Journal of Computational and Applied Mathematics 234 (2010) 505–511

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A neural network for solving a convex quadratic bilevel programming problem ☆

Yibing Lv [a,*], Zhong Chen [a], Zhongping Wan [b]

[a] School of Information and Mathematics, Yangtze University, Jingzhou 434023, PR China

[b] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, PR China

ARTICLE INFO

Article history:
Received 30 December 2008
Received in revised form 1 August 2009

Keywords:
Convex quadratic bilevel programming
Asymptotic stability
Neural network
Optimal solution

ABSTRACT

A neural network is proposed for solving a convex quadratic bilevel programming problem. Based on Lyapunov and LaSalle theories, we prove strictly an important theoretical result that, for an arbitrary initial point, the trajectory of the proposed network does converge to the equilibrium, which corresponds to the optimal solution of a convex quadratic bilevel programming problem. Numerical simulation results show that the proposed neural network is feasible and efficient for a convex quadratic bilevel programming problem.

1. Introduction

Bilevel programming (BLP) has increasingly been addressed in the literature, both from the theoretical and computational points of view [1,2]. 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 BLP problem is hard to solve. In fact, the problem has been proved to be NP-hard [3].

However, the BLP problem is used so extensively in resource allocation, finance budget, price control, transaction network, etc. [4] that many researchers have devoted attention to this field, which leads to a rapid development in theories and algorithms. For detailed expositions, the reader may consult [1,2,5,6].

It is well known that neural network has become an important computing implement, which can provide real-time optimal solutions for some practical optimization problems. In fact, there have been various types of neural networks proposed for solving linear programming, nonlinear programming, variational inequalities, etc., we cite for example [7–13].

Although there have been various types of analogue neural networks proposed for computation, there are only several reports on solving BLP problem using neural network approach. Sheng [14] firstly proposed a neural network approach based on Frank–Wolfe method for solving a class of BLP problems. However, the neural network proposed in [14] is only used to solve some linear and nonlinear programs appearing in the algorithm. Shin [15] and Lan [16] proposed neural network for solving linear BLP problem respectively. Recently, Lv [17] designed a neural network for nonlinear BLP problem. Note that the performance of the neural network proposed in [17] depending on the choice of the initial point. If the initial point chosen is not appropriate, the neural network may not get the optimal solution of the nonlinear BLP problem.

Here, for the convex quadratic BLP problem we will propose a neural network that is globally and asymptotically stable. That means, for an arbitrary initial point, the trajectory of the neural network will converge to the optimal solution of

☆ Supported by the National Natural Science Foundation of China (10926168, 70771080).

* Corresponding author.
E-mail address: Lvyibing2001@gmail.com (Y. Lv).

0377-0427/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.12.041

the convex quadratic BLP problem. The rest of the paper is organized as follows. In Section 2, following the method of replacing the lower level problem by its Kuhn–Tucker optimality conditions, we reduce the nonlinear BLP problem to a regular nonlinear programming with complementary constraints. Then we smooth the regular nonlinear programming with complementary constraints. In Section 3, a novel neural network is proposed and asymptotic stability of the equilibrium is analyzed. The numerical experiments are given in Section 4 to substantiate the theory. Finally we summarize the paper with some comments.

2. Convex quadratic BLP problem and smoothing method

Let $x \in X \subseteq R^n$, $y \in Y \subseteq R^m$, $F : R^{n\times m} \to R^1$, $f : R^{n\times m} \to R^1$, quadratic BLP problem can be written as [1]:

$$
\begin{aligned}
&(UP)\ \min_{x\geq 0} F(x,y) = \frac{1}{2}\begin{pmatrix} x^T & y^T \end{pmatrix}\begin{pmatrix} C_1 & C_3 \\ C_3^T & C_2 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + c_1 x + d_1 y \\
&(LP)\ \text{s.t.}\ \min_{y\geq 0} f(x,y) = \frac{1}{2} y^T Q y + y^T D x + d_2^T y \qquad (1)\\
&\text{s.t. } Ax + By \leq b
\end{aligned}
$$

where $c_1 \in R^n$, $d_1, d_2 \in R^m$, $C_1 \in R^{n\times n}$, $Q, C_2 \in R^{m\times m}$, $C_3^T \in R^{m\times n}$, $A \in R^{q\times n}$, $B \in R^{q\times m}$, $b \in R^q$. The term (UP) is called upper level problem and (LP) is called lower level problem, and correspondingly the terms x, y are the upper level variable and the lower level variable respectively.

Throughout the rest of the paper, we make the following assumptions:

(H_1) $C = \begin{pmatrix} C_1 & C_3 \\ C_3^T & C_2 \end{pmatrix}$ and Q are respectively positive semi-definite and positive definite matrices.

(H_2) The constraint region of the above BLP problem $S = \{(x, y) : x \geq 0, y \geq 0, Ax + By \leq b\}$ is nonempty and compact.

If the assumption (H_1) is satisfied, then the corresponding bilevel programming (1) is called convex quadratic bilevel programming (CQBP). In addition, if the assumption (H_2) is satisfied, then problem (1) has optimal solutions [18].

Following the above assumptions, we can reduce the BLP problem to the one-level programming problem [1]:

$$
\begin{aligned}
\min\quad & F(x,y) \\
\text{s.t.}\quad & Ax + By \leq b \\
& Qy + Dx + d_2 + B^T u - v = 0 \qquad (2)\\
& u^T(b - Ax - By) = 0 \\
& v^T y = 0 \\
& x, y, u, v \geq 0
\end{aligned}
$$

where $u \in R^q$, $v \in R^m$.

Problem (2) is non-convex and non-differentiable, moreover the regularity assumptions which are needed for successfully handling smooth optimization problems are never satisfied and it is not good for using the neural network approach to solve problem (2). But fortunately, Dempe [2] presents smoothing method for the BLP problem and the similar method is also presented in [19] for programs with complementary constraints. Following this smoothing method we can propose a neural network approach for problem (1).

Let $\varepsilon \in R_+$ be a parameter. Define the function $\Phi_\varepsilon : R^2 \to R$ by

$$\Phi_\varepsilon(a, b) = \sqrt{a^2 + b^2 + \varepsilon} - a - b.$$

The important property of this function can be stated in the following result.

Proposition 1. *For every $\varepsilon > 0$, we have*

$$\Phi_\varepsilon(a, b) = 0 \Leftrightarrow a > 0, b > 0, ab = \frac{\varepsilon}{2}.$$

The above proposition is obvious, and the proof is omitted. It is noted that $\Phi_\varepsilon(a, b)$ is smooth with respect to a, b for $\varepsilon > 0$.

Remark. The definition of the function $\Phi_\varepsilon(a, b)$ is inspired from the definition of perturbed *Fischer–Burmeister* function [2], which is used to smooth the programming problem with complementary constraints. In fact, if we add minus sign to perturbed *Fischer–Burmeister* function, then we can get the expression of the function $\Phi_\varepsilon(a, b)$. The reason why we adopt the function $\Phi_\varepsilon(a, b)$ to smooth problem (2) is that the smoothed problem can have convexity, and this can be shown in Proposition 3.

Before presenting the smoothed problem, we firstly give the following proposition.

Proposition 2. *For $x \in R^n$, we have*

$$x \geq 0 \Leftrightarrow \frac{1}{2}x^T(x - |x|) = 0.$$

Moreover, $\frac{1}{2}x^T(x - |x|)$ is continuous and convex.

Proof. Following the definition of continuity and convexity. Proposition 2 can be obtained immediately. □

Remark. The function of Proposition 2 is that we need not introduce slackness variables for the inequality constraints. Then, compared with the neural network proposed in [17], the structure of the neural network proposed in this paper can be simplified.

Using the function $\Phi_\varepsilon(a, b)$ and following Proposition 2, problem (2) can be approximated by

$$\begin{aligned} \min \quad & F(x, y) \\ \text{s.t.} \quad & Qy + Dx + d_2 + B^T u - v = 0 \\ & \sqrt{u_i^2 + (b - Ax - By)_i^2 + \varepsilon} - u_i - (b - Ax - By)_i = 0, \quad i = 1, \ldots, q \\ & \sqrt{v_j^2 + y_j^2 + \varepsilon} - v_j - y_j = 0, \quad j = 1, \ldots, m \\ & \frac{1}{2}x^T(x - |x|) = 0. \end{aligned} \tag{3}$$

Proposition 3. *If the assumption* (H_1) *is satisfied, then problem (3) is convex programming.*

Proof. Following the definition of convex programming [20] and through simple verifying calculation, Proposition 3 can be obtained immediately. □

Following problem (3), we overcome the difficulty that problem (2) dose not satisfy any regularity assumptions, which are needed for successfully handling smooth optimization problems, and pave the way for using neural network approach to solve problem (2). To simply our discussion, we introduce the following notations.

$$F'(x, y, u, v) = F(x, y), \qquad H(x, y, u, v) = \begin{pmatrix} Qy + Dx + d_2 + B^T u - v \\ \Phi_\varepsilon(u_i, (b - Ax - By)_i), i = 1, \ldots, q \\ \Phi_\varepsilon(y_j, v_j), j = 1, \ldots, m \\ \frac{1}{2}x^T(x - |x|) \end{pmatrix}.$$

Let $z^T = (x^T, y^T, u^T, v^T)$, then problem (3) can be written as:

$$\begin{aligned} \min \quad & F'(z) \\ \text{s.t.} \quad & H(z) = 0. \end{aligned} \tag{4}$$

Definition 1. Let z be a feasible point of problem (4). We say that z is a regular point if the gradients $\nabla H_1(z), \ldots, \nabla H_{2m+q+1}(z)$ are linearly independent.

Similar to the main result in [2] (Theorem 6.11), we can have the following result.

Theorem 1. *Let $\{z^\varepsilon\}$ be a sequence of solutions of problem (4). Suppose that the sequence $\{z^\varepsilon\}$ converges to some $\bar{z}$ for $\varepsilon \to 0+$. If $\bar{z}$ is a regular point, then $\bar{z}$ is a Bouligand stationary solution for the CQBP problem (1).*

3. Neural network for CQBP problem

3.1. Definition of the neural network

We can definite the following Lagrange function of problem (4).

$$L(z, \mu) = F'(z) + \sum_{k=1}^{2m+q+1} \mu_k H_k(z)$$

where $\mu \in R^{2m+q+1}$ is referred as the Lagrange multiplier.

Theorem 2. *If there exists* (z^*, μ^*)*, such that the following equalities are satisfied*

$$\nabla_z L(z^*, \mu^*) = 0, \qquad H(z^*) = 0.$$

Then, z^* *is an optimal solution of problem (4).*

Proof. From Proposition 3, we know firstly that problem (4) is convex programming. Following the sufficient optimality conditions of the convex programming, Theorem 2 can be obtained immediately. □

By Theorem 2, the energy function of problem (4) can be constructed as follows:

$$E(z, \mu) = \frac{1}{2}\|\nabla_z L(z, \mu)\|^2 + \frac{1}{2}\|H(z)\|^2.$$

Then we can use the gradient system to construct the following neural network for solving CQBP problem:

$$\begin{cases} \dfrac{\mathrm{d}z}{\mathrm{d}t} = -\nabla_z E(z, \mu) \\ \dfrac{\mathrm{d}\mu}{\mathrm{d}t} = -\nabla_\mu E(z, \mu). \end{cases} \tag{5}$$

That is

$$\frac{\mathrm{d}z}{\mathrm{d}t} = -\nabla^2_{zz} L(z, \mu) \cdot \nabla_z L(z, \mu) - H^T(z) \cdot \nabla_z H(z)$$
$$\frac{\mathrm{d}\mu}{\mathrm{d}t} = -\nabla^2_{z\mu} L(z, mu) \cdot \nabla_z L(z, \mu).$$

3.2. Stability analysis

Let $\eta^T = (z^T, \mu^T)$, we can have the following theorem.

Theorem 3. *Suppose that the neural network (5) has unique equilibrium point* η^**, then* η^* *is uniformly and asymptotically stable.*

Proof. Let $V(\eta) = E(\eta) - E(\eta^*) = E(\eta) \geq 0$, then $\forall \eta \neq \eta^*$, $V(\eta) > 0$. That is to say, $V(\eta)$ is a positive definite function. Moreover, given an arbitrary initial point η^0, there exists an unique trajectory $\eta = \eta(t, \eta^0)$ of the system in (5), along it there is

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} V(\eta) &= \frac{\mathrm{d}}{\mathrm{d}t} V[\eta(t, \eta^0)] \\ &= \nabla V(\eta)^T \cdot \frac{\mathrm{d}\eta}{\mathrm{d}t} \\ &= -\|\nabla E(\eta)\|^2 < 0. \end{aligned}$$

By the Lyapunov Theorem, η^* is uniformly and asymptotically stable. □

The following theorem shows that under some conditions, the trajectory does converge to the equilibrium of the neural network (5).

Theorem 4. *Suppose that the level set* $L(\eta^0) = \{\eta : E(\eta) \leq E(\eta^0)\}$ *is bound, then there exists an equilibrium point* $\bar{\eta}$ *and a strictly increasing sequence* $\{t_n\}$ $(t_n \geq 0)$*, such that*

$$\lim_{n\to+\infty} \eta(t_n, \eta^0) = \bar{\eta}, \qquad \nabla E(\bar{\eta}) = 0.$$

Proof. Firstly, we will prove the following two results.

(a) The energy function $E(\eta(t, \eta^0))$ is monotone non-increasing along the trajectory $\eta = \eta(t, \eta^0)$.

In fact, following Theorem 3 the result (a) is obvious.

(b) $\Omega = \{\eta(t, \eta^0) : t \geq 0\}$ is a bound positive semi-trajectory.

In fact, $E(\eta)$ is bound from below and a continuous function, thus by (a), the level set $L(\eta^0)$ is a bound closed set and

$$\Omega \subseteq L(\eta^0).$$

Hence, $\eta(t, \eta^0)$ $(t \geq 0)$ is a bound positive semi-trajectory.

Now, we will prove $\lim_{n\to+\infty} \eta(t_n, \eta^0) = \bar{\eta}$.

Table 1
The comparison of the optimal solution.

Examples in this paper	Optimal solution (x^*, y^*) corresponding to different ε			Optimal solution (x^*, y^*) from the references
	$\varepsilon = 0.01$	$\varepsilon = 0.001$	$\varepsilon = 0.0001$	
Example 1	(4.98, 2.01)	(5.0, 2.0)	(5.0, 2.0)	(5.0, 2.0)
Example 2	(0.62, 0.39, 0, 0, 1.83)	(0.61, 0.39, 0, 0, 1.83)	(0.61, 0.39, 0, 0, 1.83)	(0.609, 0.391, 0, 0, 1.828)
Example 3	(1.01, 0.01, 1.0)	(1.0, 0, 1.0)	(1.0, 0, 1.0)	(1.0, 0, 1.0)

Firstly, Ω is a bound set of points. Take strictly increasing sequence $\bar{t}_n$, $0 \le \bar{t}_1 < \bar{t}_2 < \cdots < \bar{t}_n \to +\infty$, then $\eta(\bar{t}_n, \eta^0)$ is a bound sequence composed of infinitely points. Then there exists a limiting point $\bar{\eta}$, that is to say, there exists a strictly increasing subsequence $\{t_n\} \subseteq \{\bar{t}_n\}$, $t_n \to +\infty$ such that

$$\lim_{n\to+\infty} \eta(t_n, \eta^0) = \bar{\eta}.$$

Finally, we prove that $\nabla E(\bar{\eta}) = 0$.

By (a), $E(\eta)$ is a Lyapunov function. $E(\bar{\eta}) = 0 \Leftrightarrow \nabla E(\bar{\eta}) = 0$. By LaSalle invariance principle [21], $\eta(t, \eta^0) \to S'$ $(t \to +\infty)$ (S' is the largest invariant set in the set of equilibrium points), that is, there exists a sequence $\{t_n\}$ $(t_n \ge 0)$ such that

$$\lim_{n\to+\infty} \eta(t_n, \eta^0) = \bar{\eta}, \qquad \nabla E(\bar{\eta}) = 0.$$

The proof is completed. □

The following theorem shows that the neural network (5) is globally and asymptotically stable.

Theorem 5. *Suppose that the neural network (5) has unique equilibrium point and the level set $L(\eta^0) = \{\eta : E(\eta) \le E(\eta^0)\}$ is bounded. Then every trajectory of network (5) converges to the optimal solution η^* of problem (4), i.e. the optimal solution η^* is globally and asymptotically stable.*

Proof. Following Theorems 3 and 4, this theorem is obvious. □

4. Numerical experiments

Now, we are in the position to give some examples to illustrate the neural network approach for the CQBP problem.

Example 1 (*[1]*). Consider the following nonlinear BLP problem, where $x \in R^1$, $y \in R^1$.

$$\begin{aligned}
&\min_{x\ge 0} F(x, y) = (x-5)^2 + (2y+1)^2\\
&\text{s.t. } \min_{y\ge 0} f(x, y) = (y-1)^2 - 1.5xy\\
&\text{s.t. } -3x + y + 3 \le 0\\
&x - 0.5y - 4 \le 0\\
&x + y - 7 \le 0.
\end{aligned}$$

After applying the Kuhn–Tucker transformation and the smoothing method, the above problem reduces to a problem similar to problem (3). Then similar to problem (5), we can get a set of ordinary differential equations, which describes the transient behavior of the neural network, and adopt the classical fourth order Runge–Kutta method to solve these ordinary differential equations.

We make program with Microsoft Visual C++ 6.0 and use a personal computer (CPU: Intel Pentium 2.1 GHz, RAM: 512 MB) to execute the program. Following Theorem 1, we let ε have different small values and Table 1 presents the different optimal solutions of Example 1 over the different ε. The initial condition is $(x, y, u_1, u_2, u_3, v) = (1, 1, 1, 1, 1, 1)$ and $(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6) = (1, 1, 1, 1, 1, 1)$. Fig. 1 shows the transient behavior of the variables in Example 1 ($\varepsilon = 0.001$). In addition, we take randomly other three different initial points $\gamma_1 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, $\gamma_2 = (1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0)$, $\gamma_3 = (1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1)$. The simulation calculations show that every trajectory, which is corresponding to the different initial point $\gamma_1, \gamma_2, \gamma_3$, of the neural network does converge to the optimal solution $(x^*, y^*) = (5.0, 2.0)$. The numerical results accord with the result proven in Theorem 5.

Noted that for the convex quadratic BLP problem, Muu [22] presented a global optimization approach based on branch-and-bound method. All the test problems in [22], i.e. the following Exams. 2 and 3, are solved using the neural network approach proposed in this paper, the results are listed in Table 1. Compared with the method in [22], the neural network

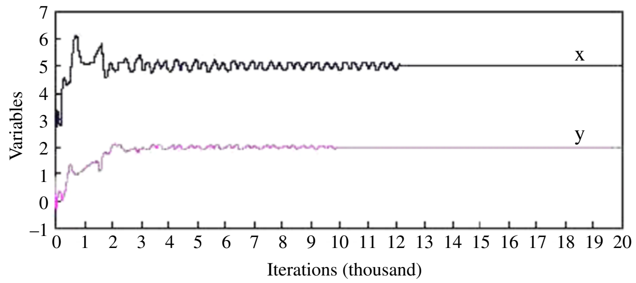

Fig. 1. The transient behavior of the variables in Example 1.

approach has the following features:

(i) The construction and realization of the neural network is simple. In fact, we can get the neural network of the CQBP problem just using the Lagrange function of the smoothed problem, and the neural network can be realized by the classical fourth order Runge–Kutta method.

(ii) The neural network approach has rapid convergence capability, and this feature can be verified by the fact that all the test problems in this paper can be solved in several seconds.

Example 2 ([22]).

$$\begin{aligned}
&\min_x -7x_1 + 4x_2 + y_1^2 + y_3^2 - y_1y_3 - 4y_2\\
&\text{s.t. } x_1 + x_2 \le 1\\
&x_1, x_2 \ge 0\\
&\min_y \ (1-3x_1)y_1 + (1+x_2)y_2 + y_1^2 + \frac{1}{2}y_2^2 + \frac{1}{2}y_3^2 + y_1y_2\\
&\text{s.t. } x_1 - 2x_2 + 2y_1 + y_2 - y_3 + 2 \le 0\\
&y_1, y_2, y_3 \ge 0.
\end{aligned}$$

Example 3 ([22]).

$$\begin{aligned}
&\min_x \ x^2 - 4x + y_1^2 + y_2^2\\
&\text{s.t. } \ 0 \le x \le 2\\
&\min_y y_1^2 + 0.5y_2^2 + y_1y_2 + (1-3x)y_1 + (1+x)y_2\\
&\text{s.t. } 2y_1 + y_2 - 2x \le 1\\
&y_1, y_2 \ge 0.
\end{aligned}$$

5. Conclusion

In this paper we present a novel neural network for the CQBP problem, and the numerical results show that the computed results converge to the optimal solution with the decreasing of the parameter ε, which corresponds to the result in Theorem 1. In fact, a mass of additional numerical experiments show that we can get a satisfying approximate solution of the CQBP problem when we take $\varepsilon = 0.01$.

The distinguishing feature of the proposed neural network is that for an arbitrary initial point, the trajectory of the neural network does converge to the equilibrium point, which is also an optimal solution of the CQBP problem. Then the neural network proposed in this paper has better calculation perspective.

References

[1] J. Bard, Practical Bilevel Optimization: Algorithm and Applications, Kluwer Academic Publishers, Dordrecht, 1998.
[2] S. Dempe, Foundation of Bilevel Programming, Kluwer Academic Publishers, London, 2002.
[3] O. Ben-Ayed, O. Blair, Computational difficulity of bilevel linear programming, Operations Research 38 (1990) 556–560.
[4] W.F. Bialas, M.H. Karwan, On two-level optimization, IEEE Transaction on Automatic Control 27 (1) (1982) 211–214.
[5] E.S. Lee, H.S. Shih, Fuzzy and Multi-Level Decision Making: An Interactive Computational Approach, Springer-Verlag, London, 2001.
[6] Y. Lv, T. Hu, Z. Wan, A penalty functin method for solving weak price control problem, Applied Mathematics and Computation 186 (2) (2007) 1520–1525.
[7] K.Z. Chen, Y. Leung, K.S. Leung, et al., A neural network for solving nonlinear programming problems, Neural Computing and Applications 11 (2002) 103–111.

[8] X.B. Gao, A novel neural network for nonlinear convex programming, IEEE Transactions on Neural Networks 15 (3) (2004) 613–621.
[9] T. Hu, Y. Guo, Neural network for multiobjective dynamic programming, Acta Electronica Sinica 27 (10) (1990) 70–72.
[10] M.P. Kennedy, L.O. Chua, Neural network for nonlinear programming, IEEE Transaction on Circuits and Systems 35 (1998) 554–562.
[11] D.W. Tank, J.J. Hopfield, Simple 'neural' optimization networks: An A/D convert, siginal decision circuit, and a linear programming circuit, IEEE Transactions on Circuits and Systems 33 (1986) 533–541.
[12] Y.S. Xia, J. Wang, A general methodology for designing globally convergent optimizatin neural networks, IEEE Transactions on Neural Networks 9 (6) (1998) 1331–1343.
[13] S. Zhang, A.G. Constantinides, Lagrange programming neural networks, IEEE Transactions on Circuits and Systems 39 (7) (1992) 441–452.
[14] Z. Sheng, Z. Lv, R. Xu, A new algorithm based on the frank-wolfe method and neural network for a class of bilevel decision making problems, Acta Automatica Sinica 22 (6) (1996) 657–665.
[15] H.S. Shih, U.P. Wen, E.S. Lee, et al., A neural network approach to multi-objective and multilevel programming problems, Computers and Mathematics with Applications 48 (2004) 95–108.
[16] K.M. Lan, U.P. Wen, et al., A hybrid neural network approach to bilevel programming problems, Applied Mathematics Letters 20 (8) (2007) 880–884.
[17] Y. Lv, T. Hu, G. Wang, Z. Wan, A neural network approach for solving nonlinear bilevel programming problem, Computer and Mathematics with Applications 58 (12) (2008) 2823–2829.
[18] Chenggen Shi, Guangquan Zhang, Jie Lu, On the definition of linear bilevel programming solutin, Applied Mathematics and Computation 160 (2005) 169–173.
[19] F. Facchinei, H. Jiang, L. Qi, A smoothing method for mathematical programming with equilibrium constraints, Mathematical Programming 85 (1999) 107–134.
[20] J.P. Brans, Operational Research, North Holland Company, 1984.
[21] J.P LaSalle, The Stability of Dynamical Systems, Springer, New York, 1976.
[22] L.D. Muu, N.V. Quy, A global optimization method for solving convex quadratic bilevel programming problems, Journal of Global Optimization 26 (2003) 199–219.