

Neural networks for a class of bi-level variational inequalities

M. H. Xu · M. Li · C. C. Yang

Received: 5 February 2008 / Accepted: 10 September 2008 / Published online: 30 September 2008
© Springer Science+Business Media, LLC. 2008

Abstract In this paper, a class of bi-level variational inequalities for describing some practical equilibrium problems, which especially arise from engineering, management and economics, is presented, and a neural network approach for solving the bi-level variational inequalities is proposed. The energy function and neural dynamics of the proposed neural network are defined in this paper, and then the existence of the solution and the asymptotic stability of the neural network are shown. The simulation algorithm is presented and the performance of the proposed neural network approach is demonstrated by some numerical examples.

Keywords Bi-level variational inequality · Neural networks · Monotone mapping

1 Introduction

In engineering, management, and economics, there are many practical problems can be formulated as the following variational inequalities [3, 8]: Find x such that

$$x \in \Omega_1, \quad (x' - x)^T f(x) \geq 0, \quad \forall x' \in \Omega_1, \quad (1.1)$$

where Ω_1 is a closed convex set in R^n and $f(x)$ is a mapping from R^n into itself. In many cases, the solution x of variational inequalities (1.1) can solve the associated practical problems.

M. H. Xu (✉)

School of Mathematics and Physics, Jiangsu Polytechnic University, Changzhou, Jiangsu Province
213164, People's Republic of China
e-mail: xuminghua@jpu.edu.cn; mathxumh@yahoo.com.cn

M. Li

Department of Management Science and Engineering, School of Economics and Management, Southeast University, Nanjing, Jiangsu Province 210096, People's Republic of China

C. C. Yang

School of Information Science and Engineering, Jiangsu Polytechnic University, Changzhou,
Jiangsu Province 213164, People's Republic of China

However, in some practical problems, it is often required that the solution x of (1.1) must also satisfy some additional constraints, say $x \in \Omega_2$, where $\Omega_2 \subset \Omega_1$ is a closed convex set of R^n . Unfortunately, the solution of (1.1) generally doesn't satisfy the additional constraints. For some of those practical problems, it is very lucky that the additional constraints can be satisfied if we introduce a control variable $y \in R^n$ and can solve the following variational inequalities: Find (x, y) such that

$$x \in \Omega_2, \quad (x' - x)^T y \geq 0, \quad \forall x' \in \Omega_2, \tag{1.2}$$

where x is a solution of the following variational inequalities:

$$x \in \Omega_1, \quad (x' - x)^T [f(x) - y] \geq 0, \quad \forall x' \in \Omega_1. \tag{1.3}$$

Using the terminology in bi-level programming, (1.2) and (1.3) are called the upper-level VI and the lower-level VI, respectively. For convenience, we call (1.2–1.3) a class of bi-level variational inequalities (denoted by $BVI(\Omega_1, \Omega_2, f)$). Now, our question is how to solve the bi-level variational inequalities and thus obtain the value of the control variable y and the corresponding value of variable x .

Throughout of this paper, we assume that f is monotone and Lipschitz continuous on Ω_1 , i.e.,

$$(x - \tilde{x})^T (f(x) - f(\tilde{x})) \geq 0, \quad \forall x, \tilde{x} \in \Omega_1,$$

and there is a constant $L > 0$ such that

$$\|f(x) - f(\tilde{x})\| \leq L\|x - \tilde{x}\|, \quad \forall x, \tilde{x} \in \Omega_1.$$

Moreover, we assume that the solution set of $BVI(\Omega_1, \Omega_2, f)$, denoted by Ω^* , is nonempty.

For every solution point of $BVI(\Omega_1, \Omega_2, f)$, $(x^*, y^*) \in \Omega^*$, we have

$$x(y^*) \in \Omega_1, \quad (x - x(y^*))^T [f(x(y^*)) - y^*] \geq 0, \quad \forall x \in \Omega_1. \tag{1.4}$$

and

$$x(y^*) \in \Omega_2, \quad (x - x(y^*))^T y^* \geq 0, \quad \forall x \in \Omega_2. \tag{1.5}$$

Let $\Omega = \Omega_1 \times \Omega_2 \subset R^l$, $\Omega_3 = (\Omega_1 \cap \Omega_2) \times R^n$, thus, $BVI(\Omega_1, \Omega_2, f)$ can be also viewed as a *general variational inequality* $GVI(\Omega, \Omega_3, F, Q)$ [6], Find $u^* = (x^{*T}, y^{*T})^T \in \Omega_3$ such that

$$F(u^*) \in \Omega \quad \text{and} \quad (F - F(u^*))^T Q(u^*) \geq 0, \quad \forall F \in \Omega, \tag{1.6}$$

where

$$F(u) = \begin{pmatrix} x \\ x \end{pmatrix}, \quad Q(u) = \begin{pmatrix} f(x) - y \\ y \end{pmatrix}. \tag{1.7}$$

A general variational inequality $GVI(\Omega, \Omega_3, F, Q)$ is monotone if

$$(F(u) - F(\tilde{u}))^T (Q(u) - Q(\tilde{u})) \geq 0, \quad \forall u, \tilde{u} \in \Omega_3.$$

Note that for our investigated problem,

$$\begin{aligned} (F(u) - F(\tilde{u}))^T (Q(u) - Q(\tilde{u})) &= (x - \tilde{x})^T (f(x) - f(\tilde{x})) \\ &\geq 0, \quad \forall u, \tilde{u} \in \Omega_3, \end{aligned}$$

and thus $GVI(\Omega, \Omega_3, F, Q)$ (1.6–1.7) is monotone whenever the mapping f is. The existence of the solution of problem $GVI(\Omega, \Omega_3, F, Q)$ is investigated by Pang and Yao [12].

For solving monotone GVI(Ω, Ω_3, F, Q), there are some methods in the literature [5, 6, 11]. Since our original interest is to solve problem (1.2–1.3) and the neural networks method is very efficient for solving some variational inequalities [1, 2, 7], thus, in this paper we present a simple neural network model for this GVI(Ω, Ω_3, F, Q) with special structure (1.7).

In the next Section, a neural network model for solving the bi-level variational inequalities (1.2–1.3), and some preliminaries are presented. The stability analysis, simulation algorithm and convergence properties of the neural network model are studied in Sect. 3 and 4. Applications and illustrative examples are given in Sect. 5 and 6, respectively.

2 Neural network model

Let $P_\Omega : R^l \rightarrow \Omega$ be a projection operator defined by

$$P_\Omega(v) = \operatorname{argmin}\{\|v - u\| \mid u \in \Omega\}.$$

Then the basic properties of the projection on a closed convex set are [8]

$$(v - P_\Omega(v))^T(u - P_\Omega(v)) \leq 0, \quad \forall v \in R^l, \forall u \in \Omega, \tag{2.1}$$

$$\|P_\Omega(u) - P_\Omega(v)\| \leq \|u - v\|, \quad \forall u, v \in R^l. \tag{2.2}$$

Using the projection notation, the general variational inequality (1.6) is equivalent to the following projection equation (e.g., see [6] for a proof)

$$F(u) = P_\Omega[F(u) - Q(u)]. \tag{2.3}$$

Let Ω^* denote the solution set of BVI(Ω_1, Ω_2, f) and

$$e(u, r) := \begin{pmatrix} e_x(u, r) \\ e_y(u, r) \end{pmatrix} = \begin{pmatrix} x - P_{\Omega_1}[x - (1/r)(f(x) - y)] \\ (1/r)\{x - P_{\Omega_2}[x - ry]\} \end{pmatrix} \tag{2.4}$$

denote the scaled *residue* of (2.3). It follows that for any $r > 0$,

$$u \in \Omega^* \iff e(u, r) = 0.$$

For given u and r , throughout of this paper, we use the notations

$$\tilde{u} = u - e(u, r) \tag{2.5}$$

and

$$\xi = \begin{pmatrix} (f(x) - f(\tilde{x})) - (y - \tilde{y}) \\ x - \tilde{x} \end{pmatrix}. \tag{2.6}$$

In order to obtain a convergent neural network method, we assume that r is chosen such that u, \tilde{u} and ξ (see (2.5–2.6)) satisfy

$$\|\xi\| \leq rv\|u - \tilde{u}\| \quad \text{with } v \in (0, 1). \tag{2.7}$$

Under the assumption that f is Lipschitz continuous with Lipschitz constant L , we have

$$\|\xi\| \leq \|f(x) - f(\tilde{x})\| + \|u - \tilde{u}\| \leq (L + 1)\|u - \tilde{u}\|,$$

and condition (2.7) is satisfied when $r \geq (L + 1)/\nu$. In practical computation, we can get such r by the following test procedure: For give u and a trial parameter r , we obtain \tilde{u} and ξ by (2.5–2.6), then calculate the ratio

$$t := \|\xi\|/\|r(u - \tilde{u})\|.$$

If $t \leq \nu$, r is accepted. Otherwise, enlarge r by a factor $\mu > 1$ and repeat the procedure.

Now, for any $u^* \in \Omega^*$, let $E(u) = \|u - u^*\|^2$ be the energy function. It is easy to know that $E(u)$ is a differentiable convex function and $E(u) = 0$ if and only if u is a solution of (1.2–1.3). Thus, our objective is to find the solution to

$$\min_{u \in \Omega_3} E(u). \tag{2.8}$$

To solve the problem (1.2–1.3), let us define the dynamics of the proposed neural network to be

$$\frac{du(t)}{dt} = -d(u), \quad u \in \Omega_3, \tag{2.9}$$

where $d(u) = e(u, r) - \xi/r$.

The stability analysis of the neural network will be established in next section.

3 Stability analysis

In order to establish the convergence of the neural network (2.9), we first prove the following lemmas and theorems.

Lemma 3.1 *Suppose that the mapping f is Lipschitz continuous with Lipschitz constant L on \mathbb{R}^n , i.e.,*

$$\|f(x^1) - f(x^2)\| \leq L\|x^1 - x^2\|, \quad \forall x^1, x^2 \in \mathbb{R}^n,$$

then $d(u)$ is Lipschitz continuous on Ω_3 .

Proof Let

$$v^1 = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix}, \quad v^2 = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} \in \Omega_3,$$

and for $i = 1, 2$, let

$$e(v^i, r) = \begin{pmatrix} x^i - P_{\Omega_1}[x^i - (1/r)(f(x^i) - y^i)] \\ (1/r)\{x^i - P_{\Omega_2}[x^i - ry^i]\} \end{pmatrix},$$

$$\tilde{v}^i = \begin{pmatrix} \tilde{x}^i \\ \tilde{y}^i \end{pmatrix} = v^i - e(v^i, r)$$

and

$$\xi^i = \begin{pmatrix} (f(x^i) - f(\tilde{x}^i)) - (y^i - \tilde{y}^i) \\ x^i - \tilde{x}^i \end{pmatrix}.$$

Using (2.2), we have

$$\begin{aligned} \|P_{\Omega_2}[x^1 - ry^1] - P_{\Omega_2}[x^2 - ry^2]\| &\leq \|x^1 - x^2\| + r\|y^1 - y^2\| \\ &\leq (1 + r)\|v^1 - v^2\| \end{aligned} \tag{3.1}$$

and

$$\begin{aligned} &\|P_{\Omega_1}[x^1 - (1/r)(f(x^1) - y^1)] - P_{\Omega_1}[x^2 - (1/r)(f(x^2) - y^2)]\| \\ &\leq (1 + \frac{L}{r})\|x^1 - x^2\| + \frac{1}{r}\|y^1 - y^2\| \\ &\leq (1 + \frac{1+L}{r})\|v^1 - v^2\|. \end{aligned} \tag{3.2}$$

Combining (3.1) and (3.2), we obtain

$$\begin{aligned} &\|e(v^1, r) - e(v^2, r)\| \\ &\leq \|x^1 - x^2 - (P_{\Omega_1}[x^1 - (1/r)(f(x^1) - y^1)] - P_{\Omega_1}[x^2 - (1/r)(f(x^2) - y^2)])\| \\ &\quad + \frac{1}{r}\|x^1 - x^2 - (P_{\Omega_2}[x^1 - ry^1] - P_{\Omega_2}[x^2 - ry^2])\| \\ &\leq (2 + \frac{1+L}{r})\|v^1 - v^2\| + (1 + \frac{2}{r})\|v^1 - v^2\| \\ &= (3 + \frac{3+L}{r})\|v^1 - v^2\| \\ &= L_1\|v^1 - v^2\|, \end{aligned} \tag{3.3}$$

where $L_1 = (3 + \frac{3+L}{r})$.

Following from (3.3), we have

$$\|\tilde{v}^1 - \tilde{v}^2\| \leq (1 + L_1)\|v^1 - v^2\|, \tag{3.4}$$

and hence

$$\|\tilde{x}^1 - \tilde{x}^2\| \leq (1 + L_1)\|v^1 - v^2\|, \tag{3.5}$$

$$\|\tilde{y}^1 - \tilde{y}^2\| \leq (1 + L_1)\|v^1 - v^2\|. \tag{3.6}$$

By a simple computation, we obtain

$$\begin{aligned} \|\xi^1 - \xi^2\| &\leq \|((f(x^1) - f(\tilde{x}^1)) - (y^1 - \tilde{y}^1)) - ((f(x^2) - f(\tilde{x}^2)) - (y^2 - \tilde{y}^2))\| \\ &\quad + \|(x^1 - \tilde{x}^1) - (x^2 - \tilde{x}^2)\| \\ &\leq L(\|x^1 - x^2\| + \|\tilde{x}^1 - \tilde{x}^2\|) + \|y^1 - y^2\| + \|\tilde{y}^1 - \tilde{y}^2\| \\ &\quad + \|x^1 - x^2\| + \|\tilde{x}^1 - \tilde{x}^2\| \\ &\leq (L + 1)(\|x^1 - x^2\| + \|\tilde{x}^1 - \tilde{x}^2\|) + \|y^1 - y^2\| + \|\tilde{y}^1 - \tilde{y}^2\| \\ &\leq (2 + L)\|v^1 - v^2\| + (2 + L)(1 + L_1)\|v^1 - v^2\| \\ &= L_2\|v^1 - v^2\|, \end{aligned} \tag{3.7}$$

where $L_2 = (2 + L)(2 + L_1)$. Following from (3.3) and (3.7), we have

$$\|d(v^1) - d(v^2)\| \leq (L_1 + \frac{L_2}{r})\|v^1 - v^2\|, \tag{3.8}$$

and the proof is completed. □

Lemma 3.2 *Let $u^* \in \Omega^*$ be a solution of (1.2–1.3), u, ξ , and r satisfy the inequality (2.7), then*

$$\begin{aligned} \frac{dE(u)}{dt} &= 2(u - u^*)^T \frac{du}{dt} \\ &= -2(u - u^*)^T d(u) \\ &\leq -(1 - \nu)\|e(u, r)\|^2. \end{aligned} \tag{3.9}$$

Proof It follows from (2.4) and (2.5) that

$$\tilde{x} = P_{\Omega_1}[x - (1/r)(f(x) - y)] \tag{3.10}$$

and

$$\tilde{y} = (1/r)\{P_{\Omega_2}[x - ry] - (x - ry)\}. \tag{3.11}$$

Since $\tilde{x} \in \Omega_1$, according to (1.4), we have

$$(\tilde{x} - x^*)^T (f(x^*) - y^*) \geq 0. \tag{3.12}$$

Because $x^* \in \Omega_1$ and \tilde{x} is the projection of $x - (1/r)(f(x) - y)$ onto Ω_1 , it follows from (2.1) that

$$(\tilde{x} - x^*)^T r\{x - (1/r)(f(x) - y) - \tilde{x}\} \geq 0. \tag{3.13}$$

Adding (3.12) and (3.13), we get

$$(\tilde{x} - x^*)^T \{r(x - \tilde{x}) - (f(x) - f(x^*)) + (y - y^*)\} \geq 0. \tag{3.14}$$

On the other hand, since $P_{\Omega_2}[x - ry] \in \Omega_2$, according to (1.5), we have

$$\{P_{\Omega_2}[x - ry] - x^*\}^T ry^* \geq 0. \tag{3.15}$$

Because $x^* \in \Omega_2$, it follows from (2.1) that

$$\{P_{\Omega_2}[x - ry] - x^*\}^T \{[x - ry] - P_{\Omega_2}[x - ry]\} \geq 0. \tag{3.16}$$

Adding (3.15) and (3.16), and using (3.11), we get

$$\{(x - x^*) - r(y - \tilde{y})\}^T \{r(y^* - \tilde{y})\} \geq 0$$

and thus

$$(\tilde{y} - y^*)^T \{r(y - \tilde{y}) - (x - x^*)\} \geq 0. \tag{3.17}$$

Combining (3.14) and (3.17), we obtain

$$\begin{pmatrix} \tilde{x} - x^* \\ \tilde{y} - y^* \end{pmatrix}^T \begin{pmatrix} r(x - \tilde{x}) - (f(x) - f(x^*)) + (y - y^*) \\ r(y - \tilde{y}) - (x - x^*) \end{pmatrix} \geq 0. \tag{3.18}$$

It follows from the monotonicity of f that

$$\begin{pmatrix} \tilde{x} - x^* \\ \tilde{y} - y^* \end{pmatrix}^T \begin{pmatrix} (f(\tilde{x}) - f(x^*)) - (\tilde{y} - y^*) \\ \tilde{x} - x^* \end{pmatrix} \geq 0. \tag{3.19}$$

Adding (3.18) and (3.19), we get

$$\begin{pmatrix} \tilde{x} - x^* \\ \tilde{y} - y^* \end{pmatrix}^T \begin{pmatrix} r(x - \tilde{x}) - (f(x) - f(\tilde{x})) + (y - \tilde{y}) \\ r(y - \tilde{y}) - (x - \tilde{x}) \end{pmatrix} \geq 0. \tag{3.20}$$

Following from (3.20) and the notations of $e(u, r)$ and ξ , we have

$$(u - u^*)^T d(u) \geq e(u, r)^T d(u). \tag{3.21}$$

Note that under Condition (2.7) we have

$$e(u, r)^T d(u) \geq (1 - \nu) \|e(u, r)\|^2. \tag{3.22}$$

Assertion (3.9) follows from (3.21) and (3.22), immediately. □

Furthermore, Lemma 3.2 implies that $E(u)$ is non-increasing with respect to t . Thus the set $U^+ = \{u(t) | t \geq 0\} \subset \Omega_3$ is bounded. In addition, following from (2.6) and (2.7), we have

$$\begin{aligned} e(u, r)^T d(u) &= \|e(u, r)\|^2 - e(u, r)^T \xi / r \\ &> \frac{1}{2} \|e(u, r)\|^2 - e(u, r)^T \xi / r + \frac{1}{2} \|\xi / r\|^2 \\ &= \frac{1}{2} \|d(u)\|^2. \end{aligned} \tag{3.23}$$

Lemma 3.3 $u \in \Omega_3$ is a solution of bi-level variational inequalities (1.2–1.3) if and only if u is an equilibrium point of the dynamic system (2.9).

Proof It follows from (2.4) that $u \in \Omega_3$ is a solution of bi-level variational inequalities (1.2–1.3) if and only if $e(u, r) = 0$ for any $r > 0$. Using (2.7) and (2.5), we have

$$\|d(u)\| \geq \|e(u, r)\| - \|\xi\|/r \geq \|e(u, r)\| - \nu \|u - \tilde{u}\| = (1 - \nu) \|e(u, r)\| \tag{3.24}$$

and

$$\|d(u)\| \leq \|e(u, r)\| + \|\xi\|/r \leq \|e(u, r)\| + \nu \|u - \tilde{u}\| = (1 + \nu) \|e(u, r)\| \tag{3.25}$$

Combining (3.24) and (3.25), we obtain

$$e(u, r) = 0 \iff d(u) = 0.$$

The proof is completed. □

By Lemma 3.1 and [4], we have

Theorem 3.1 For any $u^0 \in \Omega_3$, the initial value problem of the system of differential equations (2.9) has unique solution.

As a result of Lemma 3.2 and 3.3, we have the following theorem.

Theorem 3.2 Suppose that the mapping f is monotone on R^n , and $u^* \in \Omega^*$ is a solution of (1.2–1.3). Then u^* is asymptotically stable in the sense of Liapunov.

Proof According to Lemma 3.3, we know that u^* is an equilibrium point of (2.9). Since for any $u(t) \notin \Omega^*$, $e(u(t), r) \neq 0$. Thus, for $u(t) \notin \Omega^*$, by Lemma 3.2, we have

$$\frac{dE(u)}{dt} \leq -(1 - \nu)\|e(u(t), r)\|^2 < 0.$$

This means that $\frac{dE(u)}{dt}$ is negative definite. Therefore, u^* is asymptotically stable in the large [4]. The proof is completed. \square

4 Simulation algorithm

Since the neural network (2.9) is actually the initial value problem of a differential equation, the simplest simulation algorithm is the Euler method. Thus, we now use the Euler method to solve the neural network (2.9), and have the following algorithm:

Algorithm 1:

Step 1: Initialization:

Let $t_0 = 0$. Choose arbitrarily an initial point $u(t_0) = u^0 \in \Omega_3$.

Set $k = 0, \mu > 1, r > 0, \gamma \in (0, 2), \nu \in (0, 1)$ and $\varepsilon > 0$ be a pre-specified tolerance.

Step 2: Computation:

Compute $e(u^k, r), \xi^k$ by using (2.4) and (2.6), respectively.

If $\|e(u^k, r)\| < \varepsilon$,

then stop and output u^k ,

otherwise

Do while $\rho = \|\xi^k\|/\|re(u^k, r)\| > \nu$
 $r := r \times \mu$.

End Do

End If

$d(u^k) = e(u^k, r) - \xi^k/r$,

$\alpha_k^* = e(u^k, r)^T d(u^k)/\|d(u^k)\|^2$,

$\alpha_k = \gamma \times \alpha_k^*$.

Step 3: States Updating:

$t_{k+1} = t_k + \alpha_k$,

$u(t_{k+1}) = u^{k+1} = u^k - \alpha_k d(u^k)$.

Go to Step 2. \square

For Algorithm 1, we have the following convergence results.

Theorem 4.1 Let $u^* \in \Omega^*$, sequence $\{u^k\}$ be generated by Algorithm 1, then

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \gamma(2 - \gamma)\alpha_k^* e(u^k, r)^T d(u^k), \quad \forall u^* \in \Omega^*. \tag{4.1}$$

Proof By using (3.21), we obtain

$$\begin{aligned} \|u^{k+1} - u^*\|^2 &= \|u^k - u^* - \alpha_k d(u^k)\|^2 \\ &\leq \|u^k - u^*\|^2 - 2\alpha_k e(u^k, r)^T d(u^k) + \alpha_k^2 \|d(u^k)\|^2. \end{aligned}$$

Substituting $\alpha_k = \gamma\alpha_k^*$ and $\alpha_k^* = e(u^k, r)^T d(u^k)/\|d(u^k)\|^2$ in the above inequality, we obtain (4.1) immediately. \square

By using (3.22) and (3.23), we have $\alpha_k^* > \frac{1}{2}$ and (4.1) can be written as

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - c\|e(u^k, r)\|^2, \quad \forall u^* \in \Omega^*, \tag{4.2}$$

where $c = \frac{\gamma(2-\gamma)(1-\nu)}{2} > 0$ is a constant. The convergence of Algorithm 1 can be deduced from (4.2) immediately.

5 Applications

There are many practical problems which can be described by bi-level variational inequalities, especially in the area of network management and control. In this section, we mainly outline the application of bi-level variational inequalities in traffic assignment or traffic equilibrium problem.

We consider a transportation network G . Let W be the set of origin-destination (OD) pairs with N elements. OD pairs are denoted by $w, w = 1, 2, \dots, N$. d_w is the fixed demands associated with OD pair w . Let P_w be the set of paths joining the OD pair w with $|P_w|$ elements, P the set of paths with n elements. We group together the travel demands into a vector d in R^N . The travel demand d induces a flow vector $x \in R^n$ with defined on every path $p \in P$ by

$$d_w = \sum_{p \in P_w} x_p.$$

A user traveling on path p incurs a travel cost $F_p(x)$. We group together the travel cost $F_p(x)$ into a vector $F(x)$ in R^n . A feasible load pattern F is in equilibrium if, once established, no user has any incentive to change his travel arrangements. This state is characterized by the following equilibrium conditions which must hold for every OD pair $w \in W$ and every path $p \in P_w$,

$$F_p(x) \begin{cases} = \lambda_w, & \text{if } x_p > 0 \\ \geq \lambda_w, & \text{if } x_p = 0 \end{cases} \tag{5.1}$$

where λ_w is the optimal travel cost associated with OD pair w [10]. Let λ_p be the optimal travel cost associated with $F_p(x)$ defined in (5.1). And group together λ_p into a vector λ in R^n . Then, the above user equilibrium conditions are equivalent to the nonlinear complementarity problem (NCP) [9]:

$$x \geq 0, F(x) - \lambda \geq 0, x^T(F(x) - \lambda) = 0. \tag{5.2}$$

It is well known that nonlinear complementarity problem (5.2) is equivalent to the variational inequality: Find $x \in R_+^n$ such that

$$(x' - x)^T f(x) \geq 0, \forall x' \in R_+^n, \tag{5.3}$$

where $f(x) = F(x) - \lambda, R_+^n$ is the nonnegative orthant of R^n .

In order to control traffic assignment or rationally use the road services, the administration should find out a charge policy $y_p \geq 0$ on path $p \in P$, such that the relevant path flows $x_p(y)$ can be limited in a satisfied range, say $x_p(y) \leq b_p$. This is a road pricing problem which can be described as a nonlinear complementarity problem:

$$0 \leq y \perp b - x(y) \geq 0, \tag{5.4}$$

where $y \in R^n$ and $b \in R^n$ are vector with components y_p and b_p , respectively. Similar to (5.1), $x_p(y)$ must satisfy the following equilibrium conditions:

$$F_p(x) \begin{cases} = \lambda_p + y_p, & \text{if } x_p > 0 \\ \geq \lambda_p + y_p, & \text{if } x_p = 0 \end{cases}. \tag{5.5}$$

We know that problem (5.4) and equilibrium conditions (5.5) are equivalent to the following variational inequalities: Find (x, y) such that

$$x \in \Omega_2, (x' - x)^T y \geq 0, \forall x' \in \Omega_2, \tag{5.6}$$

where $\Omega_2 = \{x | x \leq b, x \in R^n\}$ and x is a solution of the following variational inequalities

$$x \in \Omega_1, (x' - x)^T (f(x) - y) \geq 0, \forall x' \in \Omega_1, \tag{5.7}$$

where $\Omega_1 = R^n_+, f(x) = F(x) - \lambda$ as in (5.3).

Now, we see that the problem of controlling traffic assignment or rationally using the road services can be expressed as the bi-level variational inequalities (1.2–1.3).

6 Illustrative examples

In this section, some examples are provided to illustrate both the theoretical results achieved in Sect. 3 and the simulation performance of Algorithm 1 for the dynamical system (2.9). In the following illustrative examples, the computer program for implementing Algorithm 1 is coded in MATLAB and the program runs on IBM notebook(R51).

Example 1 Consider the bi-level variational inequalities (1.2–1.3) with $n = 10, \Omega_1 = R^n_+, \Omega_2 = \{x | a \leq x \leq b, x \in R^n\}, f(x) = Ax - q$, where

$$\begin{aligned} a &= [0.5, 0.2, 1.0, 0.3, 1.1, 0.8, 2.1, 0.7, 3.1, 1.0]^T \in R^n, \\ b &= [1.2, 1.3, 2.1, 1.3, 2.2, 3.1, 4.2, 1.4, 5.6, 2.1]^T \in R^n, \\ q &= [-3.0, 5.2, -6.4, 7.2, -7.0, 4.2, 8.6, -5.2, 2.4, 3.0]^T \in R^n \end{aligned}$$

and $A = [a_{ij}] \in R^{n \times n}$, where $a_{ij} = \frac{0.1}{i+j-1}$, i.e.,

$$A = 0.1 \times \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \dots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \dots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \dots & \frac{1}{2n-1} \end{bmatrix} \in R^{n \times n}.$$

It is well known that A is positive definite and thus $f(x)$ is monotone with respect to x . Now, we solve the problem (1.2–1.3) by the neural network method with $x_0 = b, y_0 = 0 \in R^n, u_0 = (x_0^T, y_0^T)^T, \mu = 1.2, r = 0.5, \gamma = 1.8$ and $\nu = 0.8$. The stopping test is

$$\|e(u, r)\| \leq 10^{-7}.$$

The solution of this problem is that

$$\begin{aligned} x^* &= (0.5000, 1.3000, 1.0000, 1.3000, 1.1000, 3.1000, 4.2000, 0.7000, 5.6000, \\ &\quad 2.1000)^T \\ y^* &= (3.4065, -4.8827, 6.6674, -6.9669, 7.2075, -4.0126, -8.4289, 5.3575, \\ &\quad -2.2539, -2.8638)^T \end{aligned}$$

Figure 1 shows the trajectories of the neural network with the initial values above, where solid lines and dashed lines are the trajectories of x and y , respectively. Furthermore, the Matlab code for Example 1 is given in Table 1 .

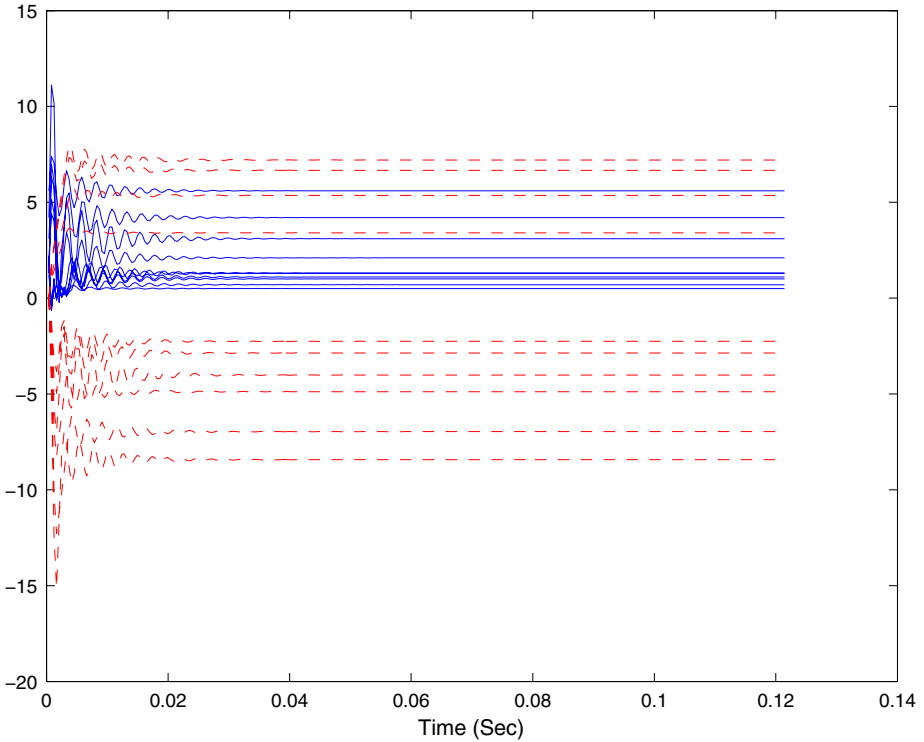


Fig. 1 Transient behavior of the neural network (2.9) in Example 1

Table 1 Matlab code for Example 1

```
% Matlab Program for Manuscript: Neural networks for a class
% of bi-level
% variational inequalities. Written by Xu Ming Hua.

function bilevelv1()
% Initialization
clc; n=10; zn=zeros(n,1); t=0; A=zeros(n,n);

for i=1:n
    for j=1:n
        A(i,j)=1/(i+j-1);
    end
end

A=0.1*A; b=2*[-1.5,2.6,-3.2,3.6,-3.5,2.1,4.3,-2.6,1.2,1.5]';
% Omega 1
ub=[1.2,1.3,2.1,1.3,2.2,3.1,4.2,1.4,5.6,2.1]';
lb=[0.5,0.2,1.0,0.3,1.1,0.8,2.1,0.7,3.1,1.0]';
x0=ub;y0=0*lb;u0=[x0;y0]; k=0; max_iter=500; mu=1.2; r=0.5;
gamma=1.8; nu=0.8; eps=10^(-7); trajectory_u=[u0];
```

```

% Begin computation
tic;
eu1=x0-max(zn,x0-(f(A,b,x0)-y0)/r);eu2=(x0-max(lb,min(ub,x0
-r*y0)))/r;
eu=[eu1;eu2]; ubar=u0-eu;xbar=ubar(1:n); ybar=ubar(n+1:2*n);
xi1=f(A,b,x0)-f(A,b,xbar)-y0+ybar; xi2=x0-xbar; xi=[xi1;xi2];
neu=norm(eu); nxi=norm(xi);

while neu > eps & k < max_iter
    k=k+1; nuo=nxi/(r*neu);

    while nuo > nu
        r=r*mu; eu1=x0-max(zn,x0-(f(A,b,x0)-y0)/r);
        eu2=(x0-max(lb,min(ub,x0-r*y0)))/r; eu=[eu1;eu2];
        ubar=u0-eu; xbar=ubar(1:n);
        ybar=ubar(n+1:2*n); xi1=f(A,b,x0)-f(A,b,xbar)
            -y0+ybar;
        xi2=x0-xbar; xi=[xi1;xi2]; neu=norm(eu);
        nxi=norm(xi); nuo=nxi/(r*neu);
    end

    du=eu-xi/r; ndu=norm(du); alpha_star=eu'*du/ndu^2;
    alpha=gamma*alpha_star; t=t+alpha; u0=u0-alpha*du;
    x0=u0(1:n); y0=u0(n+1:2*n);
    trajectory_u=[trajectory_u,u0];
    eu1=x0-max(zn,x0-(f(A,b,x0)-y0)/r);
    eu2=(x0-max(lb,min(ub,x0-r*y0)))/r;
    eu=[eu1;eu2]; ubar=u0-eu; xbar=ubar(1:n);
    ybar=ubar(n+1:2*n);
    xi1=f(A,b,x0)-f(A,b,xbar)-y0+ybar; xi2=x0-xbar;
    xi=[xi1;xi2];
    neu=norm(eu); nxi=norm(xi);
end

% Output the results.
elapsed_time =toc;
fprintf('number of iterations = %6d\n',k);

if k < max_iter
    disp(['The method is convergent within the tolerant
        number... of iterations.']);
else
    disp(['The method does not converge within', ...
        num2str(max_iter),' iterations!']);
end

fprintf('norm of e(u,r)=%8.4e\n',neu); fprintf
('The solution (x;y)=\n');
fprintf('%8.4f\n',u0); fprintf

```

```

('elapsed_time=%8.4f\n', toc);
time_unit=elapsed_time/k;
axis_x=linspace(1,k+1,k+1)*time_unit;

for i=1:n
    plot(axis_x,trajectory_u(i,:), 'b', 'LineWidth', 1.4);
    hold on;
end

for i=1:n
    plot(axis_x,trajectory_u(n+i,:), 'r--', 'LineWidth', 1.4);
    hold on;
end

xlabel('Time (Sec)'); print('-depsc', 'C:\fig1'); hold off;
return

% subfunction
function y=f(A,b,x)
y=A*x-b;
return
    
```

Example 2 Let $n = 10$, $\Omega_1 = R_+^n$, $\Omega_2 = \{x|a \leq x \leq b, x \in R^n\}$, $f(x) = 0.05(Ax - q)$, where

$$\begin{aligned}
 a &= [7.5, 7.2, 11.0, 6.3, 11.1, 20.8, 22.1, 5.7, 33.1, 11.0]^T \in R^n, \\
 b &= [10.2, 12.3, 22.1, 13.3, 21.2, 33.1, 42.2, 11.4, 53.6, 28.1]^T \in R^n, \\
 q &= [150, -260, 320, -360, 350, 210, -430, -260, 120, -150]^T \in R^n
 \end{aligned}$$

and $A = [a_{ij}] \in R^{n \times n}$, where $a_{ij} = 2 \min\{i, j\} - 1$, i.e.,

$$A = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 3 & 3 & \dots & 3 \\ 1 & 3 & 5 & \dots & 5 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 3 & 5 & \dots & 2n - 1 \end{bmatrix} \in R^{n \times n}.$$

It is well known that A is also positive definite and thus $f(x)$ is monotone with respect to x . Now, we solve problem (1.2–1.3) by the neural network method with $x_0 = b$, $y_0 = 0 \in R^n$, $u_0 = (x_0^T, y_0^T)^T$, and the other parameters are same as example 1. The solution of this problem is that

$$\begin{aligned}
 x^* &= (10.2000, 7.2000, 11.0000, 6.3000, 11.1000, 20.8000, 22.1000, 5.7000, \\
 &\quad 33.1000, 11.0000)^T \\
 y^* &= (-0.5750, 32.7550, 15.8650, 60.8750, 35.7550, 52.0250, 91.2150, \\
 &\quad 87.6950, 73.1050, 87.7050)^T
 \end{aligned}$$

Figure 2 shows the trajectories of the neural network with the initial values above, where solid lines and dashed lines are the trajectories of x and y , respectively.

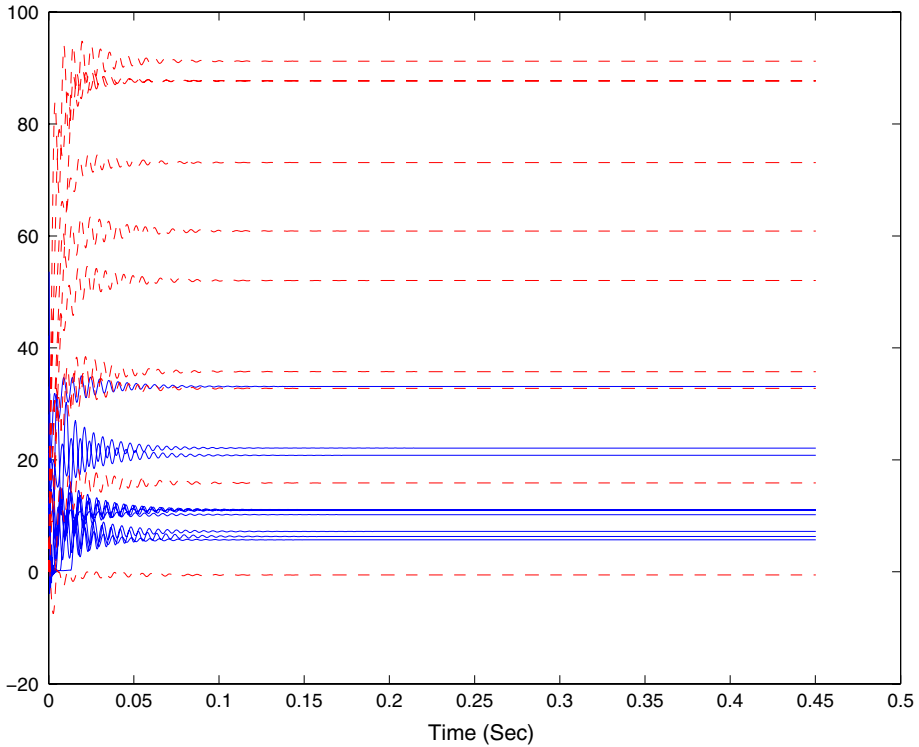


Fig. 2 Transient behavior of the neural network (2.9) in Example 2

Example 3 In this example, we consider the bi-level variational inequalities (1.2–1.3) with nonlinear function

$$f(x) = [\arctan(x_1), \arctan(x_2), \dots, \arctan(x_n)]^T + Ax + q,$$

and $n = 10$, $\Omega_1 = R_+^n$, $\Omega_2 = \{x | a \leq x \leq b, x \in R^n\}$, where

$$a = [6.5, 5.2, 11.0, 8.3, 11.1, 20.8, 32.1, 10.7, 33.1, 18.0]^T \in R^n,$$

$$b = [11.2, 12.3, 21.1, 13.3, 22.2, 31.1, 42.2, 18.4, 52.6, 27.1]^T \in R^n,$$

and

$$A = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{bmatrix} \in R^{n \times n}, \quad q = \begin{bmatrix} 1 - n/2 \\ 2 - n/2 \\ 3 - n/2 \\ \vdots \\ n/2 - 1 \\ n/2 \end{bmatrix} \in R^n.$$

Now, we solve the nonlinear bi-level variational inequalities (1.2–1.3) by the neural network method with $x_0 = b, y_0 = 0 \in R^n, u_0 = (x_0^T, y_0^T)^T$, and the other parameters are

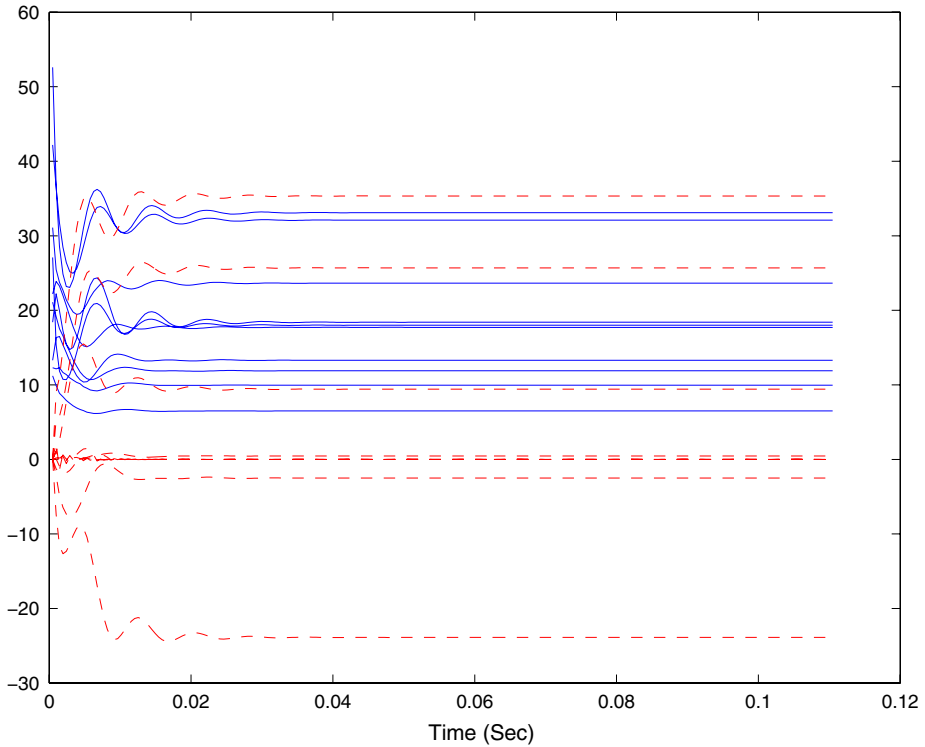


Fig. 3 Transient behavior of the neural network (2.9) in Example 3

same as example 1. We obtain a solution of this problem as follow

$$\begin{aligned}
 x^* &= (6.5000, 9.9572, 11.8852, 13.3000, 17.7142, 23.6428, 32.1000, 18.4000, \\
 &\quad 33.1000, 18.0000)^T \\
 y^* &= (0.4609, -0.0000, -0.0000, -2.5037, -0.0000, 0.0000, 25.6968, \\
 &\quad -23.8835, 35.3406, 9.4153)^T
 \end{aligned}$$

Figure 3 shows the trajectories of the neural network with the initial values above, where solid lines and dashed lines are the trajectories of x and y , respectively.

Example 4 In this example, we also consider the bi-level variational inequalities (1.2–1.3) with nonlinear function

$$f(x) = \begin{bmatrix} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_1 + x_2^2 + 3x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 3x_4 - 1 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{bmatrix}$$

and $n = 4$, $\Omega_1 = R_+^n$, $\Omega_2 = \{x | a \leq x \leq b, x \in R^n\}$, where

$$\begin{aligned}
 a &= [-2.5, -5.0, -7.5, -10.0]^T \in R^n, \\
 b &= [2.5, 5.0, 7.5, 10.0]^T \in R^n.
 \end{aligned}$$

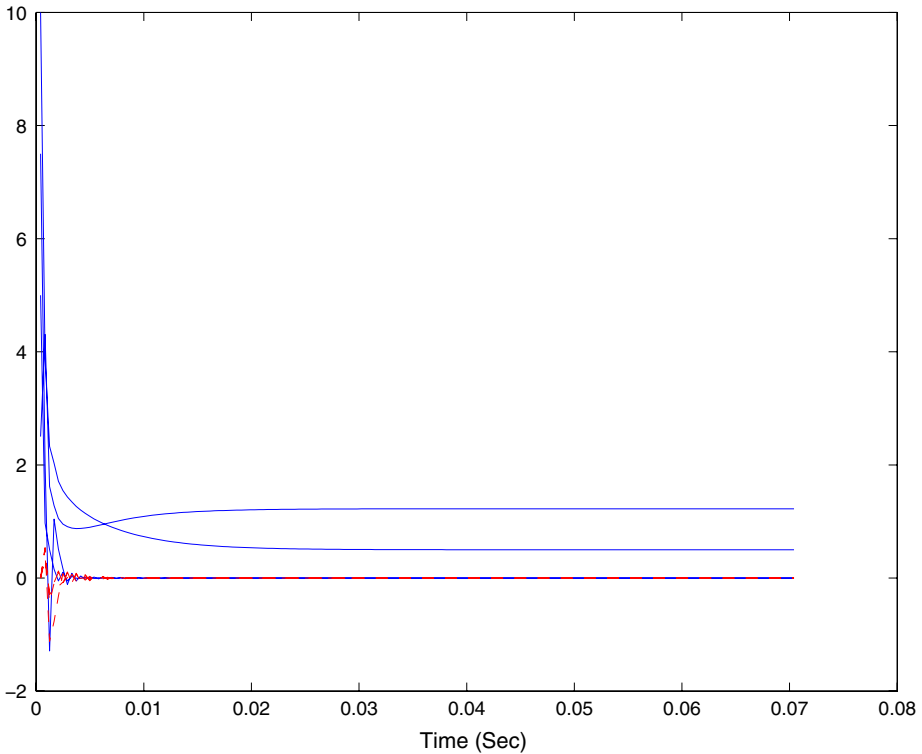


Fig. 4 Transient behavior of the neural network (2.9) in Example 4

Now, we solve the nonlinear bi-level variational inequalities (1.2–1.3) by the neural network method with $x_0 = b, y_0 = 0 \in R^n, u_0 = (x_0^T, y_0^T)^T$, and the other parameters are also same as example 1. We obtain a solution of this problem as follow

$$\begin{aligned} x^* &= (1.2247, -0.0000, -0.0000, 0.5000)^T \\ y^* &= (-0.0000, 0.0000, -0.0000, 0.0000)^T \end{aligned}$$

Figure 4 shows the trajectories of the neural network with the initial values above, where solid lines and dashed lines are also the trajectories of x and y , respectively.

Remark 6.1 Since bi-level variational inequalities (1.2–1.3) can be reformulated to a general variational inequalities (GVI) of form (1.6–1.7), one can try to solve problem (1.2–1.3) via some existing method for (1.6–1.7). For example, excellent paper [6] suggests an implicit method by inexactly solving a series of systems of equations for problem (1.6–1.7) as follows.

Algorithm 2 Inexact implicit method for GVI

Given $u^0 \in R^{2n}, \gamma \in (0, 2)$, a positive matrix G and a nonnegative sequence $\{\eta_k\}$ satisfies $\sum_{k=0}^\infty \eta_k^2 < \infty$. For $k = 0, 1, 2, \dots$, if $u^k \notin S^*$, then finding u^{k+1} such that

$$\|\Theta_k(u^{k+1})\| \leq \eta_k \|e(u^k)\|, \tag{6.1}$$

where S^* is the solution set of the problem (1.6–1.7),

$$e(u) = F(u) - P_\Omega[F(u) - Q(u)], \tag{6.2}$$

$$\Theta_k(u) = F(u) + Q(u) - F(u^k) - Q(u^k) + \gamma\rho(u^k)G^{-1}e(u^k) \tag{6.3}$$

and

$$\rho(u) = \frac{\|e(u)\|^2}{e(u)^T G^{-1} e(u)}. \tag{6.4}$$

Obviously, when $f(x)$ is a nonlinear function, each iteration of the inexact implicit method needs to solve approximately a system of nonlinear equations $\Theta_k(u) = 0$. Although we can use the Matlab function *solve* to give the solution(s) of nonlinear equations $\Theta_k(u) = 0$, it needs Algorithm 2 taking relatively more computing time than Algorithm 1 to solve the problem (1.2–1.3). This is somewhat a disadvantage to Algorithm 2 when $f(x)$ is a nonlinear function. However, in Algorithm 1, we just need to use the value of the function $f(x)$ and compute the projection of a vector on Ω in the iterative processes. This property makes Algorithm 1 easier to be implemented for the variational inequalities we consider.

Furthermore, numerical experiments show that Algorithm 1 and 2 are both efficient for the preceding examples 1 and 2, and Algorithm 2 is less effective than Algorithm 1 for example 3 and 4.

Remark 6.2 Since only the values of function $f(x)$ are used in the iterative processes of Algorithm 1, the method proposed in this paper can be called a direct method. This property makes Algorithm 1 very practical for those variational inequalities arising from the real world, in which the function $f(x)$ usually does not have any explicit expression and only its value can be observed or evaluated for a given variable.

Remark 6.3 It is possible to formulate the general variational inequalities (1.6–1.7) to a problem of form (1.1): Find a vector $(u^*, v^*) \in R^{2n} \times \Omega$, such that

$$\left[\begin{pmatrix} u \\ v \end{pmatrix} - \begin{pmatrix} u^* \\ v^* \end{pmatrix} \right]^T \left[\begin{pmatrix} v^* - F(u^*) \\ Q(u^*) \end{pmatrix} \right] \geq 0, \quad \forall (u, v) \in R^{2n} \times \Omega. \tag{6.5}$$

One can try to solve problem (6.5) via some existing method. But this seems to be unwise because of the large increase in dimensionality [6].

7 Conclusions

In this paper, a special bi-level variational inequalities arising from some practical equilibrium problems is presented. A neural networks for solving this bi-level variational inequalities is constructed. It has been shown theoretically and by numerical results that the trajectory of the neural networks does converge to a solution of the bi-level variational inequalities and the solution of the neural networks is asymptotic stability.

Acknowledgements The authors are grateful to the anonymous referees and associate editors for their valuable comments and helpful suggestions which have greatly improved the exposition of this work. This research is financially supported by a research grant from the Research Grant Council of the People’s Republic of China (Project No. 10571083) and MOEC grant 20060284001.

References

1. Bouzerdorm, A., Pattison, T.R.: Neural network for quadratic optimization with bound constraints. *IEEE Trans. Neural Netw.* **4**, 293–304 (1993)

2. Dang, C., Leung, Y., Gao, X.B., Chen, K.Z.: Neural networks for nonlinear and mixed complementarity problems and their applications. *Neural Netw.* **17**, 271–283 (2004)
3. Ferris, M.C., Pang, J.S.: Engineering and economic applications of complementarity problems. *SIAM Rev.* **39**, 669–713 (1997)
4. Hale, J.K.: *Ordinary Differential Equations*. Wiley/Interscience, New York (1969)
5. He, B.S.: A class of projection and contraction methods for monotone variational inequalities. *Appl. Math. Optim.* **35**, 69–76 (1997)
6. He, B.S.: Inexact implicit methods for monotone general variational inequalities. *Math. Prog.* **86**, 199–217 (1999)
7. He, B.S., Yang, H.: A neural network model for monotone linear asymmetric variational inequalities. *IEEE Trans. Neural Netw.* **11**, 3–16 (2000)
8. Kinderlehrer, D., Stampacchia, G.: *An Introduction to Variational Inequalities and Their Applications*. Academic Press, New York (1980)
9. Lo, H.K., Chen, A.: Traffic equilibrium problem with route-specific cost: formulation and algorithms. *Transport. Res. Part B* **34**, 493–513 (2000)
10. Nagurney, A.B.: Comparative tests of multimodal traffic equilibrium methods. *Transport. Res. Part B* **18**, 469–485 (1984)
11. Pang, J.S., Qi, L.: Nonsmooth equations: motivation and algorithms. *SIAM J. Optim.* **3**, 443–465 (1993)
12. Pang, J.S., Yao, J.C.: On a generalization of a normal map and equation. *SIAM J. Control Optim.* **33**, 168–184 (1995)