrm{R_{n+1}(x)}$$

where the error term $\mathrm{R_{n+1}(x)}$, for some c between a and x, satisfies

$$\mathrm{R_{n+1}(x)} = \frac{\mathrm{f^{(n+1)}(c)}}{\mathrm{(n+1)!}}(\mathrm{x}-\mathrm{a})^{\mathrm{n+1}}$$

This form for the error $R_{n+1}(x)$is called the Lagrange formula for the reminder.

The infinite Taylor series converge to f,

$$\mathrm{f(x)} = \left[\sum_{k=0}^{\infty} \frac{\mathrm{f^k(a)}}{\mathrm{k!}}(\mathrm{x}-\mathrm{a})^{\mathrm{k}}\right]$$

If and only if$\lim_{n\to\infty} R_{n+1}(x) = 0$.

Proof:

The proof of this theorem was given by [28].

In mathematics, an approximation of a k-times differentiablefunction near a point is given by Taylor's theorem. Taylor's theorem is one of the fundamental tools in pure mathematics and it is the starting point of advanced asymptotic analysis, also it is usually used in applied fields of mathematics. If a real-valued functionf is differentiable at the point "a" then it has a linear approximation at the point "a". This means that there exists a function g such that

$$\mathrm{f(x)} = \mathrm{f(a)} + \mathrm{f'(a)(x-a)} + \mathrm{g(x)(x-a)}, \ \lim_{\mathrm{x}\to\mathrm{a}} \mathrm{g(x)} = 0.$$

Here

$$P_1(x) = f(a) + f'(a)(x - a)$$

Which $P_1(x)$is the linear approximation of f at the point “a” .

By applying Taylor theorem at “a” feasible point such as t^k for function G, H, F and take only two linear part of them, the following linear functions is constructed:

$$G_i(t^k) + \nabla G_i(t^k)(t - t^k) = 0, \quad i = 1,2, \ldots m.$$
$$H_i(t^k) + \nabla H_i(t^k)(t - t^k) = 0, \quad i = 1,2, \ldots m \quad (14)$$
$$F_i(t^k) + \nabla F_i(t^k)(t - t^k) = 0, \quad i = 1,2, \ldots m$$

Because the obtained problem by using Taylor theorem is linear programming, it can be solved using simplex methods. The steps of the proposed algorithm are as follows:

Step 1: Initialization

The feasible point t^1 is created randomly, error $\mathcal{E}_1$ is given and suppose k=1.

$\mathcal{E}_1$ is a small and appropriate given error and finishing the algorithm depends to $\mathcal{E}_1$ such that it is finished whenever difference between produced solutions by the algorithm in two consecutive iterations is less than $\mathcal{E}_1$.

Step 2: finding solution.

According to the step 1, k=1 and feasible solutiont^1 has been defined. Using these assumptions and Taylor theorem for $G(t), H(t)$ *and* $F(t)$at t^k, we obtain following problem:

$$\begin{aligned} \min \quad & F_i(t^k) + \nabla F_i(t^k)(t - t^k) \\ \text{s.t}\ & H_i(t^k) + \nabla H_i(t^k)(t - t^k) = 0, \quad i = 1,2, \ldots m \\ & G_i(t^k) + \nabla G_i(t^k)(t - t^k) = 0, \quad i = 1,2, \ldots m. \\ & x, y, \mu_i \geq 0, i = 1, \ldots, m. \end{aligned} \quad (15)$$

Solve the problem (15) using simplex method (by MATLAB 7.1). By solving this problem, an optimal solution such as t^{k+1} is obtained.

Step 3: Keeping the present best solution.

Because (15) is an approximation for (6) by Taylor theorem, therefore optimal solution for (15) is an approximation of optimal solution for (6). Thus t^{k+1} can be a good approximation of problem (6) optimal solution. Therefore let $t^* = t^{k+1}$ and go to next step.

Step 4: Termination

If $d(F(t^{k+1}), F(t^k)) < \mathcal{E}_1$ then the algorithm is finished andt^* is the best solution by the proposed algorithm. Otherwise, let k=k+1 and go to the step 2. Which d is metric and,

$$d\left(F(t^{k+1}), F(t^k)\right) = \left(\sum_{i=1}^{n+2m}\left(F\left(t_i^{k+1}\right) - F(t_i^k)\right)^2\right)^{\frac{1}{2}}.$$

Following theorems show that proposed algorithm is convergent.

Theorem 4.2: Every Cauchy sequence in real line and complex plan is convergent.

Proof:

Proof of this theorem is given in [34].

Theorem 4.3: Sequence $\{F_k\}$ which was proposed in above algorithm is convergent to the optimal solution, so that the algorithm is convergent.

Proof:

Let $(F_l) = (F(t^l)) = (F(t_1^l), F(t_2^l), \dots, F(t_{n+2m}^l)) = (F_1^{(l)}, F_2^{(l)}, \dots, F_{n+2m}^{(l)})$.

According to step 4

$$d(F_{k+1}, F_k) = \mathrm{d}\left(\mathrm{F}\left(\mathrm{t}^{k+1}\right), \mathrm{F}\left(\mathrm{t}^{k}\right)\right) = \left(\sum_{i=1}^{n+2m} \left(F\left(t_i^{k+1}\right) - F(t_i^k)\right)^2\right)^{\frac{1}{2}} < \varepsilon_1 \qquad (21)$$

therefore $\left(\sum_{i=1}^{n+2m} \left(F\left(t_i^{k+1}\right) - F\left(t_i^k\right)\right)^2\right) < {\varepsilon_1}^2$. There is large number such as N which k+1>k>N and j=1,2,…,2m+n we have:

$$(F_j^{(k+1)} - F_j^{(k)})^2 < {\varepsilon_1}^2 \text{, therefore } \left|F_j^{(k+1)} - F_j^{(k)}\right| < \varepsilon_1$$

$$\textit{Now let } m = k + 1, r = k \textit{ then we have}$$

$$\forall_{m>r>N} \left|F_j^{(m)} - F_j^{(r)}\right| < \varepsilon_1.$$

This shows that for each fixed $\mathrm{j}, (1 \leq \mathrm{j} \leq 2\mathrm{m} + \mathrm{n})$, the sequence $(\mathrm{F}_\mathrm{j}^{(1)}, \mathrm{F}_\mathrm{j}^{(2)}, \dots)$ is Cauchy of real numbers, then it converges by theorem 4.6.

Say, $F_j^{(m)} \to F_j$ as $m \to \infty$. Using these 2m+n limits, we define $F = (F_1, F_2, \dots, F_{2m+n})$.From (21) and m=k+1, r=k,

$$d(F_m, F_r) < \varepsilon_1$$

Now if $\mathrm{r} \to \infty$, by$F_r \to \mathrm{F}$ we have $d(F_m, F) \leq \varepsilon_1$.

This shows that F is the limit of (F_m) and the sequence is convergent.

Theorem 4.4: If sequence $\{\mathrm{f}(\mathrm{t}_\mathrm{k})\}$ is converge to f(t) and f be linear function then $\{\mathrm{t}_\mathrm{k}\}$ is converge to t.

Proof:

Proof of this theorem is given in [34].

5. Computational results

Example 1[30] (solving by hybrid algorithm (HA)):

Consider the following linear bi-level programming problem:

$$\min_{x} \; x^2 + (y-10)^2$$

$$s.t \; \min_{y \geq 0} \; (x + 2y - 30)^2$$

$$s.t \quad x - y^2 \geq 0,$$

$$20 - x - y^2 \geq 0$$

$$0 \leq x \leq 15.$$

Using KKT conditions the following problem is obtained:

$$
\begin{aligned}
\min_{x}\ & x^2+(y-10)^2 \\
s.t\ & 4(x+2y-30)=0, \\
& 2y(\lambda_1+\lambda_2)=0, \\
& \lambda_1(y^2-x)=0, \\
& \lambda_2(y^2+x-20)=0, \\
& \lambda_3(x-15)=0 \\
& y^2-x\le 0, \\
& y^2+x-20\le 0, \\
& x-15\le 0, \\
& \lambda_1,\lambda_2,\lambda_3\ge 0.
\end{aligned}
$$

Using theFischer – Burmeister function, the above problem as follows:

$$
\begin{aligned}
\min\ & x^2+(y-10)^2 \\
s.t\ & 4(x+2y-30)=0, \\
& (\lambda_1+\lambda_2)-2y-\sqrt{(\lambda_1+\lambda_2)^2+(2y)^2+\varepsilon}=0, \\
& \lambda_1-(y^2-x)-\sqrt{\lambda_1^2+(y^2-x)^2+\varepsilon}=0, \\
& \lambda_2-(y^2+x-20)-\sqrt{\lambda_2^2+(x+y^2-20)^2+\varepsilon}=0, \\
& \lambda_3-(x-15)-\sqrt{\lambda_3^2+(x-15)^2+\varepsilon}=0,
\end{aligned}
$$

Using (7) we obtain an unconstraint problem as follows:

$$
\begin{aligned}
\min\ & x^2+(y-10)^2+4\mu_1(x+2y-30)^2+\mu_2(\lambda_1+\lambda_2-2y-\sqrt{(\lambda_1+\lambda_2)^2+(2y)^2+\varepsilon})^2+ \\
& \mu_3(\lambda_1-(y^2-x)-\sqrt{\lambda_1^2+(y^2-x)^2+\varepsilon})^2+\mu_4(\lambda_2-(y^2+x-20)-\sqrt{\lambda_2^2+(x+y^2-20)^2+\varepsilon})^2 \\
& +\mu_5(\lambda_3-(x-15)-\sqrt{\lambda_3^2+(x-15)^2+\varepsilon})^2
\end{aligned}
$$

We solve this problem using the proposed line search algorithm and we present the optimal solution in the Table 2.

Example 2[30] (solving by hybrid algorithm (HA)):

Consider the following linear bi-level programming problem.

$$
\begin{aligned}
\min_{x}\ & -x_1^2-2x_1+x_2^2-2x_2+y_1^2+y_2^2 \\
s.t\ & \min\ y_1^2-2x_1y_1+y_2^2-2x_2y_2 \\
& \quad s.t\ .25-(y_1-1)^2\ge 0, \\
& \quad\quad .25-(y_2-1)^2\ge 0.
\end{aligned}
$$

After applying KKT conditions and smoothing method, and then proposed penalty function in(7)above problem will be transformed to the following problem:

$$\begin{aligned}\min_{x}\ & -x_1^2 - 2x_1 + x_2^2 - 2x_2 + y_1^2 + y_2^2 + \mu_1(2y_1 - 2x_1 + 2y_2 - 2x_2) \\ & + \mu_2(\lambda_1 - (y_1-1)^2 + .25 - \sqrt{\lambda_1^2 + ((y_1-1)^2 + .25)^2 + \varepsilon}\ )^2 \\ & + \mu_3(\lambda_2 + .25 - (y_2-1)^2 - \sqrt{\lambda_2^2 + ((y_2-1)^2 + .25)^2 + \varepsilon}\ )^2\end{aligned}$$

The optimal solution is obtained using our line search method according to the Table 3.

More problems with different sizes have been solved by our approach and computation results have been proposed in Table 4. References of the examples in Table 4 are as follows:

Example 3 [30], Example 4 [32], Example 5 [31], Example 6 [33] which both of them are minimization problems .

According to the Table 4, the best solutions by our algorithm are better than the best solution by the references. The algorithm is feasible and efficient according to the Tables.

Example 1 [4] (solving by Taylor algorithm (TA)):

Consider the following non-linear bi-level programming problem:

$$\begin{aligned}\min_{x}\ & x^2 + (y-10)^2 \\ s.t\ & \min_{y\geq 0}\ (x+2y-30)^2 \\ & \quad s.t\quad x - y^2 \geq 0, \\ & \qquad\quad 20 - x - y^2 \geq 0 \\ & \qquad\quad 0 \leq x \leq 15.\end{aligned}$$

Using KKT conditions and the Fischer – Burmeister function, the following problem is obtained:

$$\begin{aligned}\min\ & x^2 + (y-10)^2 \\ s.t\ & 4(x+2y-30) = 0, \\ & (\lambda_1 + \lambda_2) - 2y - \sqrt{(\lambda_1+\lambda_2)^2 + (2y)^2 + \varepsilon} = 0, \\ & \lambda_1 - (y^2 - x) - \sqrt{\lambda_1^2 + (y^2 - x)^2 + \varepsilon} = 0, \\ & \lambda_2 - (y^2 + x - 20) - \sqrt{\lambda_2^2 + (x + y^2 - 20)^2 + \varepsilon} = 0, \\ & \lambda_3 - (x - 15) - \sqrt{\lambda_3^2 + (x-15)^2 + \varepsilon} = 0,\end{aligned}$$

We solve this problem using the proposed line search algorithm and we present the optimal solution in Table 1. By solving this problem the best solutions are found according to Table 1. It declares that the best solutions by the proposed algorithm are better than the best solution by the references in appropriate time.

Behavior of the variables in Example 1 has been show in figure 1 that variables x and y will be stable after 5000 and 4850 iterations respectively.

Example 2[4] (solving by Taylor series approach (TA)):

Consider the following linear bi-level programming problem.

$$\min_x \; -x_1^2 - 2x_1 + x_2^2 - 2x_2 + y_1^2 + y_2^2$$
$$s.t \quad \min_y \; y_1^2 - 2x_1y_1 + y_2^2 - 2x_2y_2$$
$$s.t \quad .25 - (y_1 - 1)^2 \geq 0,$$
$$.25 - (y_2 - 1)^2 \geq 0.$$

After applying KKT conditions and smoothing method, and then proposed penalty function above problem will be transformed to the following problem:

$$\min_x \; -x_1^2 - 2x_1 + x_2^2 - 2x_2 + y_1^2 + y_2^2$$
$$s.t \; +\mu_1(2y_1 - 2x_1 + 2y_2 - 2x_2)$$
$$+\mu_2(\lambda_1 - (y_1-1)^2 + .25 - \sqrt{\lambda_1^2 + ((y_1-1)^2 + .25)^2 + \varepsilon}\;)^2$$
$$+\mu_3(\lambda_2 + .25 - (y_2-1)^2 - \sqrt{\lambda_2^2 + ((y_2-1)^2 + .25)^2 + \varepsilon}\;)^2$$

The optimal solution is obtained using our method according to Table 2.

Behavior of the variables in Example 2 has been show in figure 2 that variableswill be stable after 3000iterations respectively.

More problems with different sizes have been solved by our approach and computation results have been proposed in Table 3. According to this Table, the best solutions by our algorithm are better than the best solution by the references. The algorithm is feasible and efficient according to the Tables.

We make program with MATLAB 7.1 and use a personal computer (CPU: Intel (R) Celeron(R) 1000 M @ 1.8 GHz, RAM:4 GB) to execute the program.References of the examples in Table 3 as follows:

Example 3 [3], Example 4 [7], Example 5 [26], Example 6 [27] .

7. Conclusion and future work

In this paper, we used the KKT conditions to convert the problem into a single level problem. Then,usingtheFischer-Burmeister function, the problem was made simpler and converted to a smooth programming problem. The smoothed problem was been solved,utilizing the first proposed algorithm based on Taylor theorem. Also, it was solved using the second proposed hybrid algorithm by combining the penalty function and the line search algorithm.Comparing with the results of previous methods, both algorithms have better numerical results and present better solutions in much less times. The bestsolutions produced by proposed algorithms are feasible unlike the previous best solutions by other researchers.

In the future works, the following should be researched:

(1) Examples in larger sizes can be supplied to illustrate the efficiency of the proposed algorithms.

(2) Showing the efficiency of the proposed algorithms for solving other kinds of BLP.

Best solution by our method $\varepsilon = 0.001$		Best solution according to reference [30]		Optimal solution	
(x^*, y^*)	z^*	(x^*, y^*)	z^*	(x^*, y^*)	z^*
(2.601,1.611)	-77.14	(2.600,1.613)	-77.10	(2.600,1.612)	-77.11

Table 1 comparison optimal solutions in HA- Example 1

Best solution by our method $\varepsilon = 0.001$		Best solution according to reference [4]		Optimal solution	
(x^*, y_1^*, y_2^*)	z^*	(x^*, y_1^*, y_2^*)	z^*	(x^*, y_1^*, y_2^*)	z^*
(0.51,0.51,0.49,0.50)	-1.590	(0.5,0.5,0.5,0.5)	-1.5	(0.51,0.51,0.51,0.51)	-1.598

Table 2 comparison optimal solution in HA Example 2

	Best solution according to reference [3, 7, 26, 27]	Best solution by our method $\varepsilon = 0.001$	Iterations	Time	Optimal solution
Example 3	(1.883,0.891,0.003)	(1.887,0.889,0.001)	8250	3.57 s	$(17/9, 8/9, 0)$
Example 4	(0,0)	(0,0)	3500	2.30 s	(0,0)
Example 5	(1,0)	(1,0)	6700	3.20 s	(1,0)
Example 6	(0,0.75,0,0.5,0)	(0.001,0.73,0,0.54,0)	8500	4.10 s	(0,0.75,0,0.5,0)

Table 3 comparison optimal solutions with deferent Examples 3-6by HA

Best solution by our method $\varepsilon = 0.001$		Best solution according to reference [30]		Optimal solution	
(x^*, y^*)	z^*	(x^*, y^*)	z^*	(x^*, y^*)	z^*
(2.6,1.61)	-77.12	(2.600,1.613)	-77.10	(2.600,1.612)	-77.11

Table 4 comparison optimal solutions in TA - Example 1

Best solution by our method $\varepsilon = 0.001$		Best solution according to reference [32]		Optimal solution	
(x^*, y_1^*, y_2^*)	z^*	(x^*, y_1^*, y_2^*)	z^*	(x^*, y_1^*, y_2^*)	z^*
(0.52,0.51,0.53,0.51)	-1.583	(0.5,0.5,0.5,0.5)	-1.5	(0.51,0.51,0.51,0.51)	-1.598

Table 5 comparison optimal solution in TA Example2

	Best solution according to reference [3, 7, 26, 27]	Best solution by our method $\varepsilon = 0.001$	Iterations	Time	Optimal solution
Example 3	(1.883,0.891,0.003)	(1.88,0.87,0)	7100	3.05 s	$(17/9, 8/9, 0)$
Example 4	(0,0)	(0,0)	2800	1.46 s	(0,0)
Example 5	(1,0)	(1,0)	5000	2.51 s	(1,0)
Example 6	(0,0.75,0,0.5,0)	(0,0.76,0,0.51,0)	7300	3.15 s	(0,0.75,0,0.5,0)

Table 6 comparison optimal solutions with deferent Examples 3-6 by TA

	Example 1			Example 2		
	Gap of Optimal Solution	Iterations	Time	Gap of Optimal Solution	Iterations	Time
TA	0	4000	2.16 s	0.006	2000	1.37 s
HA	0.1	7000	3.05 s	0.04	7000	2.54 s

Table 7- Comparison of TA and HA

References

[1] J.F. Bard, Some properties of the bi-level linear programming, Journal of OptimizationTheory and Applications (1991)68 371–378.

[2] L. Vicente, G. Savard, J. Judice, Descent approaches for quadratic bi-level programming, Journal of Optimization Theory and Applications(1994) 81 379–399.

[3] Lv. Yibing, Hu. Tiesong, Wang. Guangmin , A penalty function methodBased on Kuhn–Tucker condition for solving linear bilevel programming, Applied Mathematics and Computation (2007)188 808–813.

[4] G. B. Allende, G. Still, Solving bi-level programs with the KKT-approach, Springer and Mathematical Programming Society(2012)131:37 – 48.

[5] M. Sakava, I. Nishizaki, Y. Uemura, Interactive fuzzy programming for multilevel linear programming problem, Computers & Mathematics with Applications(1997) 36 71–86.

[6] S Sinha, Fuzzy programming approach to multi-level programming problems,Fuzzy Sets And Systems (2003) 136 189–202.

[7] S. Pramanik, T.K. Ro, Fuzzy goal programming approach to multilevel programming problems, European Journal of Operational Research (2009) 194 368–376.

[8] S.R. Arora, R. Gupta, Interactive fuzzy goal programming approach for bi-level programming problem, European Journal of Operational Research (2007)176 1151–1166.

[9] J. Nocedal, S.J. Wright, 2005 Numerical Optimization, Springer-Verlag, , New York.

[10]A.AL Khayyal, Minimizing a Quasi-concave Function Over a Convex Set: A Case Solvable by Lagrangian Duality, proceedings, I.E.E.E. International Conference on Systems, Man,andCybemeties, Tucson AZ (1985) 661-663.

[11] R. Mathieu, L. Pittard, G. Anandalingam, Genetic algorithm based approach to bi-level Linear Programming, Operations Research(1994) 28 1–21.

[12] G. Wang, B. Jiang, K. Zhu, (2010) Global convergent algorithm for the bi-level linear fractional-linear programming based on modified convex simplex method, Journal of Systems Engineering and Electronics 239–243.

[13]W. T. Wend, U. P. Wen, (2000) A primal-dual interior point algorithm for solving bi-level programming problems, Asia-Pacific J. of Operational Research, 17.

[14]N. V. Thoai, Y. Yamamoto, A. Yoshise, (2002) Global optimization method for solving mathematical programs with linear complementary constraints, Institute of Policy and Planning Sciences, University of Tsukuba, Japan 978.

[15]S.R. Hejazi, A. Memariani, G. Jahanshahloo, (2002) Linear bi-level programming solution by genetic algorithm, Computers & Operations Research 29 1913–1925.

[16] G. Z. Wang, Wan, X. Wang, Y.Lv,Genetic algorithm based on simplex method for solving Linear-quadratic bi-level programming problem,Computers and Mathematics with Applications(2008) 56 2550–2555.

[17] T. X. Hu, Guo, X. Fu, Y. Lv, (2010) A neural network approach for solving linear bi-level programming problem, Knowledge-Based Systems 23 239–242.

[18] B. Baran Pal, D .Chakraborti , P. Biswas, (2010) A Genetic Algorithm Approach to Fuzzy Quadratic Bi-level Programming, Second International Conference on Computing, Communication and Networking Technologies.

[19] Z. G.Wan, Wang, B. Sun, (2012) A hybrid intelligent algorithm by combining particle Swarm optimization with chaos searching technique for solving nonlinear bi-level programming Problems, Swarm and Evolutionary Computation.

[20]J.F. Bard, Practical bi-level optimization: Algorithms and applications, Kluwer Academic Publishers, Dordrecht, 1998.

[21] J.F. Bard, Some properties of the bi-level linear programming, Journal of Optimization Theory and Applications 68 (1991) 371–378.

[22] B.Luce, Saïd.H, Raïd.M, One-level reformulation of the bi-level Knapsack problem using dynamic programming, Discrete Optimization 10 (2013) 1–10.

[23] S. Dempe, A.B. Zemkoho, On the Karush–Kuhn–Tucker reformulation of the bi-level optimization problem, Nonlinear Analysis 75 (2012) 1202–1218.

[24] S. Masatoshi, Takeshi.M, Stackelberg solutions for random fuzzy two-level linear programming through possibility-based probability model, Expert Systems with Applications 39 (2012) 10898–10903.

[25] J. Yan, Xuyong.L, Chongchao.H, Xianing.W, Application of particle swarm optimization based on CHKS smoothing function for solving nonlinear bi-level programming problem, Applied Mathematics and Computation 219 (2013) 4332–4339.

[26] W.Zhongping, Guangmin.W, An Interactive Fuzzy Decision Making Method for a Class of Bi-level Programming, Fifth International Conference on Fuzzy Systems and Knowledge Discovery 2008.

[27] L. Kuen-Ming, Ue-Pyng.W, Hsu-Shih.S, A hybrid neural network approach to bi-level programming problems, Applied Mathematics Letters 20 (2007) 880–884

[28] F. Facchinei, H. Jiang, L. Qi, A smoothing method for mathematical programming with equilibrium constraints, Mathematical Programming 85 (1999) 107-134.

[29] M.S. Bazara, H.D. Sherali, Non-Linear Programming, Theory and Algorithm, New York .2005.

[30] A. Silverman. Richard, Calculus with analytic geometry, ISBN:978-964-311-008-6, 2000.

[31] Y.Zheng, J. Liu, Z. Wan, Interactive fuzzy decision making method for solving bi-level programming problem, Applied Mathematical Modelling, Volume 38, Issue 13, 1 July 2014, Pages 3136-3141.

[32] Y. Jiang, X. Li, C. Huang, X. Wu, An augmented Lagrangian multiplier method based on a CHKS smoothing function for solving nonlinear bi-level programming problems, Knowledge-Based Systems, Volume 55, January 2014, Pages 9-14.

[33] X. He, C. Li, T. Huang, C. Li, Neural network for solving convex quadratic bilevel programming problems, Neural Networks, Volume 51, March 2014, Pages 17-25.

[34] Z. Wan, L. Mao, G. Wang, Estimation of distribution algorithm for a class of nonlinear bilevel programming problems, Information Sciences, Volume 256, 20 January 2014, Pages 184-196.

[35] P. Xu, L. Wang, An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions, Computers & Operations Research, Volume 41, January 2014, Pages 309-318.

[36] E. Hosseini, I.NakhaiKamalabadi, A Genetic Approach for Solving Bi-Level Programming Problems, Advanced Modeling and Optimization, Volume 15, Number 3, 2013.

[37] E. Hosseini, I.NakhaiKamalabadi, Taylor Approach for Solving Non-Linear Bi-level Programming Problem, ACSIJ Advances in Computer Science: an International Journal, Volume 3, 2014.

[38] E. Hosseini, I.NakhaiKamalabadi, Solving Linear-Quadratic Bi-Level Programming and Linear-Fractional Bi-Level Programming Problems Using Genetic Based Algorithm, Applied Mathematics and Computational Intellegenc, Volume 2, 2013.

[39] E. Hosseini, I.NakhaiKamalabadi, Line Search and Genetic Approaches for Solving Linear Tri-level Programming Problem, International Journal of Management, Accounting and Economics, Accepted.

[40] N .Safaei ,M.Saraj, A new method for solving fully fuzzy linear bi-level programming problems, International Journal Of applied operation research. Vol .4 , No.1 , pp. 51-58 , winter 2014.

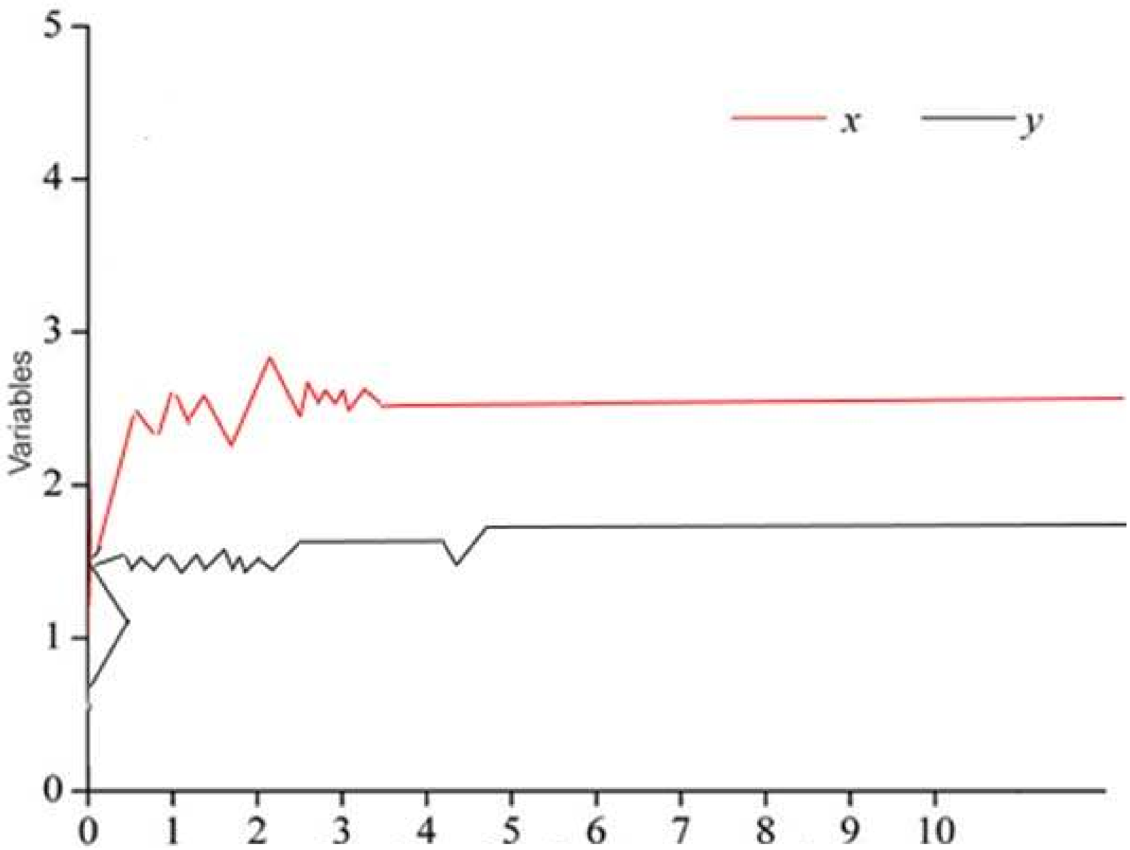


Figure 1 – The transient behavior of the variables using TA in Example 1.

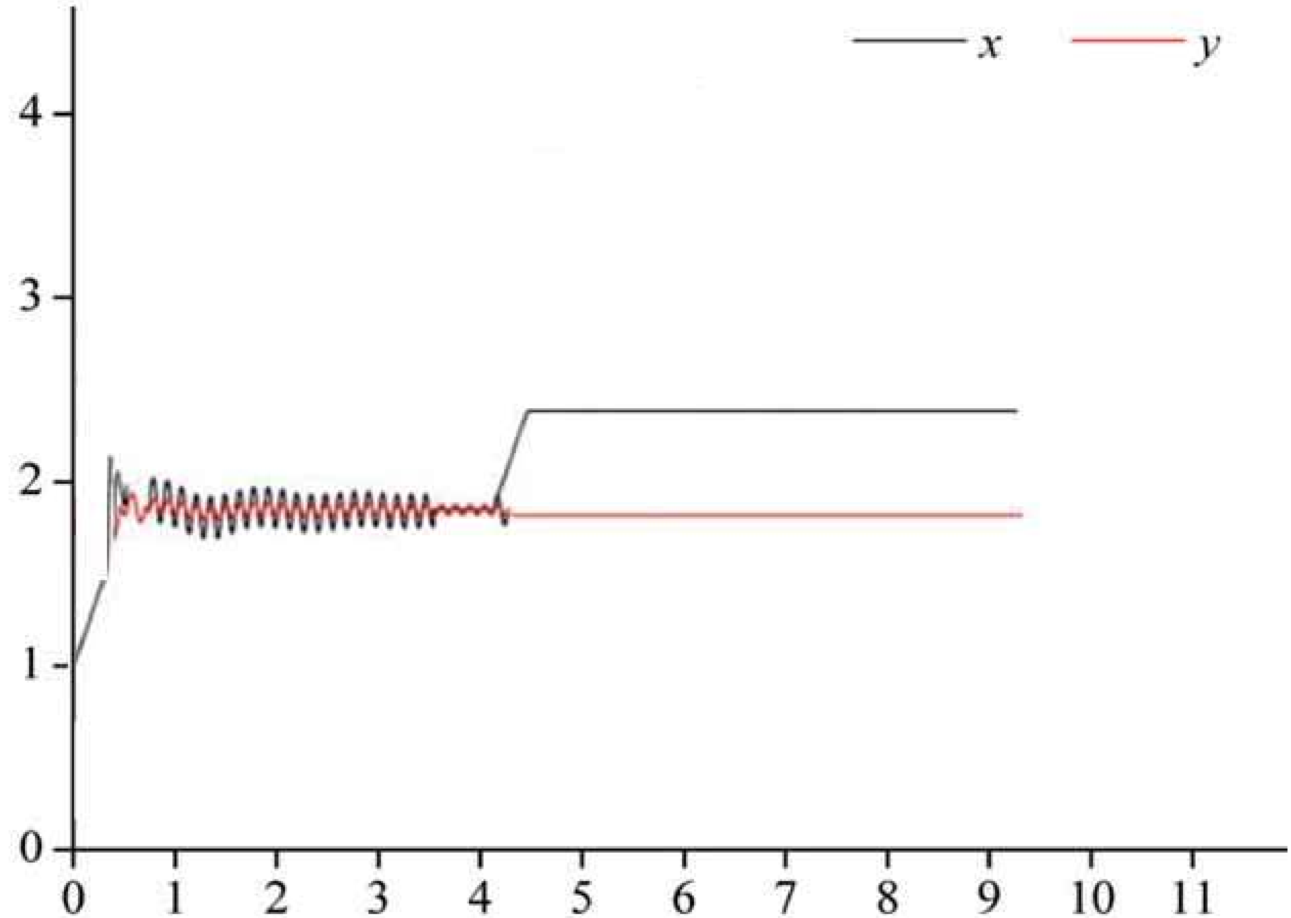
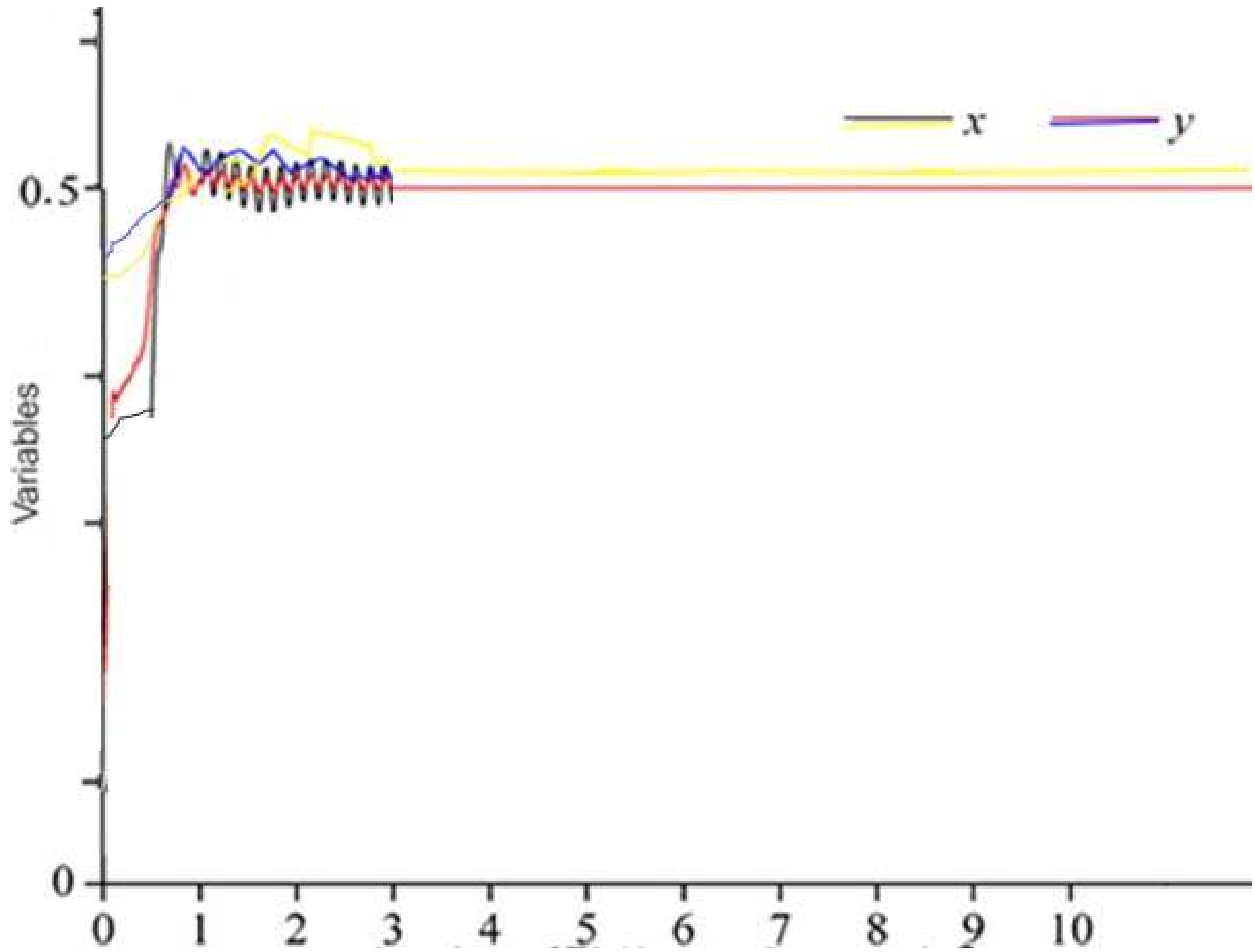


Figure 2 – The transient behavior of the variables using HA in Example 1.

385



389 **Figure 3 – The transient behavior of the variables using TA in Example 2.**

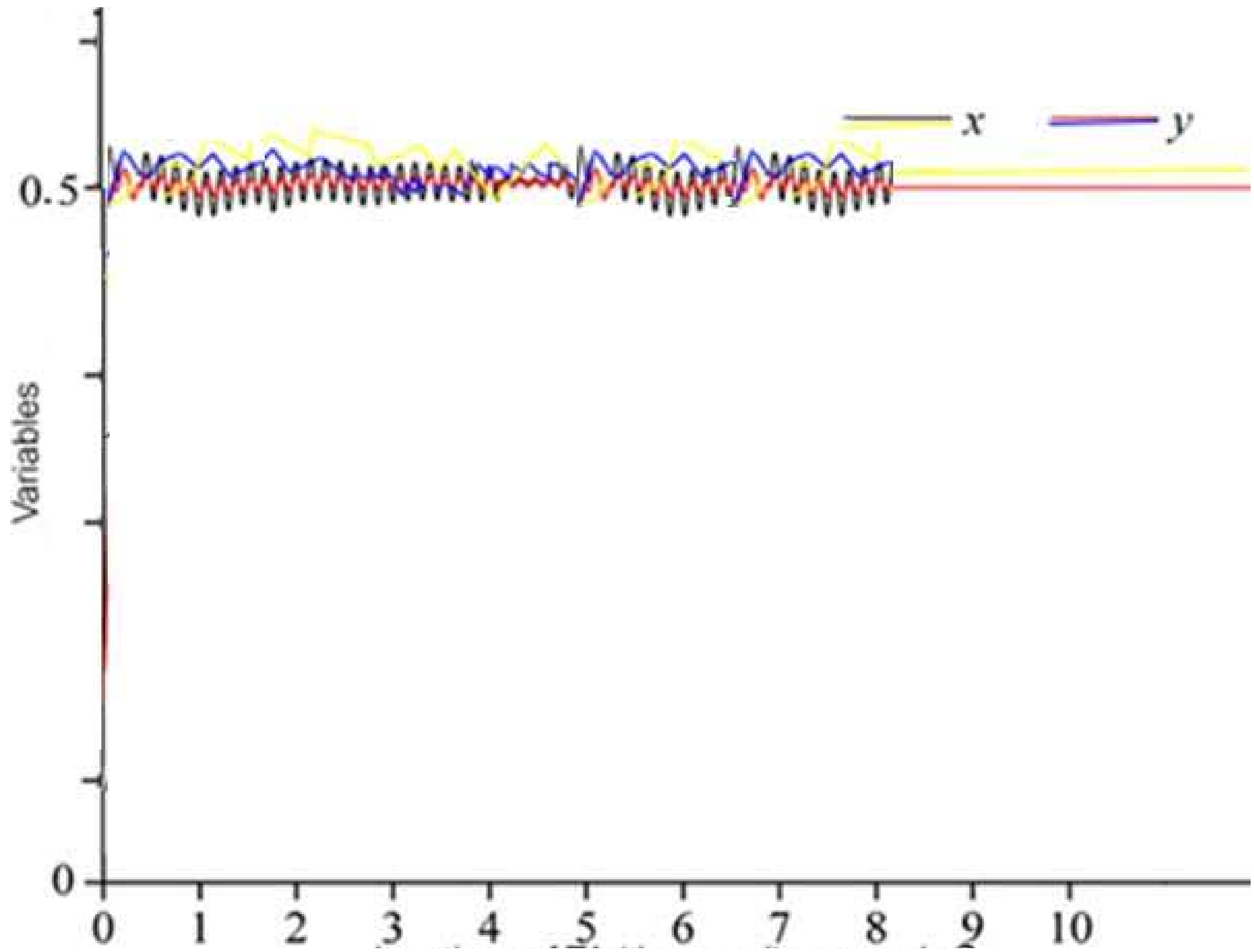


Figure 4 – The transient behavior of the variables using HA in Example 2.