Transportation Research Part C 26 (2013) 1–19

Contents lists available at SciVerse ScienceDirect

Transportation Research Part C

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

An efficient dynamic model for solving the shortest path problem

Alireza Nazemi [a,*], Farahnaz Omidi [b]

[a] Department of Mathematics, School of Mathematical Sciences, Shahrood University of Technology, P.O. Box 3619995161-316, Shahrood, Iran
[b] Department of Mathematics, Semnan University, Semnan, Iran

ARTICLE INFO

Article history:
Received 21 February 2012
Received in revised form 11 July 2012
Accepted 11 July 2012

Keywords:
Neural network
Shortest path problem
Linear programming
Convergent
Stability

ABSTRACT

The shortest path problem is the classical combinatorial optimization problem arising in numerous planning and designing contexts. This paper presents a neural network model for solving the shortest path problems. The main idea is to replace the shortest path problem with a linear programming (LP) problem. According to the saddle point theorem, optimization theory, convex analysis theory, Lyapunov stability theory and LaSalle invariance principle, the equilibrium point of the proposed neural network is proved to be equivalent to the optimal solution of the original problem. It is also shown that the proposed neural network model is stable in the sense of Lyapunov and it is globally convergent to an exact optimal solution of the shortest path problem. Several illustrative examples are provided to show the feasibility and the efficiency of the proposed method in this paper.

1. Introduction

The shortest path problem is concerned with finding the shortest path from a specified origin to a specified destination in a given network while minimizing the total cost associated with the path. It is well known that the shortest path problems arise in a wide variety of scientific and engineering applications including vehicle routing in transportation systems, traffic routing in communication networks, path planning in robotic systems, etc. (Ahuja et al., 1993; Antonio et al., 1992; Bazaraa et al., 1990; Bodin et al., 1983; Ephremides and Verdu, 1989; Jun and Shin, 1991; Lin and Chang, 1993; Soueres and Laumond, 1996). Furthermore, the shortest path problem also has numerous variations such as the minimum weight problem, the quickest path problem, the most reliable path problem, and so on.

The shortest path problem has been investigated extensively. The well-known algorithms for solving the shortest path problem include the $O(n^2)$ Bellman's dynamic programming algorithm for directed a cycle networks, the $O(n^2)$ Dijkstra-like labeling algorithm, and the $O(n^3)$ Bellman-Ford successive approximation algorithm for networks with nonnegative cost coefficients only, where n denotes the number of vertices in the network (Lawler, 1976). Besides the classical methods, many new and modified methods have been developed during the past few years. For large-scale and real-time applications such as traffic routing and path planning, the existing series algorithms may not be effective and efficient due to the limitation of sequential processing in computational time. Therefore, parallel solution methods are more desirable.

The dynamic system approach is one of the important methods for solving optimization problems. Artificial recurrent neural networks for solving constrained optimization problems can be considered as a tool to transfer the optimization problems into a specific dynamic system of first-order differential equations. Furthermore, it is expected that the dynamic system will approach its static state (or an equilibrium point), which corresponds to the solution for the underlying optimization problem, starting from an initial point. In addition, neural networks for solving optimization problems are hardware-implementable; that is, the neural networks can be implemented by using integrated circuits.

* Corresponding author. Tel./fax: +98 273 3392012.
E-mail addresses: nazemi20042003@yahoo.com (A. Nazemi), omidi_far@yahoo.com (F. Omidi).

0968-090X/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.trc.2012.07.005

Neural networks for optimization problems were first introduced in the 1980s by Tank and Hopfield (1986). Since then, neural networks have been applied to various optimization problems, including linear and quadratic programming, nonlinear programming, variational inequalities, and linear and nonlinear complementarity problems (Ding and Huang, 2008; Effati et al., 2007; Effati and Jafarzadeh, 2007; Effati and Nazemi, 2006; Friesz et al., 1994; Gao, 2004; Hu, 2009; Hu and Wang, 2007; Kennedy and Chua, 1988; Lillo et al., 1993; Maa and Shanblatt, 1992; Nazemi, 2011a; Tao et al., 2001; Wang, 1996, 1998; Xia, 1996; Xia et al., 2004; Xia and Feng, 2005; Xia and Wang, 2000, 2004a,b; Xue and Bian, 2007, 2009; Yang and Cao, 2006a,b, 2008). In particular, neural networks for solving the shortest path problem have been rather extensively studied and some important results have also been obtained (Effati and Jafarzadeh, 2007; Xia and Wang, 2000; Wang, 1996, 1998). The structures of these models are rather complicated and further simplification can be achieved. Therefore, it is necessary to build a different neural network for (1) and (2) with a simple structure, good stability and convergence results.

Motivated by the above discussions, in the present paper, a neural network model for solving the shortest path problem is presented. On the basis of the Saddle theorem, the equilibrium point of the proposed neural network is proved to be equivalent to the Karush–Kuhn–Tucker (KKT) point of the shortest path problem. The existence and uniqueness of an equilibrium point of the proposed neural network are analyzed. By constructing a suitable Lyapunov function, a sufficient condition to ensure globally stable in the sense of Lyapunov for the unique equilibrium point of neural network is obtained. This neural network model has been also successfully used for solving the minimax problems by Nazemi (2011b).

The remainder of this paper is organized as follows. In Section 2, the problem statement and formulation are described. In Section 3, the system model and some necessary preliminaries are given. The stability and convergence of the proposed neural network are analyzed in Section 4. Numerical simulations are provided in Section 5. Finally, some concluding remarks are drawn in Section 6.

2. Problem formulation

Given a weighted direct graph $G = (V,E)$ where V is a set of n vertices and E is an ordered set of m edges. A fixed cost c_{ij} is associated with the edge from vertices i to j in the graph G. In a transportation or a robotic system, for example, the physical meaning of the cost can be the distance between the vertices, the time or energy needed for travel from one vertex to another. In a telecommunication system, the cost can be determined according to the transmission time and the link capacity from one vertex to another. In general, the cost coefficients matrix $[c_{ij}]$ is not necessarily symmetric, i.e., the cost from vertices i to j may not be equal to the cost from vertices j to i. Furthermore, the edges between some vertices may not exist (i.e., $m < n^2$). The values of cost coefficients for the $n^2 - m$ nonexistent edges are defined as infinity. More generally, a cost coefficient can be either positive or negative. A positive cost coefficient represents a loss, whereas a negative one represents a gain. It is admittedly more difficult to determine the shortest path for a network with mixed positive and negative cost coefficients (Wang, 1998). For an obvious reason, we assume that there are neither negative cycles nor negative loops in the networks (i.e., $\forall i,j; c_{ii} \geqslant 0, \sum_i \sum_j c_{ij} \geqslant 0$). Hence the total cost of the shortest path is bounded from below. Since the vertices in a network can be labeled arbitrarily, without loss of generality, we assume hereafter vertices 1 and n are origin and destination, respectively.

There are many applications of the direct graph since it can be used to model a wide variety of real-world problems. For example, in a road network, the vertices can represent intersections, the edges can represent streets, and the physical meaning of the cost can be the distance between the vertices. In this paper, the shortest path problem to be discussed is: find the shortest (least costly) possible directed path from a specified origin vertex to a specified destination vertex. The cost of the path is the sum of the cost coefficients on the edges in the path and the shortest path is the minimum cost path (Xia and Wang, 2000).

For convenience, we consider the shortest path from vertex 1 to vertex n in a directed graph with n vertices, n edges, and a cost c_{ij} associated with each edge (i,j) in G. In order to formulate the shortest path problem, there are two typical path representations methods: vertex representation and edge representation. Because of the advantages of the edge representation over the vertex representation (Lawler, 1976), the development of this paper is based on the edge path representation. Thus, the shortest path problem can be mathematically formulated as a linear integer program as follows (Wang, 1998):

$$\text{minimize} \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} c_{ij} x_{ij} \tag{1}$$

$$\text{subject to} \sum_{k=1,k\neq i}^{n} x_{ik} - \sum_{l=1,l\neq i}^{n} x_{li} = \begin{cases} 1 & \text{if } i = 1, \\ 0 & \text{if } i = 2,3,\ldots,n-1, \\ -1 & \text{if } i = n, \end{cases} \tag{2}$$

$$x_{ij} \in \{0,1\}, \quad i \neq j, \quad i,j = 1,2,\ldots,n, \tag{3}$$

where the objective function to be minimized, (1), is also the total cost for the path. The equality constraint coefficients and the right-hand sides are −1, 0, or 1. Eq. (2) ensures that a continuous path starts from a specified origin and ends at a specified destination. x_{ij} denotes the decision variable associated with the edge from vertices i to j as defined below

$$x_{ij} = \begin{cases} 1, & \text{if the edge from vertices } i \text{ to } j \text{ is in the path;} \\ 0, & \text{otherwise.} \end{cases}$$

Because of the total unimodulity property of the constraint coefficient matrix defined in (2) (Bazaraa et al., 1990), the integrality constraint in the shortest path problem formulation can be equivalently replaced with the nonnegativity constraint, if the shortest path is unique. In other words, the optimal solutions of the equivalent LP problem are composed of zero and one integers if a unique optimum exists (Bazaraa et al., 1990). The equivalent LP problem based on the simplified edge path representation can be described as follows:

$$\text{minimize} \sum_{i=1}^{n} \sum_{j=2, j\neq i}^{n} c_{ij} x_{ij} \tag{4}$$

$$\text{subject to} \sum_{k=1,k\neq i}^{n-1} x_{ik} - \sum_{l=2,l\neq i}^{n} x_{li} = \delta_{i1} - \delta_{in}, \quad i = 1,2,\ldots,n, \tag{5}$$

$$x_{ij} \geqslant 0, \quad i \neq j, \quad i = 1,2,\ldots,n-1, \quad j = 2,3,\ldots,n, \tag{6}$$

where δ_{ij} is the Kronecker delta function defined as $\delta_{ij} = 1(i = j)$, and $\delta_{ij} = 0(i \neq j)$.

Based on the edge path representation, the dual shortest path problem can be formulated as a LP problem as follows (Wang, 1998):

$$\text{maximize } y_n - y_1 \tag{7}$$

$$\text{subject to } y_j - y_i \leqslant c_{ij}, \quad i \neq j, \quad i,j = 1,2,\ldots,n, \tag{8}$$

where y_i denotes the dual decision variable associated with vertex i and $y_i - y_1$ is the shortest distance from vertex 1 to vertex i at optimality. Note that the value of the objective function at its maximum is the total cost of the shortest path (Bazaraa et al., 1990).

Although the last component of the optimal dual solution gives the total cost of the shortest path, the optimal dual solution needs decoding to the optimal primal solution in terms of edges. According to the Complementary Slackness Theorem (Bazaraa et al., 1990): given the feasible solutions of x_{ij} and y_i to the primal and dual problems, respectively, the solutions are optimal if and only if

(1) $x_{ij} = 1$ implies $y_j - y_i = c_{ij}$, $i,j = 1, 2, \ldots, n$,
(2) $x_{ij} = 0$ implies $y_j - y_i \leqslant c_{ij}$, $i,j = 1, 2, \ldots, n$,

From the above analysis, it is seen that the shortest path problem formulation based on the edge path representation, (4)–(6) with dual (7) and (8), is a LP problem. Thus, we consider the general form of LP problem as

$$\text{minimize } D^T x \tag{9}$$

$$\text{subject to } Ax - b \leqslant 0, \tag{10}$$

$$Ex - f = 0, \tag{11}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $E \in \mathbb{R}^{l\times n}$, $f \in \mathbb{R}^l$, $x \in \mathbb{R}^n$ and rank $(A,E) = m + l(0 \leqslant m, l < n)$. Throughout this paper, we assume that LP (9)–(11) has a unique optimal solution. In the next section, we will try to propose an efficient neural network model for solving LP problem (9)–(11) and discuss its architecture.

3. A neural network model

In this section, using standard optimization techniques, we transform (9)–(11) into a dynamic system. First, we introduce some notation, definitions, two lemmas and two theorems. Throughout this paper, $\mathbb{R}^n$ denotes the space of n-dimensional real column vectors and ^{T} denotes the transpose. In what follows, $\|.\|$ denotes the l^2-norm of $\mathbb{R}^n$ and $x = (x_1, x_2, \ldots, x_n)^T$. For any differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, $\nabla f(x) \in \mathbb{R}^n$ means the gradient of f at x. For any differentiable mapping $F = (F_1, \ldots, F_m)^T : \mathbb{R}^n \to \mathbb{R}^m$, $\nabla F = [\nabla F_1(x), \ldots, \nabla F_m(x)] \in \mathbb{R}^{n+m}$, denotes the transposed Jacobian of F at x.

Definition 3.1. A function $F : \mathbb{R}^n \to \mathbb{R}^n$ is said to be Lipschitz continuous with constant L on $\mathbb{R}^n$ if for each pair of points $x, y \in \mathbb{R}^n$,

$$\|F(x) - F(y)\| \leqslant L\|x - y\|.$$

F is said to be locally Lipschitz continuous on $\mathbb{R}^n$ if each point of $\mathbb{R}^n$ has a neighborhood $D_0 \subset \mathbb{R}^n$ such that the above inequality holds for each pair of points $x, y \in D_0$.

Definition 3.2. A mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is monotone if

$$(x-y)^T(F(x)-F(y)) \geqslant 0, \quad \forall x, y \in \mathbb{R}^n.$$

F is said to be strictly monotone if the strict inequality holds whenever $x \neq y$.

Lemma 3.3 (Ortega and Rheinboldt, 1970). *If a mapping F is continuously differentiable on an open convex set D including Ω, then F is monotone (strictly monotone) on Ω if and only if its Jacobian matrix $\nabla F(x)$ is positive semidefinite (positive definite) for all $x \in \Omega$.*

Definition 3.4. Let $x(t)$ be a solution trajectory of a system $x' = F(x)$, and let X^* denotes the set of equilibrium points of this equation. The solution trajectory of the system is said to be globally convergent to the set X^* if $x(t)$ satisfies

$$\lim_{t\to\infty} \operatorname{dist}(x(t), X^*) = 0,$$

where $\operatorname{dist}(x(t), X^*) = \inf_{y \in X^*} \|x - y\|$. In particular, if the set X^* has only one point x^*, then $\lim_{t\to\infty} x(t) = x^*$, and the system is said to be globally asymptotically stable at x^* if the system is also stable at x^* in the sense of Lyapunov.

Lemma 3.5 (Quarteroni et al., 2007). *If $A \in \mathbb{R}^{m\times n}$ is of full rank, then A^TA is a symmetric positive definite matrix.*

Theorem 3.6 (Bazaraa et al., 1990). *$x^* \in \mathbb{R}^n$ is an optimal solution of (9)–(11) if and only if there exist $u^* \in \mathbb{R}^m$ and $v^* \in \mathbb{R}^l$ such that $(x^{*T}, u^{*T}, v^{*T})^T$ satisfies the following KKT system*

$$\begin{cases} u^* \geqslant 0, \quad Ax^* - b \leqslant 0, \quad u^{*T}(Ax^* - b) = 0, \\ D + A^T u^* + E^T v^* = 0, \\ Ex^* - f = 0. \end{cases} \tag{12}$$

x^* is called a KKT point of (9)–(11) and a pair $(u^{*T}, v^{*T})^T$ is called the Lagrangian multiplier vector corresponding to x^*.

Theorem 3.7 (Bazaraa et al., 1990). *x^* is an optimal solution of (9)–(11) if and only if x^* is a KKT point of (9)–(11).*

Now, let $x(.)$, $u(.)$ and $v(.)$ be some time dependent variables. The aim is to construct a continuous-time dynamical system that will settle down to the KKT pair of the LP problem (9)–(11) and its dual. We propose a neural network for solving (9)–(11) and its dual, whose dynamical equation is defined as follows:

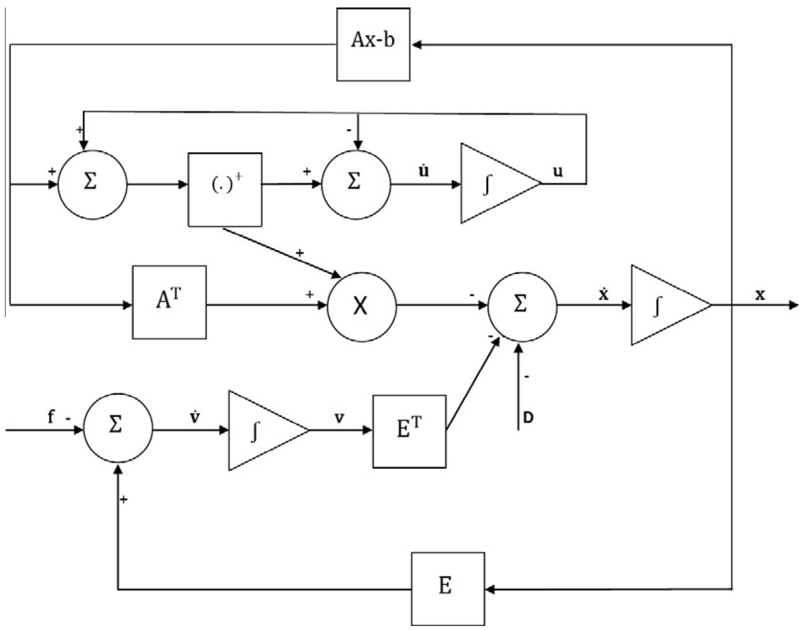

Fig. 1. A simplified block diagram for the neural network (17) and (18).

$$\frac{dx}{dt} = -(D + A^T(u + Ax - b)^+ + E^T\upsilon), \tag{13}$$

$$\frac{du}{dt} = (u + Ax - b)^+ - u, \tag{14}$$

$$\frac{d\upsilon}{dt} = Ex - f, \tag{15}$$

with the initial point $(x_0^T, u_0^T, \upsilon_0^T)^T$, where

$$(u + Ax - b)^+ = \left([(u + Ax - b)_{1.}]^+, [(u + Ax - b)_{2.}]^+, \ldots, [(u + Ax - b)_{m.}]^+\right),$$

$$[(u + Ax - b)_{k.}]^+ = \max\{(u + Ax - b)_{k.}, 0\}, \quad k = 1, 2, \ldots, m.$$

To simply the discussion, we define $y = (x^T, u^T, \upsilon^T)^T \in \mathbb{R}^{n+m+l}$, D^* as the optimal point set of (9)–(11) and its dual and define

$$\eta(y) = \begin{bmatrix} -(D + A^T(u + Ax - b)^+ + E^T\upsilon) \\ (u + Ax - b)^+ - u \\ Ex - f \end{bmatrix}. \tag{16}$$

Thus neural network 13, 21, 15 can be written in the following form

$$\frac{dy}{dt} = \eta(y), \tag{17}$$

$$y(t_0) = y_0. \tag{18}$$

An indication on how the neural network (17) and (18) can be implemented on hardware is provided in Fig. 1.

4. Stability analysis

In this section, we will study some stability and convergence properties for (17) and (18).

Theorem 4.1. *Let $y^* = (x^{*T}, u^{*T}, \upsilon^{*T})^T$ be the equilibrium point of the neural network (17) and (18). Then x^* is a KKT point of the problem (9)–(11) and its dual. On the other hand, if $x^* \in \mathbb{R}^n$ is an optimal solution of problem (9)–(11), then there exist $u^* \in \mathbb{R}^m$ and $\upsilon^* \in \mathbb{R}^l$ such that $y^* = (x^{*T}, u^{*T}, \upsilon^{*T})^T$ is an equilibrium point of the proposed neural network (17) and (18).*

Proof. Assume $y^* = (x^{*T}, u^{*T}, \upsilon^{*T})^T$ to be the equilibrium of the neural network (17) and (18). Then $\frac{dx^*}{dt} = 0$, $fracdu^*dt = 0$ and $\frac{d\upsilon^*}{dt} = 0$. It follows easily that

$$D + A^T(u^* + Ax^* - b)^+ + E^T\upsilon^* = 0, \tag{19}$$

$$(u^* + Ax^* - b)^+ - u^* = 0, \tag{20}$$

$$Ex^* - f = 0. \tag{21}$$

It is clear that $(u^* + Ax^* - b)^+ = u^*$ if and only if

$$u^* \geqslant 0, \quad Ax^* - b \leqslant 0, \quad u^{*T}(Ax^* - b) = 0. \tag{22}$$

Moreover, substituting (20) into (19) we have

$$D + A^T u^* + E^T\upsilon^* = 0. \tag{23}$$

From (21)–(23), it is seen that $y^* = (x^{*T}, u^{*T}, \upsilon^{*T})^T$ satisfies the KKT condition (12). The converse is straightforward. □

Lemma 4.2. *For any initial point $y(t_0) = (x(t_0)^T, u(t_0)^T, \upsilon(t_0)^T)^T$, there exists a unique continuous solution $y(t) = (x(t)^T, u(t)^T, \upsilon(t)^T)^T$ for system (17) and (18).*

Proof. It is easy to verify that $D + A^T(u + Ax - b)^+ + E^T\upsilon$, $(u + Ax - b)^+ - u$ and $Ex - f$ are locally Lipschitz continuous on an open convex set $D \subseteq \mathbb{R}^{n+m+l}$. According to the local existence of ordinary differential equations (Miller and Michel, 1982), neural network (17) and (18) has a unique continuous solution $y(t)$, $t \in [t_0, \tau)$ for some $\tau > t_0$, as $\tau \to \infty$. □

Lemma 4.3. *Let $A \in \mathbb{R}^{m\times n}$ be of full rank. Then the Jacobian matrix $\nabla\eta(y)$ of the mapping η defined in (16) is a negative semidefinite matrix.*

Proof. Without loss of generality, assume that there exists $0 < p < m$ such that

$$(u + Ax - b)^+ = ((u + Ax - b)_{1.}, (u + Ax - b)_{2.}, \ldots, (u + Ax - b)_{p.}, \underbrace{0, 0, \ldots, 0}_{m-p}).$$

With a simple calculation, it is clearly shown that

$$\nabla\eta(y) = \begin{pmatrix} -(A^p)^T A^p & -(A^p)^T & -E^T \\ A^p & S_{m\times m} & O_{m\times l} \\ E & O_{l\times m} & O_{l\times l} \end{pmatrix},$$

where O indicates a zero matrix,

$$A^p = \begin{pmatrix} U_{p\times n} \\ O_{(m-p)\times n} \end{pmatrix} = \begin{pmatrix} A_{1.} \\ A_{2.} \\ \cdots \\ \cdots \\ A_{p.} \\ O_{1\times n} \\ O_{1\times n} \\ \cdots \\ \cdots \\ O_{1\times n} \end{pmatrix},$$

and

$$S_{m\times m} = \begin{pmatrix} O_{p\times p} & O_{p\times(m-p)} \\ O_{(m-p)\times p} & -I_{(m-p)\times(m-p)} \end{pmatrix}.$$

By the assumption that A is of full rank, the first p rows of matrix A^p are linearly independent, i.e. matrix U is of full rank. Using Lemma 3.5 and the fact that $(A^p)^T A^p = U^T U$, we see that $(A^p)^T A^p$ is a positive definite matrix. Moreover, it is clear that matrix $S_{m\times m}$ is negative semidefinite matrix. From the stated observations, we can derive that the Jacobian matrix $\nabla\eta(y)$ is a negative semidefinite matrix.

If $p = m$, i.e. $(u + Ax - b)^+ = ((u + Ax - b)_{1.}, (u + Ax - b)_{2.}, \ldots, (u + Ax - b)_{m.})$, then

$$\nabla\eta(y) = \begin{pmatrix} -A^T A & -A^T & -E^T \\ A & O_{m\times m} & O_{m\times l} \\ E & O_{l\times m} & O_{l\times l} \end{pmatrix}.$$

Again, utilizing Lemma 3.5 and the fact that A is of full rank, it is easily proved that $\nabla\eta(y)$ is a negative semidefinite matrix.

Finally, if $p = 0$, i.e. $(u + Ax - b)^+ = (\underbrace{0, 0, \ldots, 0}_{m})$, then we obtain

$$\nabla\eta(y) = \begin{pmatrix} O_{n\times n} & O_{n\times m} & -E^T \\ O_{m\times n} & -I_{m\times m} & O_{m\times l} \\ E & O_{l\times m} & O_{l\times l} \end{pmatrix}.$$

In this case also it is easy to verify that $\nabla\eta(y)$ is a negative semidefinite matrix. This completes the proof. □

Theorem 4.4. *Let the assumption of Lemma 4.3 be satisfied. Then the proposed neural network model in (17) and (18) is globally stable in the Lyapunov sense and is globally convergent to* $y^* = (x^{*T}, u^{*T}, v^{*T})^T$*, where* x^* *is the optimal solution of (9)–(11).*

Proof. Consider the Lyapunov function $V : \mathbb{R}^{m+n+l} \to \mathbb{R}$ as follows

$$V(y) = \|\eta(y)\|^2 + \frac{1}{2}\|y - y^*\|^2. \tag{24}$$

From (16), it is seen that

$$\frac{d\eta}{dt} = \frac{\partial\eta}{\partial y}\frac{dy}{dt} = \nabla\eta(y)\eta(y).$$

Calculating the derivative of $V(t)$ along the solution $y(t)$ of the neural network (17) and (18), we have

$$\frac{dV(y(t))}{dt}=\left(\frac{d\eta}{dt}\right)^T\eta+\eta^T\left(\frac{d\eta}{dt}\right)+(y-y^*)^T\frac{dy(t)}{dt}=\eta^T(\nabla\eta(y)^T+\nabla\eta(y))\eta+(y-y^*)^T\eta(y). \tag{25}$$

Employing Lemma 4.3, we attain

$$\eta^T(y)(\nabla\eta(y)^T+\nabla\eta(y))\eta(y)\leqslant 0,\quad \forall y\neq y^*. \tag{26}$$

Moreover, from Definiton 3.2 and Lemma 3.3, we have

$$(y-y^*)^T(\eta(y)-\eta(y^*))=(y-y^*)^T\eta(y)\leqslant 0,\quad \forall y\neq y^*.$$

Thus

$$\frac{dV(y(t))}{dt}\leqslant 0. \tag{27}$$

This means that the neural network (17) and (18) is globally stable in the sense of Lyapunov. Next, since

$$V(y)\geqslant\frac{1}{2}\|y-y^*\|^2, \tag{28}$$

there exists a convergent subsequence

$$\{\left(x(t_k)^T,u(t_k)^T,v(t_k)^T\right)^T|t_0<t_1<\cdots<t_k<t_{k+1}\},\quad\text{and}\quad t_k\to\infty\text{ as }k\to\infty,$$

such that $\lim_{k\to\infty}(x(t_k)^T,u(t_k)^T,v(t_k)^T)^T=(\bar{x}^T,\bar{u}^T,\bar{v}^T)^T$, where $(\bar{x}^T,\bar{u}^T,\bar{v}^T)^T$ satisfies

$$\frac{dV(y(t))}{dt}=0,$$

which indicates that $(\bar{x}^T,\bar{u}^T,\bar{v}^T)^T$ is an ω-limit point of $\{(x(t)^T,u(t)^T,v(t)^T)^T|t\geqslant t_0\}$. Using the LaSalle invariant set theorem (Vegas and Zufiria, 2004), one has that $\{(x(t)^T,u(t)^T,v(t)^T)^T\to M\}$ as $t\to\infty$, where M is the largest invariant set in $K=\{(x(t)^T,u(t)^T,v(t)^T)^T|\frac{dV(y(t))}{dt}=0\}$. From 17 and 18 and 27 it follows that $\frac{dx}{dt}=0$, $\frac{du}{dt}=0$ and $\frac{dv}{dt}=0\iff\frac{dV(y(t))}{dt}=0$. Thus $(\bar{x}^T,\bar{u}^T,\bar{v}^T)^T\in D^*$ by $M\subseteq K\subseteq D^*$.

Substituting $x^*=\bar{x}$, $u^*=\bar{u}$ and $v^*=\bar{v}$ in (24), we define another Lyapunov function

$$\overline{V}(y)=\|\eta(y)\|^2+\frac{1}{2}\|y-\bar{y}\|^2. \tag{29}$$

Then $\overline{V}(y)$ is continuously differentiable and $\overline{V}(\bar{y})=0$. Noting that

$$\lim_{k\to\infty}(x(t_k)^T,u(t_k)^T,v(t_k)^T)^T=(\bar{x}^T,\bar{u}^T,\bar{v}^T)^T,$$

we therefore have $\lim_{k\to\infty}\overline{V}(x(t_k)^T,u(t_k)^T,v(t_k)^T)^T=\overline{V}(\bar{x},\bar{u},\bar{v})$.

So, $\forall\epsilon>0$ there exists $q>0$ such that for all $t\geqslant t_q$, we have $\overline{V}(y(t))<\epsilon$. Similarly, we can obtain $\frac{d\overline{V}(y(t))}{dt}\leqslant 0$. It follows that for $t\geqslant t_q$,

$$\frac{1}{2}\|y(t)-\bar{y}\|^2\leqslant\overline{V}(y(t))\leqslant\epsilon.$$

It follows that $\lim_{t\to\infty}\|y(t)-\bar{y}\|=0$ and $\lim_{t\to\infty}y(t)=\bar{y}$. Therefore, the proposed neural network in (17) and (18) is globally convergent to an equilibrium point $\bar{y}=(\bar{x}^T,\bar{u}^T,\bar{v}^T)^T$, where $\bar{x}$ is the optimal solution of (9)–(11). □

As an immediate corollary of Theorem 4.4, we can get the following result.

Corollary 1. *Assume that the hypothesis of Lemma 4.3 is satisfied. If $D^*=\{(x^{*T},u^{*T},v^{*T})^T\}$, then the neural network (17) and (18) for solving (9)–(11) is globally asymptotically stable to the unique equilibrium point $y^*=(x^{*T},u^{*T},v^{*T})^T$.*

5. Simulation experiments

In order to demonstrate the effectiveness of the proposed neural network, in this section, we test 7 examples by the neural network (17) and (18). The simulation is conducted on Matlab 7, the ordinary differential equation solver engaged is ode45s.

Example 5.1.

$$\text{minimize} \ -x_1 - x_2$$
$$\text{subject to} \begin{cases} \frac{5}{12}x_1 - x_2 \leqslant \frac{35}{12}, \\ \frac{5}{2}x_1 + x_2 \leqslant \frac{35}{2}, \\ -x_1 \leqslant 5, \\ x_2 \leqslant 5. \end{cases}$$

The optimal solution to this problem is $x^* = (5,5)^T$. We apply the proposed neural network in (17) and (18) to solve this LP problem. Simulation results show the trajectory of (17) and (18) with any initial point is always convergent to $y^* = (x^{*T}, u^{*T})^T$. For example, Fig. 2 displays the transient behavior of $x(t) = (x_1(t),x_2(t))^T$ based on (17) and (18) with 50 random initial points. It is easy to verify that whether or not an initial point is taken inside or outside the feasible region, the proposed network always converges to the theoretical optimal solution x^*.

To see how well the proposed network model (17) and (18) is, we compare it with the Friesz et al. dynamic system in Friesz et al. (1994), given by

$$\frac{dx}{dt} = -(x - P_\Omega(x - D - A^T u)), \tag{30}$$
$$\frac{du}{dt} = -\left(u - (u + Ax - b)^+\right), \tag{31}$$

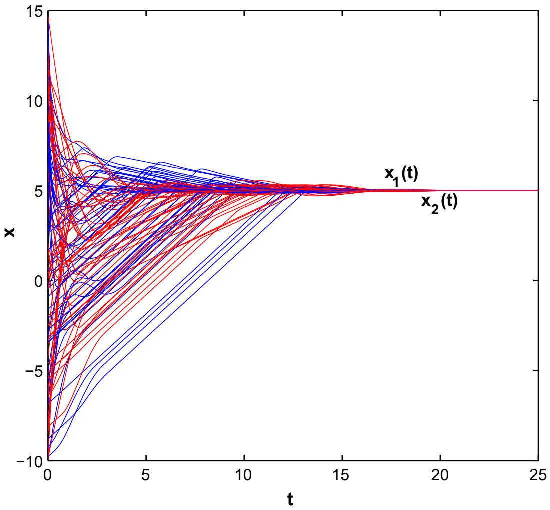

Fig. 2. Transient behaviors of $x(t)$ of the neural network (17) and (18) with 50 various initial points in Example 5.1.

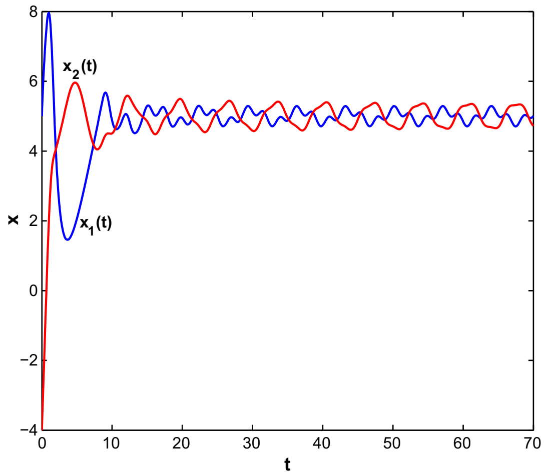

Fig. 3. Divergent behavior of the system (30) and (31) with the initial point $y_0 = (5,-4,3,-2,1,0)^T$ in Example 5.1.

where $\Omega \subset \mathbb{R}^n$ is a closed convex set and $P_\Omega : \mathbb{R}^n \to \Omega$ is a projection operator (Xia and Wang, 2004a) defined by

$$P_\Omega(x) = \mathrm{argmin}_{v\in\Omega}\|x - v\|.$$

Fig. 3 shows that this model with the initial point $y_0 = (5, -4, 3, -2, 1, 0)^T$ is not stable.

Moreover, the above problem is solved by using Kennedy and Chua's neural network in Maa and Shanblatt (1992), given by

$$\frac{dx}{dt} = -(D + \gamma(A^T(Ax - b)^+)), \tag{32}$$

where γ is a penalty parameter. It is obtained that this network is not capable to find an exact optimal solution due to a finite penalty parameter and is difficult to implement when the penalty parameter is very large (Lillo et al., 1993). Thus, this network only converges an approximate optimal solution of the problem for any given finite penalty parameter.

Example 5.2.

$$\text{minimize } 2x_{12} - x_{13} - 4x_{23} + 3x_{24} - 6x_{34}$$
$$\text{subject to } \begin{cases} x_{12} + x_{13} = 1, \\ x_{23} + x_{24} - x_{12} = 0, \\ x_{34} - x_{13} - x_{23} = 0, \\ -x_{24} - x_{34} = -1. \end{cases}$$

For simplicity, let assume $x_1 = x_{12}$, $x_2 = x_{13}$, $x_3 = x_{23}$, $x_4 = x_{24}$, $x_5 = x_{34}$. According to the simplified edge path representation, the equivalent LP problem can be described as follows:

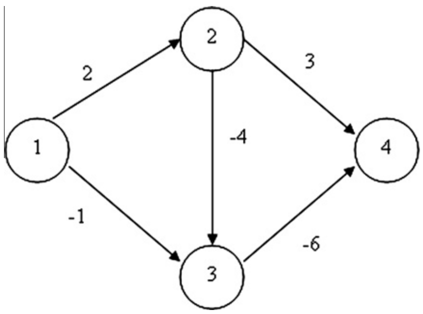

Fig. 4. The graph for Example 5.2.

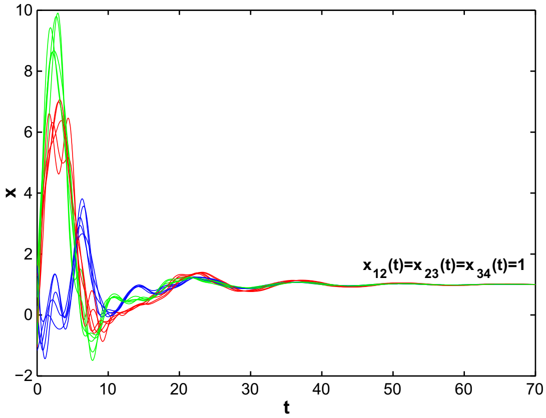

Fig. 5. Transient behaviors of x_{12}, x_{23} and x_{34} of the neural network (17) and (18) with 5 various initial points in Example 5.2.

$$\text{minimize } 2x_1 - x_2 - 4x_3 + 3x_4 - 6x_5$$
$$\text{subject to } \begin{cases} x_1 + x_2 = 1, \\ -x_1 + x_3 + x_4 = 0, \\ -x_2 - x_3 + x_5 = 0, \\ -x_4 - x_5 = -1, \\ x \geqslant 0. \end{cases}$$

The shortest path in the graph of Fig. 4 is: $1 \to 2 \to 3 \to 4$. All simulation results show that the state trajectory of the proposed model converges to the unique optimal solution $x^*_{12} = x^*_{23} = x^*_{34} = 1$ and $x^*_{13} = x^*_{24} = 0$. Figs. 5 and 6 show that the trajectories of the neural network model (17) and (18) with 5 random initial points converge to the optimal solution of this example. An l_2 norm error between x and x^* is also shown in Fig. 7.

To make a comparison, the above problem is solved by using the dynamic system in Friesz et al., 1994. Figs. 8 and 9 show that this model with the initial point $y_0 = (-7, 6, -5, 4, -3, 2, -1, 0, 1, -2, 3, -4, 5, -6)^T$ is not stable.

Example 5.3.

$$\text{minimize } 4x_{41} + 2x_{12} - 3x_{24} + 6x_{31} - 7x_{32} + 3x_{34} + 5x_{42}$$
$$\text{subject to } \begin{cases} x_{12} - x_{31} - x_{41} = 1, \\ x_{24} - x_{42} - x_{12} - x_{32} = 0, \\ x_{32} + x_{31} + x_{34} = 0, \\ x_{42} + x_{41} - x_{24} - x_{34} = -1. \end{cases}$$

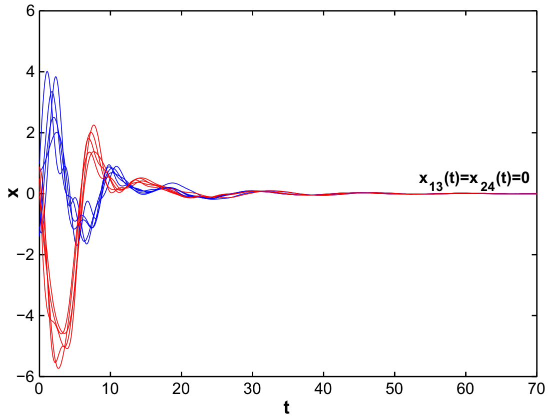

Fig. 6. Transient behaviors of x_{13} and x_{24} of the neural network (17) and (18) with 5 various initial points in Example 5.2.

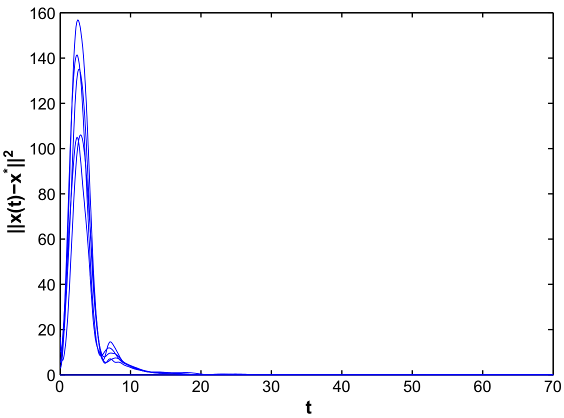

Fig. 7. The convergence behavior of $\|x(t) - x^*\|^2$ in Example 5.2.

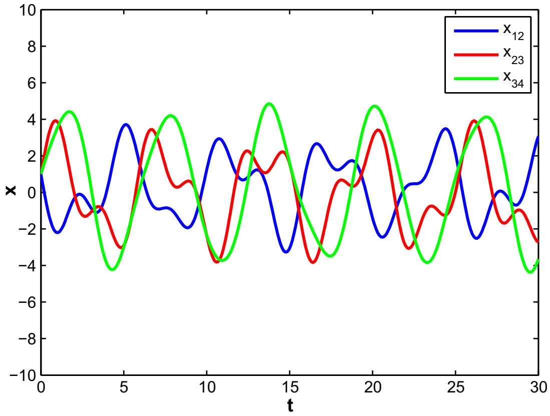

Fig. 8. Divergent behaviors of x_{12}, x_{23} and x_{34} of the neural network (17) and (18) with $y_0 = (-7, 6, -5, 4, -3, 2, -1, 0, 1, -2, 3, -4, 5, -6)^T$ in Example 5.2.

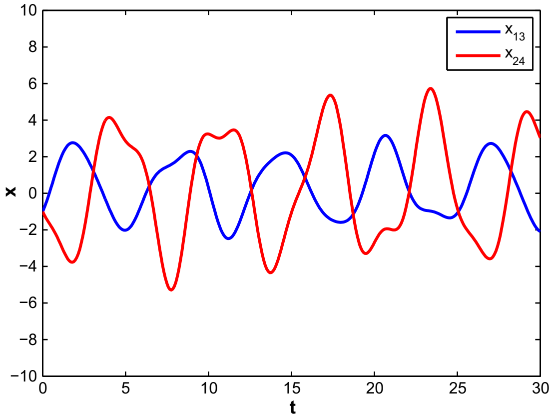

Fig. 9. Divergent behaviors of x_{13} and x_{24} of the neural network (17) and (18) with $y_0 = (-7, 6, -5, 4, -3, 2, -1, 0, 1, -2, 3, -4, 5, -6)^T$ in Example 5.2.

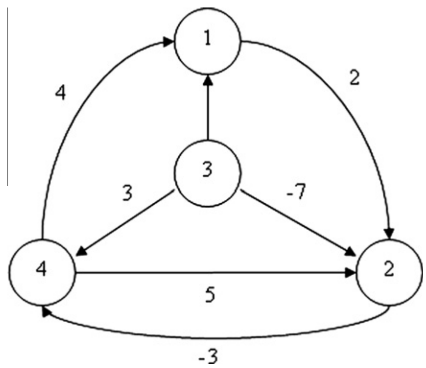

Fig. 10. The graph for Example 5.3.

The optimal solution for this problem is

$$x^*_{41} = x^*_{31} = x^*_{32} = x^*_{34} = x^*_{42} = 0,$$
$$x^*_{12} = x^*_{24} = 1.$$

Thus, the shortest path in the graph of Fig. 10 is: $1 \to 2 \to 4$. Figs. 11–13 display the transient behavior of the proposed network with the initial point

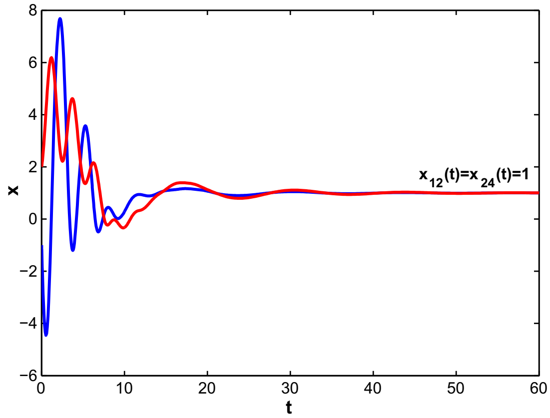

Fig. 11. Transient behaviors of x_{12} and x_{24} of the neural network (17) and (18) with $x_0 = (1, -1, 2, -2, 3, -3, 4)^T$ in Example 5.3.

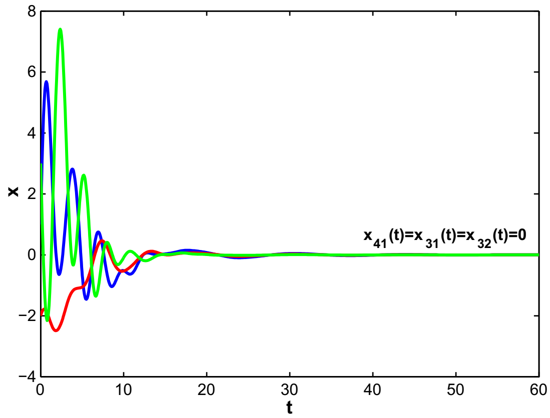

Fig. 12. Transient behaviors of x_{41}, x_{31} and x_{32} of the neural network (17) and (18) with $x_0 = (1, -1, 2, -2, 3, -3, 4)^T$ in Example 5.3.

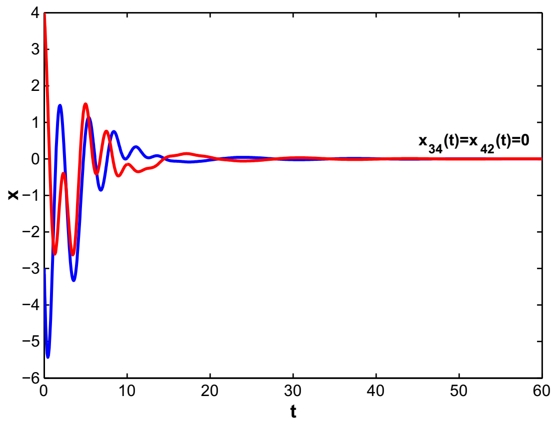

Fig. 13. Transient behaviors of x_{34} and x_{42} of the neural network (17) and (18) with $x_0 = (1, -1, 2, -2, 3, -3, 4)^T$ in Example 5.3.

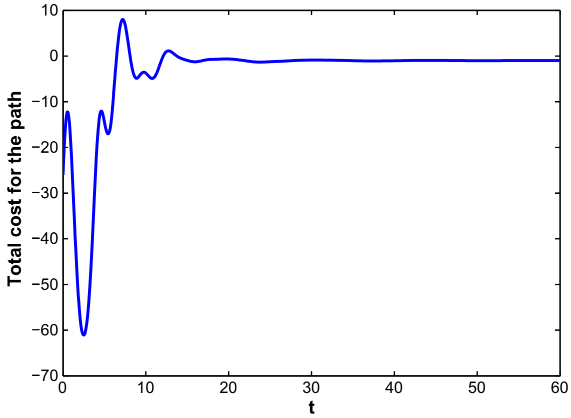

Fig. 14. The total cost for the path with $x_0 = (1,-1,2,-2,3,-3,4)^T$ in Example 5.3.

$$x_0 = (x_{41}(0), x_{12}(0), x_{24}(0), x_{31}(0), x_{32}(0), x_{34}(0), x_{42}(0))^T = (1,-1,2,-2,3,-3,4)^T.$$

The total cost for the path, that is -1, is also shown in Fig. 14.

Example 5.4.

$$\text{minimize } 3x_{12} + 9x_{13} + 12x_{15} + 2x_{24} + 9x_{27} + 8x_{25} + 4x_{35} + 13x_{38} + 12x_{36} + 5x_{47} + x_{57} + 16x_{59} + 6x_{58} + 2x_{68} + 15x_{79} + 4x_{89}$$

$$\text{subject to } \begin{cases} x_{12} + x_{13} + x_{15} = 1, \\ x_{24} + x_{27} + x_{25} - x_{12} = 0, \\ x_{35} + x_{38} + x_{36} - x_{13} = 0, \\ x_{47} - x_{24} = 0, \\ x_{57} + x_{59} + x_{58} - x_{15} - x_{25} - x_{35} = 0, \\ x_{68} - x_{36} = 0, \\ x_{79} - x_{57} - x_{27} - x_{47} = 0, \\ x_{89} - x_{38} - x_{58} - x_{38} = 0, \\ -x_{89} - x_{59} - x_{79} = -1. \end{cases}$$

The optimal solution for this problem is

$$x^*_{12} = x^*_{25} = x^*_{58} = x^*_{89} = 1,$$
$$x^*_{13} = x^*_{15} = x^*_{24} = x^*_{27} = x^*_{35} = x^*_{38} = x^*_{36} = x^*_{47} = x^*_{57} = x^*_{59} = x^*_{68} = x^*_{79} = 0.$$

Thus, the shortest path in the graph of Fig. 15 is: $1 \to 2 \to 5 \to 8 \to 9$. An l_2 norm error between x and x^* with 10 different initial points is shown in Fig. 16.

Example 5.5.

$$\text{minimize } 3x_{12} + 5x_{13} + 3x_{14} + 7x_{25} + x_{23} + x_{34} + 3x_{35} + 2x_{42} + 5x_{45}$$

$$\text{subject to } \begin{cases} x_{12} + x_{13} + x_{14} = 1, \\ x_{23} + x_{25} - x_{42} - x_{12} = 0, \\ x_{34} + x_{35} - x_{13} - x_{23} = 0, \\ x_{42} + x_{45} - x_{34} - x_{14} = 0, \\ -x_{45} - x_{25} - x_{35} = -1. \end{cases}$$

The optimal solution for this problem is

$$x^*_{12} = x^*_{23} = x^*_{35} = 1,$$
$$x^*_{13} = x^*_{14} = x^*_{25} = x^*_{34} = x^*_{42} = x^*_{45} = 0.$$

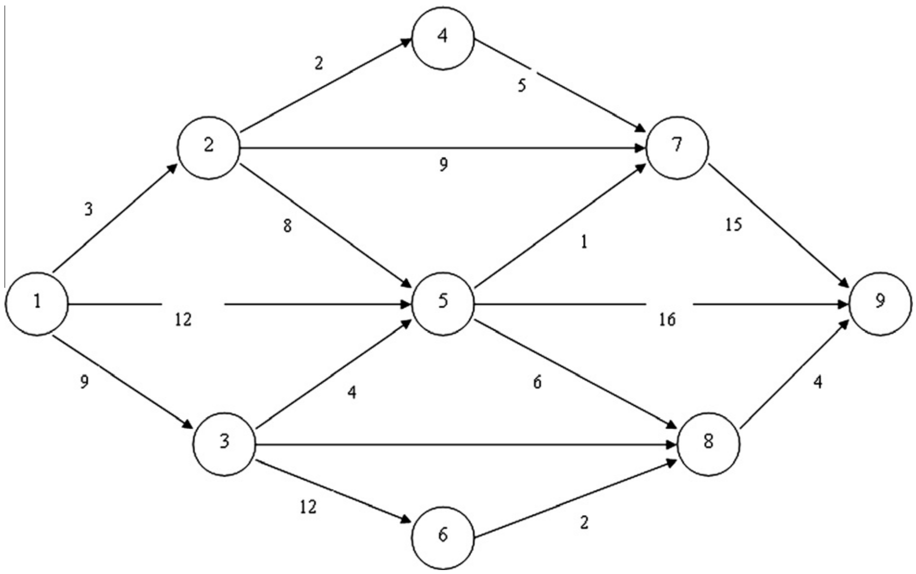

Fig. 15. The graph for Example 5.4.

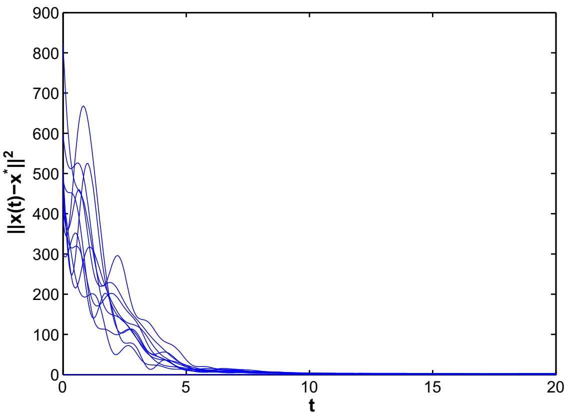

Fig. 16. The convergence behavior of $\|x(t) - x^*\|^2$ in Example 5.4.

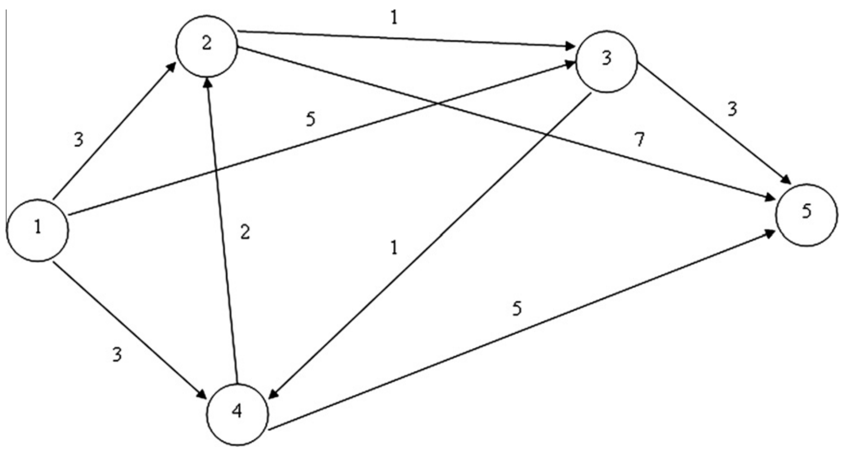

Fig. 17. The graph for Example 5.5

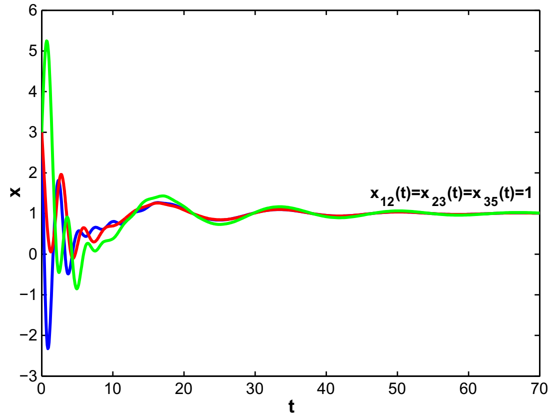

Fig. 18. Transient behaviors of x_{12}, x_{23} and x_{35} of the neural network (17) and (18) with $x_0 = (3, -3, 3, -3, 3, -3, 3, -3, 3)^T$ in Example 5.5.

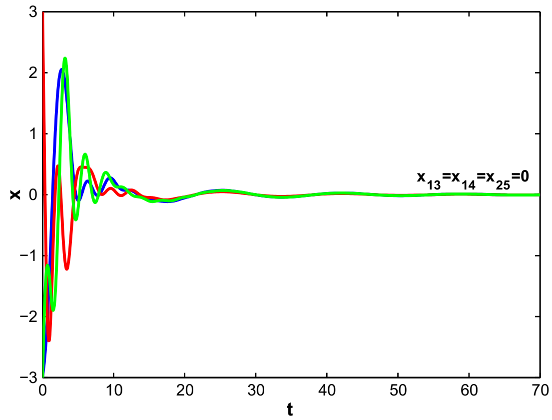

Fig. 19. Transient behaviors of x_{13}, x_{14} and x_{25} of the neural network (17) and (18) with $x_0 = (3, -3, 3, -3, 3, -3, 3, -3, 3)^T$ in Example 5.5.

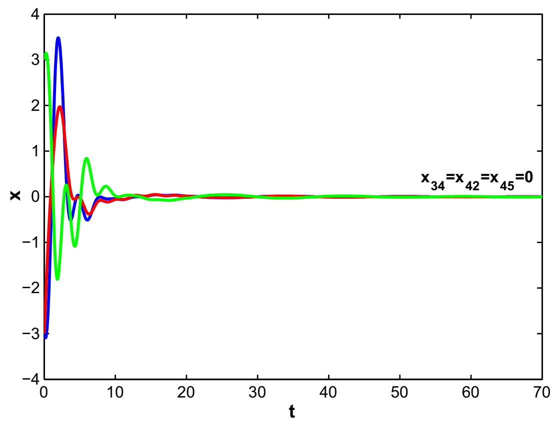

Fig. 20. Transient behaviors of x_{34}, x_{42} and x_{45} of the neural network (17) and (18) with $x_0 = (3, -3, 3, -3, 3, -3, 3, -3, 3)^T$ in Example 5.5.

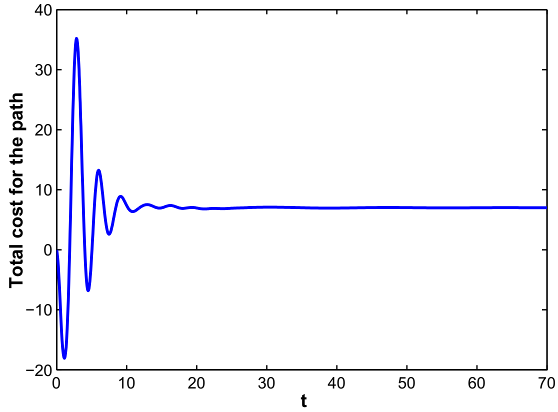

Fig. 21. The total cost for the path with $x_0 = (3,-3,3,-3,3,-3,3,-3,3)^T$ in Example 5.5.

Thus, the shortest path in the graph of Fig. 17 is: $1 \to 2 \to 3 \to 5$. Figs. 18–20 display the transient behavior of the proposed network with the initial point

$$x_0 = (x_{12}(0), x_{13}(0), x_{14}(0), x_{25}(0), x_{23}(0), x_{34}(0), x_{35}(0), x_{42}(0), x_{45}(0))^T = (3,-3,3,-3,3,-3,3,-3,3)^T.$$

The total cost for the path, that is 7, is also shown in Fig. 21.

Example 5.6.

$$\text{minimize } 7x_{12} + 4x_{13} + 6x_{14} + x_{15} + 2x_{32} + 5x_{34} + 3x_{42} + x_{54}$$

$$\text{subject to } \begin{cases} x_{12} + x_{13} + x_{14} + x_{15} = 1, \\ -x_{12} - x_{32} - x_{42} = 0, \\ x_{32} + x_{34} - x_{13} = 0, \\ x_{42} - x_{34} - x_{14} - x_{54} = 0, \\ x_{54} - x_{15} = -1. \end{cases}$$

The optimal solution for this problem is

$$x^*_{15} = 1,$$
$$x^*_{12} = x^*_{13} = x^*_{14} = x^*_{32} = x^*_{34} = x^*_{42} = x^*_{54} = 0.$$

Thus, the shortest path in the graph of Fig. 22 is: $1 \to 5$. Figs. 23–25 display the transient behavior of the proposed network with the initial point

$$x_0 = (x_{12}(0), x_{13}(0), x_{14}(0), x_{15}(0), x_{32}(0), x_{34}(0), x_{42}(0), x_{54}(0))^T = (-2,2,-2,2,-2,2,-2,2)^T.$$

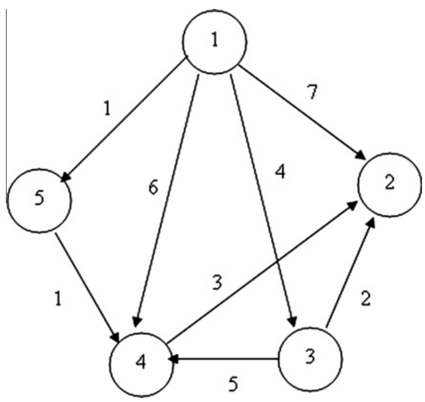

Fig. 22. The graph for Example 5.6.

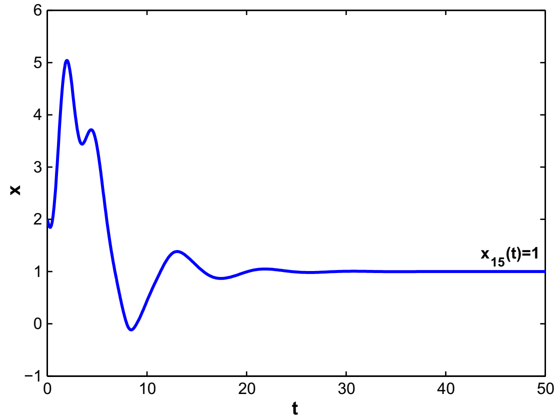

Fig. 23. Transient behaviors of x_{15} with $x_0 = (-2,2,-2,2,-2,2,-2,2)^T$ in Example 5.6.

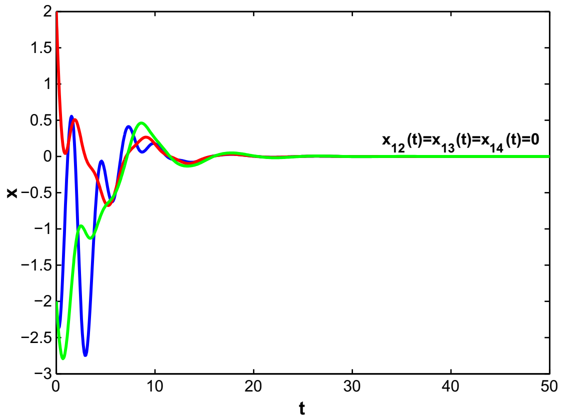

Fig. 24. Transient behaviors of x_{12}, x_{13} and x_{14} with $x_0 = (-2,2,-2,2,-2,2,-2,2)^T$ in Example 5.6.

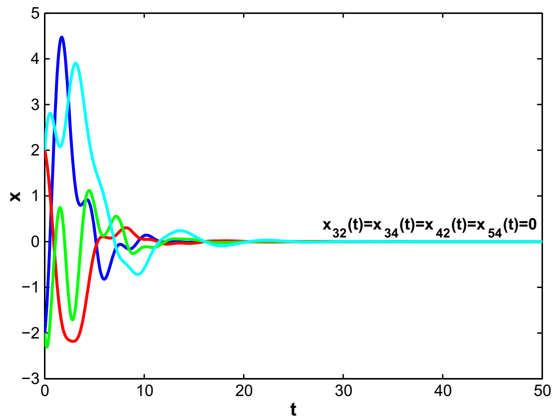

Fig. 25. Transient behaviors of x_{32}, x_{34}, x_{42} and x_{54} with $x_0 = (-2,2,-2,2,-2,2,-2,2)^T$ in Example 5.6.

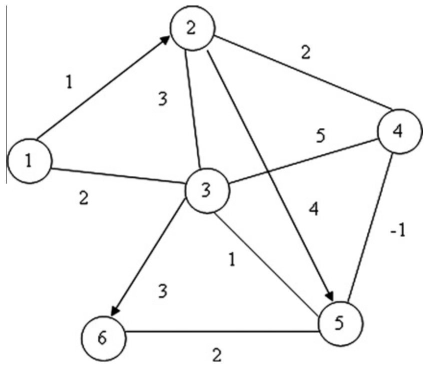

Fig. 26. The graph for Example 5.7.

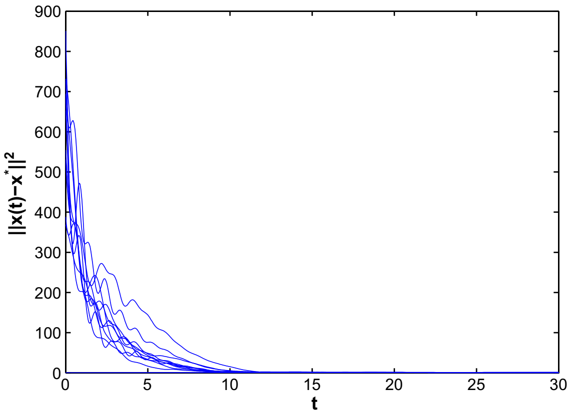

Fig. 27. The convergence behavior of $\|x(t) - x^*\|^2$ in Example 5.7.

Example 5.7.

$$\text{minimize } x_{12} + 2x_{13} + 3x_{23} + 3x_{32} + 2x_{31} + 4x_{25} + 2x_{24} + 2x_{42} + 5x_{34} + 5x_{43} - x_{45} - x_{54} + 2x_{56} + 2x_{65} + 3x_{36} + x_{35} + x_{53}$$

$$\text{subject to } \begin{cases} x_{12} + x_{13} - x_{31} = 1, \\ x_{23} + x_{25} + x_{24} - x_{12} - x_{32} - x_{42} = 0, \\ x_{31} + x_{32} + x_{34} + x_{35} + x_{36} - x_{13} - x_{23} - x_{43} - x_{53} = 0, \\ x_{42} + x_{43} + x_{45} - x_{24} - x_{34} - x_{54} = 0, \\ x_{54} + x_{53} + x_{56} - x_{45} - x_{35} - x_{65} - x_{25} = 0, \\ x_{65} - x_{56} - x_{36} = -1. \end{cases}$$

The optimal solution for this problem is

$$x^*_{13} = x_{36} = 1,$$
$$x^*_{12} = x^*_{23} = x^*_{32} = x^*_{31} = x^*_{25} = x^*_{24} = x^*_{42} = x^*_{34} = x^*_{43} = x^*_{45} = x^*_{54} = x^*_{56} = x^*_{65} = x^*_{35} = x^*_{53} = 0.$$

Thus, the shortest path in the graph of Fig. 26 is: $1 \to 3 \to 6$. An l_2 norm error between x and x^* with 10 different initial points is shown in Fig. 27.

6. Conclusion

In this paper, we have proposed a high-performance neural network model for solving the shortest path problem. Based on the Lyapunov stability theory and LaSalle invariance principle, we prove strictly the asymptotic stability of the proposed network. From any initial point, the trajectory of this network converges to an optimal solution of the original programming problem. The structure of the proposed network is reliable and efficient. The other advantages of the proposed neural network are that it can be implemented without a penalty parameter and can be convergent to an exact solution of the problem. The results obtained are highly valuable in both theory and practice for solving the shortest path problems in engineering.

Acknowledgments

The authors are grateful to the anonymous reviewer for his (her) suggestions in improving the quality of the paper.

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs, New Jersey.
Antonio, J.K., Huang, G.M., Tsai, W.K., 1992. A fast distributed shortest path algorithm for a class of hierarchically clustered data networks. IEEE Transactions on Computers 41 (6), 710–724.
Bazaraa, M.S., Jarvis, J.J., Sherali, H.D., 1990. Linear Programming and Network Flows. Wiley, New York.
Bodin, L., Golden, B.L., Assad, A., Ball, M., 1983. Routing and scheduling of vehicles and crews: the state of the art. Computers & Operations Research 10 (2), 63–211.
Ding, K., Huang, N.-J., 2008. A new class of interval projection neural networks for solving interval quadratic program. Chaos, Solitons and Fractals 35 (4), 718–725.
Effati, S., Ghomashi, A., Nazemi, A.R., 2007. Application of projection neural network in solving convex programming problems. Applied Mathematics and Computation 188 (2), 1103–1114.
Effati, S., Jafarzadeh, M., 2007. Nonlinear neural networks for solving the shortest path problem. Applied Mathematics and Computation 189 (1), 567–574.
Effati, S., Nazemi, A.R., 2006. Neural network models and its application for solving linear and quadratic programming problems. Applied Mathematics and Computation 172 (1), 305–331.
Ephremides, A., Verdu, S., 1989. Control and optimization methods in communication network problems. IEEE Transactions on Automatic Control 34 (9), 930–942.
Friesz, T.L., Bernstein, D.H., Mehta, N.J., Tobin, R.L., Ganjlizadeh, S., 1994. Day-to-day dynamic network disequilibria and idealized traveler information systems. Operation Research 42 (6), 1120–1136.
Gao, X.B., 2004. A novel neural network for nonlinear convex programming. IEEE Transactions on Neural Networks 15 (3), 613–621.
Hu, X., 2009. Applications of the general projection neural network in solving extended linear-quadratic programming problems with linear constraints. Neurocomputing 72 (4–6), 1131–1137.
Hu, X., Wang, J., 2007. Design of general projection neural networks for solving monotone linear variational inequalities and linear and quadratic optimization problems. IEEE Transactions on Systems, Man, and Cybernetics, Part B 37 (5), 1414–1421.
Jun, S., Shin, K.G., 1991. Shortest path planning in distributed workspace using dominance relation. IEEE Transactions on Robotics and Automation 7 (3), 342–350.
Kennedy, M.P., Chua, L.O., 1988. Neural networks for nonlinear programming. IEEE Transactions on Circuits and Systems 35 (5), 554–562.
Lawler, E.L., 1976. Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston, New York.
Lillo, W.E., Loh, M.H., Hui, S., Zăk, S.H., 1993. On solving constrained optimization problems with neural networks: a penalty method approach. IEEE Transactions on Neural Networks 4 (6), 931–939.
Lin, P.L., Chang, S., 1993. A shortest path algorithm for a nonrotating object among obstacles of arbitrary shapes. IEEE Transactions on Systems, Man, and Cybernetics 23 (3), 825–833.
Maa, C.Y., Shanblatt, M.A., 1992. Linear and quadratic programming neural network analysis. IEEE Transactions on Neural Networks 3 (4), 580–594.
Miller, R.K., Michel, A.N., 1982. Ordinary Differential Equations. Academic Press, New York.
Nazemi, A.R., 2011a. A dynamic system model for solving convex nonlinear optimization problems. Communications in Nonlinear Science and Numerical Simulation 17 (4), 1696–1705.
Nazemi, A.R., 2011b. A dynamical model for solving degenerate quadratic minimax problems with constraints. Journal of Computational and Applied Mathematics 236 (6), 1282–1295.
Ortega, T.M., Rheinboldt, W.C., 1970. Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York.
Quarteroni, A., Sacco, R., Saleri, F., 2007. Numerical Mathematics. Springer-Verlag, Berlin.
Soueres, P., Laumond, J.-P., 1996. Shortest paths synthesis for a car-like robot. IEEE Transactions on Automatic Control 41 (5), 672–688.
Tank, D.W., Hopfield, J.J., 1986. Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming pircuit. IEEE Transactions on Circuits and Systems 33 (5), 533–541.
Tao, Q., Cao, J., Sun, D., 2001. A simple and high performance neural network for quadratic programming problems. Applied Mathematics and Computation 124 (2), 251–260.
Vegas, J.M., Zufiria, P.J., 2004. Generalized neural networks for spectral analysis: dynamics and Liapunov functions. Neural Networks 17 (2), 233–245.
Wang, J., 1996. A recurrent neural network for solving the shortest path problems. IEEE Transactions on Circuits and Systems – Part I: Fundamental Theory and Applications 43 (6), 482–486.
Wang, J., 1998. Primal and dual neural networks for shortest-path routing. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 28 (6), 864–869.
Xia, Y., 1996. A new neural network for solving linear and quadratic programming problems. IEEE Transactions on Neural Networks 7 (6), 1544–1547.
Xia, Y., Feng, G., Wang, J., 2004. A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equation. Neural Networks 17 (7), 1003–1015.
Xia, Y., Feng, G., 2005. An improved network for convex quadratic optimization with application to real-time beamforming. Neurocomputing 64, 359–374.
Xia, Y., Wang, J., 2000. A discrete-time recurrent neural network for shortest-path routing. IEEE Transactions on Automatic Control 45 (11), 2129–2134.
Xia, Y., Wang, J., 2004a. A general projection neural network for solving monotone variational inequality and related optimization problems. IEEE Transactions on Neural Networks 15 (2), 318–328.
Xia, Y., Wang, J., 2004b. A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. IEEE Transactions on Circuits and Systems 51 (7), 447–458.
Xue, X., Bian, W., 2007. A project neural network for solving degenerate convex quadratic program. Neurocomputing 70 (13–15), 2449–2459.
Xue, X., Bian, W., 2009. A project neural network for solving degenerate quadratic minimax problem with linear constraints. Neurocomputing 72 (7–9), 1826–1838.
Yang, Y., Cao, J., 2006a. Solving quadratic programming problems by delayed projection neural network. IEEE Transactions on Neural Networks 17 (6), 1630–1634.
Yang, Y., Cao, J., 2006b. A delayed neural network method for solving convex optimization problems. International Journal of Neural Systems 16 (4), 295–303.
Yang, Y., Cao, J., 2008. A feedback neural network for solving convex constraint optimization problems. Applied Mathematics and Computation 201 (1–2), 340–350.