http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 11 (6): 713-718, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.713.718

Applications of the General Nonlinear Neural Networks in Solving the Inverse Optimal Value Problem with Linear Constraints

[1]Huaiqin Wu, [1]Kewang Wang, [1]Ning Li, [2]Chongyang Wu, [1]Qiangqiang Guo and [1]Guohua Xu
[1]Department of Applied Mathematics, Yanshan University, Qinhuangdao 066004, China
[2]College of Economics and Finance, Huaqiao University, Quanzhou 362021, China

Abstract: In this research, the aim of the study is to present a method to solve a class of inverse optimal value problem with linear constraints by using a nonlinear gradient neural network. Firstly, based on optimal theory, solving the inverse optimal value problem is changed as solving a nonlinear bilevel program problem equivalently. Then a nonlinear gradient neural network model for solving the nonlinear bilevel programming is presented. By employing Lyapunov function approach, the proposed neural network is analyzed to be globally Lyapunov stable and capable of generating approximal optimal solution to the nonlinear bilevel programming problem. Finally, numerical examples are provided to verify the feasibility and the efficiency of the proposed method in this study.

Key words: Neural networks, bilevel programming, inverse optimal value problem, global asymptotical stability, fish-burmeister function

INTRODUCTION

In the recent years, the inverse optimal value problems have received extensive attention from a lot of scholars (Tarantola, 1987; Burton and Toint, 1994; Zhang *et al*., 1995, 1996; Sokkalingam *et al*., 1999; Zhang and Liu, 1996, 1999; Yang *et al*., 1997; Zhang and Cai, 1998; Ahuja and Orlin, 2000, 2001, 2002; Heuberger, 2004; Ahmed and Guan, 2005; Lv *et al*., 2008a; 2010a). Let min p (x, c), x∈X, be an optimization problem where, X is the feasible region and c is an parameter vector representing costs, capacities, weights, returns, etc. The general optimization problem (also called forward optimization) is to find an x*∈X such that the objective p (x, c) is optimal at x*. The inverse optimization problem can be described as follows. Assume that it has been known that an $\hat{x}$ and a vector c which is an estimate to the parameter values of the problem. We are interested in finding out whether there exist acceptable values of the parameter vector c, denoted by $\hat{c}$ which make $\hat{x}$ optimal to problem min $p(x,\hat{c})$. If the answer is positive, find the vector $\hat{c}$ that differs from the vector c as little as possible.

Burton and Toint (1994) first investigated an inverse shortest paths problem. Since, then, many different inverse optimization problems (discrete or continuous) have been considered. Zhang *et al*. (1996) addressed inverse shortest paths problems with l_1 norm for p paths using a column generation method. Zhang *et al*. (1996) and Sokkalingam *et al*. (1999) worked on inverse minimum spanning tree problems.

Zhang and Liu (1996, 1999) studied inverse linear programming problems. Worked on inverse maximum flow and minimum cut problems done by Yang *et al*. (1997) Zhang and Cai (1998). Ahuja and Orlin (2000, 2001, 2002) also developed a general approach to inverse linear programming problems and discussed applications to some special cases. Heuberger (2004) gave a survey on inverse combinatorial optimization problems, methods and results. Ahmed and Guan (2005) proved that the inverse optimal value problem is NP-hard and on the basis of some assumptions, got the optimal solutions of the inverse optimal value problem by solving a series of linear and bilinear programming problems. By using the penalty method, Lv *et al*. (2008b, 2010a) gave an existence theorem of solutions of the inverse optimal value problem and propose some numerical algorithms.

It should be noted that in the existing literature, almost all results is to focus mainly on the study of the classical numerical algorithm for the inverse optimal value problem. However, in lots of engineering applications many optimization problems need to be solved in real time. It is obvious that the classical methods cannot render real-time solutions to these optimization problems, especially large-scale problems. Compared with classical optimization approaches, the appearance of neural computing approach satisfies the demand of real-time optimal solutions. The main advantage of neural network approach to optimization is that the nature of the dynamic solution procedure is inherently parallel and distributed. Therefore, the neural network approach can solve optimization problems in running time at the orders

Corresponding Author: Huaiqin Wu, Department of Applied Mathematics, Yanshan University, Qinhuangdao 066004, China

of magnitude much faster than the most popular optimization algorithms executed on general purpose digital computers. Recently, there have been some works to focus on solving varied optimization problems in engineering by using neural networks, the reader may consult (Lv *et al*., 2010b; Gao, 2004; Xia and Wang, 1998; Chen *et al*., 2002; Wu *et al*., 2010; Al-Bastaki, 2006; Dempe, 2002; Lijia *et al*., 2011; Yedjour *et al*., 2011; Salazar *et al*., 2010; Arafa *et al*., 2011; Niknafs and Parsa, 2011; Lotfi and Benyettou, 2011; Liu *et al*., 2012) and the references therein. To the best of the authors' knowledge, solving the inverse optimal value problems by neural networks has not been investigated to date which inspirits us to carry out the present study.

Motivated by the above discussion, in the present paper, a new nonlinear neural networks for solving the inverse optimal value problem with linear constraints is presented. Firstly, the inverse value problem with linear constraints is transformed into the corresponding nonlinear bilevel programming problem; Secondly, a nonlinear neural network model for solving the bilevel programming is developed. The equilibrium point of the proposed neural network is equivalent to the solution of the inverse optimal value problem. Thirdly, a new sufficient condition to ensure the global stability for the proposed neural network is given, by the use of the Lyapunov function method. Finally, numerical examples are provided to show the effectiveness of the proposed approach.

PROBLEM STATEMENT AND PRELIMINARIES

Consider the following linear program problem:

$$\begin{aligned} &\min_x \; c^T x \\ &\text{subject to} \;\; Ax \le b \\ &\qquad\qquad\qquad x \ge 0 \end{aligned} \tag{1}$$

where, c, x$\in R^n$, A$\in R^{p\times n}$, b$\in R^p$. Given a set $C\subseteq R^n$ and a real number z*. Arguing as in Tarantola (1987), the inverse optimal value problem corresponding to Eq. 1 is to find $c\in C$ such that the optimal objective value Q(c) of the linear programming Eq. 1 is close to z*. It is obvious that the inverse optimal value problem can be written as:

$$\begin{aligned} &\min_c \; f(c) \\ &\text{subject to} \;\; c \in C \end{aligned} \tag{2}$$

where, f(c) = $|Q(c)-z^*|$, $|\cdot|$ denotes the Euclidean norm of R.

Combining Eq. 1 and 2, the inverse optimal value problem corresponding the linear program can be described as follows:

$$\begin{aligned} &(UP) \min_c \; f(c) \\ &\qquad \text{subject to} \;\; c \in C, \\ &(LP) \min_x \; c^T x \\ &\qquad \text{subject to} \;\; Ax \le b \\ &\qquad\qquad\qquad\qquad x \ge 0. \end{aligned} \tag{3}$$

The model Eq. 3 is a class of the bilever program problem with the optimal value of the lower level problem feeding back to the upper level. (UP) is called the upper level problem and (LP) is called the lower level problem.

Throughout, the rest of this study, the following assumptions are made:

- **H1:** The set of cost vectors C is nonempty, C = {c: Dc = d}, $D\subseteq R^{m\times n}$, $d\in R^m$
- **H2:** The feasible region {x: Ax$\le$b, x$\ge$0} is nonempty bound

Without losing generality, replacing f(c) = $|Q(c)-z^*|$ with $f^2(c) = (Q(c)-z^*)^2$, then the bilever program problem Eq. 3 can be rewritten as:

$$\begin{aligned} &(UP) \min_c \; (Q(c)-z^*)^2 \\ &\qquad \text{subject to} \;\; Dc = d, \\ &(LP) \min_x \; c^T x \\ &\qquad \text{subject to} \;\; Ax \le b \\ &\qquad\qquad\qquad\qquad x \ge 0. \end{aligned} \tag{4}$$

For fixed $c\in C$, by using the Kuhn-Tucker optimality conditions to replace (LP), the problem Eq. 4 can be changed equivalently as the following program problem:

$$\begin{aligned} &\min_{(c,u,v,x)} \; (c^T x - z^*)^2 \\ &\text{subject to} \;\; Dc = d \\ &\qquad\qquad Ax \le b \\ &\qquad\qquad uA - v = -c^T \\ &\qquad\qquad u(b - Ax) + vx = 0 \\ &\qquad\qquad u, v, x \ge 0. \end{aligned} \tag{5}$$

where, $u\in R^p$, $v\in R^n$ are row vectors. It is obvious that the problem (Eq. 5) is a convex program problem.

From the above discussion, it is easy to see that the inverse optimal value problem corresponding to the linear program (Eq. 1) is equivalent to a special class of the convex program (Eq. 5). Hence, solving the inverse optimal value problem corresponding to the linear program (Eq. 1) can be changed as solving the convex program (Eq. 5).

In order to obtain the main results of this study, the following definitions and lemmas are introduced.

Definition 1: The Fish-Burmeiste function is Φ: $R^2 \to R$ defined by $\Phi(a,b) = a + b - \sqrt{a^2 + b^2}$ and the perturbed Fish-Burmeister function is Φ: $R^3 \to R$ defined by $\Phi(a,b,\varepsilon) = a + b - \sqrt{a^2 + b^2 + \varepsilon}$.

The Fish-Burmeister function $\Phi(a,b)$ has the property: $\Phi(a,b) = \Leftrightarrow a \geq 0,\ b \geq 0,\ ab = 0$. Its perturbed variant satisfies $\Phi(a,b,\varepsilon)$ if and only if:

$$a \geq 0,\ b \geq 0,\ ab = \frac{\varepsilon}{2}$$

for $\varepsilon \geq 0$. It is easy to see that $\Phi(a,b)$ is non-differentiable at $a = b = 0$. However, $\Phi(a, b, \varepsilon)$ is smooth with respect to a, b for $\varepsilon > 0$.

Let $\Phi'(a,b,\varepsilon) = \sqrt{a^2 + b^2 + \varepsilon} - a - b$. It is obvious that $\Phi'(a, b, \varepsilon)$ has the same property with the function $\Phi(a, b, \varepsilon)$. By using $\Phi(a, b, \varepsilon)$, the problem (Eq. 5) can be approximated by:

$$\begin{aligned}
&\min_{(c,u,v,x)} (c^T x - z^*)^2 \\
&\text{subject to}\quad uA - v + c^T = 0 \\
&\qquad\qquad \sqrt{{h_i}^2 + {u_i}^2 + \varepsilon} - h_i - u_i = 0, i = 1, \cdots, p \\
&\qquad\qquad \sqrt{{v_j}^2 + {x_j}^2 + \varepsilon} - v_j - x_j = 0, j = 1, \cdots, n \\
&\qquad\qquad Dc = d.
\end{aligned} \tag{6}$$

where, $h(x) = (h_1(x),\ldots, h_p(x))^T = b - Ax$.

To simply the discussion, set:

$$\begin{aligned}
&F(c,u,v,x) = (c^T x - z^*)^2, \\
&H(c,u,v,x) = \begin{pmatrix} uA - v + c^T \\ \Phi'(h(x)_i, u_i, \varepsilon), i = 1, \cdots, p \\ \Phi'(x_j, v_j, \varepsilon), j = 1, \cdots, n \\ Dc - d \end{pmatrix}, \\
&y^T = (c^T, u, v, x^T)
\end{aligned}$$

then the program problem (Eq. 6) can be written as:

$$\begin{aligned}
&\min_y F(y) \\
&\text{subject to}\quad H_l(y) = 0, l = 1, \cdots, p + 2n + m
\end{aligned} \tag{7}$$

Definition 2: Let $\{y^\varepsilon\}$ be a feasible point of the program problem (Eq. 7). y is said to a regular point if the gradients $\nabla H_1(y),\ldots, \nabla H_{p+2n+m}(y)$ are linearly independent.

By using theorem in Dempe (2002) the following lemma can be obtained.

Lemma 1: Let $\{y^\varepsilon\}$ be a sequence of solution of the program problem (Eq. 7). Suppose that the sequence $\{y^\varepsilon\}$ converges to some $\tilde{y}$ for $\varepsilon \to 0^+$. If $\tilde{y}$ is a regular point, then $\tilde{y}$ is a Bouligand stationary solution for problem (Eq. 4).

NEURAL NETWORK MODEL FOR THE PROGRAM PROBLEM

Neural network model: Define the Lagrange function of the program problem (Eq. 7) as $L(y, \lambda) = F(y) + \lambda H(y)$, where, $\lambda \in R^{p+2n+m}$ is referred as the Lagrange multiplier. Based on the well-known Saddle point theorem, we have.

Theorem 1: If there exists (y^*, λ^*), such that $\Delta_y L(y^*,\lambda) = 0$, $H(y^*)$ hold. Then, y^* is an optimal solution of the program problem (Eq. 7).

By Theorem 1, the energy function of the program problem (Eq. 7) can be constructed as:

$$E(y,\lambda) = \frac{1}{2}\left\|\nabla_y L(y,\lambda)\right\|^2 + \frac{1}{2}\left\|\nabla_\lambda L(y,\lambda)\right\|^2$$

in the following, a neural network model is proposed which is said to be the gradient network, for solving the problem (7), whose dynamical equation is described as follows:

$$\frac{dy}{dt} = -\nabla_y E(y,\lambda), \frac{d\lambda}{dt} = -\nabla_\lambda E(y,\lambda) \tag{8}$$

That is:

$$\begin{aligned}
\frac{dy}{dt} &= -\nabla^2_{yy} L(y,\lambda) \cdot \nabla_y L(y,\lambda) - H^T(y) \nabla_y H(y), \\
\frac{d\lambda}{dt} &= -\nabla^2_{y\lambda} L(y,\lambda) \cdot \nabla_y L(y,\lambda).
\end{aligned}$$

According to Theorem 1, it is easy to obtain.

Theorem 2: (y^*,λ^*) is an equilibrium point of the neural network (Eq. 8) if and only if y^* is an optimal solution of the program problem (Eq. 7).

Proof: Firstly, if (y^*,λ^*) is an equilibrium point of the neural network (Eq. 8), that means:

$$\begin{gathered}
E(y^*,\lambda^*) = \frac{1}{2}\left\|\nabla_y L(y^*,\lambda^*)\right\|^2 + \frac{1}{2}\left\|\nabla_\lambda L(y^*,\lambda^*)\right\|^2 = 0, \\
\text{i.e., } \nabla_y L(y^*,\lambda^*) = 0 \text{ and } \nabla_\lambda L(y^*,\lambda^*) = 0. \\
\text{That is } \nabla_y L(y^*,\lambda^*) = 0 \text{ and } H(y^*) = 0
\end{gathered}$$

By theorem 1, we have that y^* is an optimal solution of program problem (Eq. 7).

Conversely, if y^* is an optimal solution of program problem (Eq. 7), from K-T condition we have that there exist λ^*, such that $\nabla y_L(y^*,\lambda^*) = 0$ and $\nabla y_\lambda(y^*,\lambda^*) = 0$. i.e., $\nabla y_L(y^*,\lambda^*) = 0$ and $H(y^*) = 0$. Then, the equations $-\nabla^2_{yy}L(y^*,\lambda).\nabla_y L(y^*,\lambda^*)-H^T(y)\nabla_y H(y^*) = 0$ and $-\nabla^2_{yy}L(y^*,\lambda).-\nabla_y L(y^*,\lambda) = 0$ are existence. This means (y^*,λ^*) is an equilibrium point of the neural network (Eq. 8).

Stability analysis: Since, the program problem (Eq. 7) is a class of convex programs, there exists an unique optimal solution. Hence, by Theorem 2, the neural network (Eq. 8) has an equilibrium point.

Theorem 3: If (y^*,λ^*) is the equilibrium point of the neural network (Eq. 8), then (y^*,λ^*) is globally asymptotically stable.

Proof: Let $\eta = (y,\lambda)$. Consider Lyapunov function:

$$V(\eta) = E(\eta)-E(\eta^*) = E(\eta)$$

It is obvious that $V(\eta)>0$, $\forall\eta\neq\eta^*$ and $V(\eta^*) = 0$. Calculate the derivative of $V(\eta)$ along the solution $\eta(t)$ of the neural network (Eq. 8), we have:

$$\frac{d}{dx}V(\eta) = \nabla V(\eta)^T \cdot \frac{d\eta}{dt} = -\|\nabla E(\eta)\|^2 < 0,\ \eta \neq \eta^*$$

Based on the Lyapunov stability theory, we can obtain that η^* is globally asymptotically stable.

ILLUSTRATIVE EXAMPLES

Here, three examples will be given to illustrate the effectiveness of the proposed approach for solving the inverse optimal value problem.

Example 1: Consider the following inverse optimal value problem:

$$\begin{aligned} &(\text{UP}) \min_{c_1,c_2} (Q(c_1,c_2)-z^*)^2 \\ &\quad \text{subject to } c_1 + 2c_2 = 8, \\ &(\text{LP}) \max_{x_1,x_2} \sum_{i=1}^{2} c_i x_i \\ &\quad \text{subject to } x_1 + 2x_2 \le 8 \\ &\quad\quad 0 \le x_1 \le 4 \\ &\quad\quad 0 \le x_2 \le 3. \end{aligned} \tag{9}$$

Let $z^* = 14$. The problem (Eq. 6) corresponding to (Eq. 9) is:

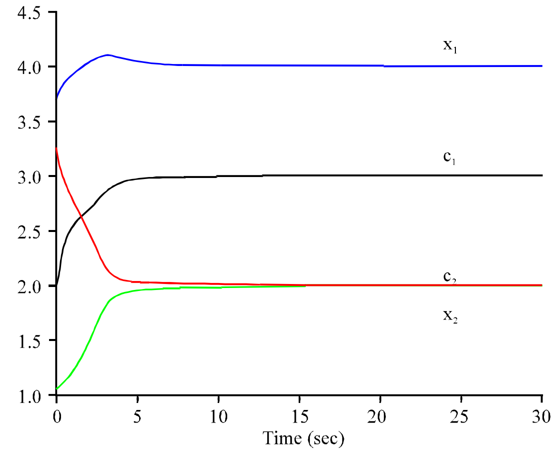

Fig. 1: The transient behavior of the variables in Example 1

$$\begin{aligned} &\min_{(c,u,v,x)} (c_1x_1 + c_2x_2 - 14)^2 \\ &\text{subject to } u_1 + u_2 - v_1 - c_1 = 0 \\ &\quad 2u_1 + u_3 - v_2 - c_2 = 0 \\ &\quad \sqrt{(8-x_1-2x_2)^2 + u_1^2 + \varepsilon} - 8 + x_1 + 2x_2 - u_1 = 0 \\ &\quad \sqrt{(4-x_1)^2 + u_2^2 + \varepsilon} - 4 + x_1 - u_2 = 0 \\ &\quad \sqrt{(3-x_2)^2 + u_3^2 + \varepsilon} - 3 + x_2 - u_3 = 0 \\ &\quad \sqrt{v_1^2 + x_1^2 + \varepsilon} - v_1 - x_1 = 0 \\ &\quad \sqrt{v_2^2 + x_2^2 + \varepsilon} - v_2 - x_2 = 0 \\ &\quad c_1 + 2c_2 - 8 = 0. \end{aligned} \tag{10}$$

By using the classical fourth-order Runge-Kutta method in MATLAB to solving the neural network (Eq. 8), we can obtain the transient behavior of the state trajectories (x_1,x_2,c_1,c_2) (Fig. 1). Take the initial values $y = (3.5,1,3,2,0,0,0,0,0)$ and $\lambda = (0,0,0,0,0,0,0,0)$. Let ε be 0.01, 0.001 and 0.00001, respectively, we can get the different optimal solutions of the problem (10) (Table 1). It is easy to see that optimal solution $(x^*_1,x^*_2,c^*_1,c^*_2)$ of the problem (10) converges to the solution (4,2,2,3) of the inverse optimal value problem (9).

Example 2: Consider the following inverse optimal value problem:

$$\begin{aligned} &(\text{UP}) \min_{c_1,c_2} (Q(c_1,c_2)-5)^2 \\ &\quad \text{subject to } c_1 + c_2 = 3, \\ &(\text{LP}) \max_{x_1,x_2} \sum_{i=1}^{2} c_i x_i \\ &\quad \text{subject to } 4x_1 + 3x_2 \le 12 \\ &\quad\quad 4x_1 + x_2 \le 8 \\ &\quad\quad x_1, x_2 \ge 0. \end{aligned} \tag{11}$$

Table 1: The comparison of the optimal solution of example 1 and 2

	$\varepsilon = 0.01$	$\varepsilon = 0.001$	$\varepsilon = 0.0001$
e1	(3.9902,2.0032,2.0034,2.9983)	(3.9990,2.0003,2.0004,2.9998)	(3.9999,2.0000,2.0001,2.9999)
e2	(1.4943,2.0013,1.9798,1.0201)	(1.4995,2.0000,1.9978,1.0022)	(1.4999,2.0000,1.9996,1.0004)

Different optimal solution (x^*_1, x^*_2, c^*_1, c^*_2) corresponding to different ε

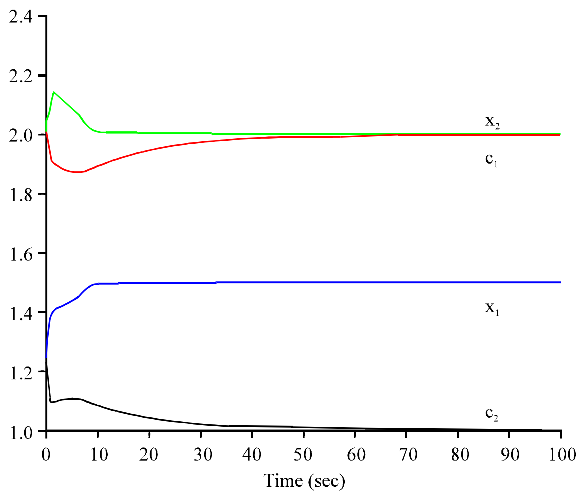

Fig. 2: The transient behavior of the variables in Example 2

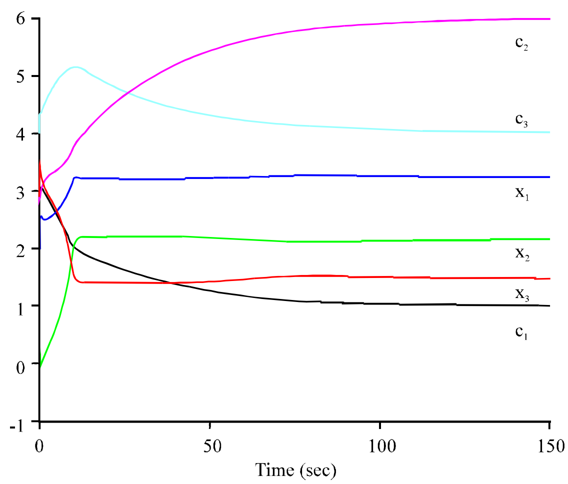

Fig. 3: The transient behavior of the variables in Example 3

Using the same method of example 1, we can obtain the transient behavior of the state trajectories (x_1,x_2,c_1,c_2) (Fig. 2). Take the initial values y = (1,2,2,1,0,0,0,0) and λ = (0,0,0,0,0,0,0,0). Let ε be 0.01, 0.001 and 0.0001, respectively, we can get the different optimal solutions of the problem (11) (Table 1). It is easy to see that optimal solution (x^*_1,x^*_2,c^*_1, c^*_2) of the problem (11) converges to the solution (1.5,2,2,1).

Example 3: Consider the following inverse optimal value problem:

Table 2: The comparison of the optimal solution of example 3

$\varepsilon = 0.01$	(3.2431,2.1355,1.4836,1.0128,5.9805,4.0066)
$\varepsilon = 0.001$	(3.2382,2.1428,1.4761,1.0074,5.9818,4.0107)
$\varepsilon = 0.0001$	(3.2376,2.1438,1.4750,1.0069,5.9819,4.0111)

Different optimal solution (x^*_1, x^*_2, x^*_3, c^*_1, c^*_2, c^*_3) corresponding to different ε

$$
\begin{aligned}
&(\text{UP}) \min_{c_1,c_2,c_3} \ (Q(c_1,c_2,c_3)-22)^2 \\
&\qquad \text{subject to} \ \ c_1+c_2+c_3=11, \\
&(\text{LP}) \max_{x_1,x_2,x_3} \ \sum_{i=1}^{3} c_i x_i \\
&\qquad \text{subject to} \ \ -x_1+2x_2+2x_3 \le 4 \\
&\qquad\qquad 4x_1-4x_2+x_3 \le 21 \\
&\qquad\qquad x_1+2x_2+x_3 \le 9 \\
&\qquad\qquad x_1,x_2,x_3 \ge 0.
\end{aligned} \tag{12}
$$

Take the initial values y = (2,0,3,3,3,4,0,0,0,0,0,0) and λ = (0,0,0,0,0,0,0,0,0,0). Let ε be 0.01, 0.001 and 0.001, respectively, we can get the different optimal solutions of the problem (12) (Table 2). The transient behavior of the state trajectories (x_1,x_2,x_3,c_1,c_2,c_3) (Fig. 3).

Table 2 the comparison of the optimal solution of example 3.

From the above examples we can find that the computed results converge to the different optimal solution with the decreasing of ε. It shows that the neural network approach is feasible to the inverse optimal value problem.

CONCLUSION

In this study, a neural network model has been developed for solving the inverse optimal value problem with linear constraints. Based on Lyapunov stability theory, the proposed neural network has been proved to be globally asymptotically stable and capable of generating approximal optimal solution of the inverse optimal value problem. Three examples have been given to show the effectiveness of the proposed approach. The results obtained in this paper are highly valuable in both theory and practice for solving the inverse optimal value problem.

ACKNOWLEDGMENT

The authors are extremely grateful to the Editor and the Reviewers for their valuable comments and suggestions which help to enrich the content and improve the presentationof this study. This study was supported by the Natural Science Foundation of Hebei Province of China (A2011203103) and the Hebei Province Education Foundation of China (2009157).

REFERENCES

Ahmed, S. and Y. Guan, 2005. The inverse optimal value problem. Math. Program., 102: 91-110.

Ahuja, R.K. and J.B. Orlin, 2000. A faster algorithm for the inverse spanning tree problem. J. Algorithms, 34: 177-193.

Ahuja, R. and J. Orlin, 2001. Inverse optimization. Operat. Res., 49: 771-783.

Ahuja, R.K. and J.B. Orlin, 2002. Combinatorial algorithms for inverse network flow problems. Networks, 40: 181-187.

Al-Bastaki, Y.A.L., 2006. GIS image compression and restoration: A neural network approach. Inform. Technol. J., 5: 88-93.

Arafa, M., M. Alqedra and H. An-Najjar, 2011. Neural network models for predicting shear strength of reinforced normal and high-strength concrete deep beams. J. Applied Sci., 11: 266-274.

Burton, D. and P.L. Toint, 1994. On the use of an inverse shortest paths algorithm for recovering correlated costs. Math. Program., 63: 1-22.

Chen, K.Z., Y. Leung, K.S. Leung and X.B. Gao, 2002. A neural network for solving nonlinear programming problems. Neural Comput. Applic., 11: 103-111.

Dempe, S., 2002. Foundation of Bilevel Programming. Kluwer Academic Publishers, London, New York, Moscow, ISBN:1-402-0631-4, Pages: 306.

Gao, X.B., 2004. A novel neural network for nonlinear convex programming. IEEE Trans. Neural Networks, 15: 613-621.

Heuberger, C., 2004. Inverse combinatorial optimization: A survey on problems, methods and results. J. Comb. Optimizat., 8: 329-361.

Lijia, C., Z. Shengxiu, L. Xiaofeng, L. Yinan and L. Ying, 2011. Nonlinear adaptive block backstepping control using command filter and neural networks approximation. Infor. Technol. J., 10: 2284-2291.

Liu, G., S.X. Yang and Y. Chai, 2012. Less conservative delay-dependent asymptotic stability criteria for stochastic neural networks of neutral-type with distributed delay. Inform. Technol. J., 11: 233-240.

Lotfi, A. and A. Benyettou, 2011. Using probabilistic neural networks for handwritten digit recognition. J. Artif. Intell., 4: 288-294.

Lv, Y., T. Hu and Z. Wan, 2008a. A penalty function method for solving inverse optimal value problem. J. Comput. Applied Math., 220: 175-180.

Lv, Y., T. Hu, G. Wang and Z. Wan, 2008b. A neural network approach for solving nonlinear bilevel programming problem. Comput Math. Applic., 55: 2823-2829.

Lv, Y., Z. Chen and Z. Wan, 2010a. A penalty function method based on bilevel programming for solving inverse optimal value problems. Applied Math. Lett., 23: 170-175.

Lv, Y., Z. Chen and Z. Wan, 2010b. A neural network for solving a convex quadratic bilevel programming problem. J. Comput. Applied Mathe., 234: 505-511.

Niknafs, A. and S. Parsa, 2011. A neural network approach for updating ranked association rules, based on data envelopment analysis. J. Artif. Intell., 4: 279-287.

Salazar, R., U. Schmidt, C. Huber, A. Rojano and I. Lopez, 2010. Neural networks models for temperature and CO_2 control. Int. J. Agric. Res., 5: 191-200.

Sokkalingam, P.T., R.K. Ahuja and J.B. Orlin, 1999. Solving inverse spanning tree problems through network ?ow techniques. Operat. Res., 47: 291-298.

Tarantola, A., 1987. Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation. Elsevier, Amsterdam, The Netherlands, ISBN-13: 9780444427656, Pages: 613.

Wu, H., L. He, L. Qin, T. Feng and R. Shi, 2010. Solving interval quadratic program with box set constraints in engineering by a projection neural network. Inform. Technol. J., 9: 1615-1621.

Xia, Y. and J. Wang, 1998. A general methodology for designing globally convergent optimizatin neural networks. IEEE Trans. Neural Networks, 9: 1331-1343.

Yang, C., J. Zhang and Z. Ma, 1997. Inverse maximum flow and minimum cut problems. Optimization, 40: 147-170.

Yedjour, H., B. Meftah, D. Yedjour and A. Benyettou, 2011. Combining spiking neural network with hausdorff distance matching for object tracking. Asian J. Applied Sci., 4: 63-71.

Zhang, J., Z. Ma and C. Yang, 1995. A column generation method for inverse shortest paths problems. Math. Meth. Operat. Res., 41: 347-358.

Zhang, J. and Z. Liu, 1996. Calculating some inverse linear programming problems. J. Computat. Applied Math., 72: 261-273.

Zhang, J., Z. Liu and Z. Ma, 1996. On the inverse problem of minimum spanning tree with partition constraints. Math. Meth. Operat. Res., 44: 171-188.

Zhang, J. and M.C. Cai, 1998. Inverse problem of minimum cuts. Math. Meth. Operat. Res., 47: 51-58.

Zhang, J. and Z. Liu, 1999. A further study on inverse linear programming problems. J. Comput. Applied Math., 106: 345-359.