Information Sciences 180 (2010) 697–711

Contents lists available at ScienceDirect

Information Sciences

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

A neural network based on the generalized Fischer–Burmeister function for nonlinear complementarity problems

Jein-Shan Chen [a,*,1], Chun-Hsu Ko [b], Shaohua Pan [c]

[a] Department of Mathematics, National Taiwan Normal University, Taipei 11677, Taiwan
[b] Department of Electrical Engineering, I-Shou University, Kaohsiung 840, Taiwan
[c] School of Mathematical Sciences, South China University of Technology, Guangzhou 510640, China

ARTICLE INFO

Article history:
Received 27 February 2008
Received in revised form 18 August 2009
Accepted 6 November 2009

Keywords:
The nonlinear complementarity problem
Neural network
Exponentially convergent
Generalized Fischer–Burmeister function

ABSTRACT

In this paper, we consider a neural network model for solving the nonlinear complementarity problem (NCP). The neural network is derived from an equivalent unconstrained minimization reformulation of the NCP, which is based on the generalized Fischer–Burmeister function $\phi_p(a,b) = \|(a,b)\|_p - (a+b)$. We establish the existence and the convergence of the trajectory of the neural network, and study its Lyapunov stability, asymptotic stability as well as exponential stability. It was found that a larger p leads to a better convergence rate of the trajectory. Numerical simulations verify the obtained theoretical results.

1. Introduction

For decades, the nonlinear complementarity problem (NCP) has attracted a lot of attention because of its wide applications in operations research, economics, and engineering [9,12]. Given a mapping $F : \mathbb{R}^n \to \mathbb{R}^n$, the NCP is to find a point $x \in \mathbb{R}^n$ such that

$$x \geqslant 0, \quad F(x) \geqslant 0, \quad \langle x, F(x) \rangle = 0, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product. Throughout this paper, we assume that F is continuously differentiable, and let $F = (F_1, \ldots, F_n)^T$ with $F_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, n$.

There have been many methods proposed for solving the NCP [9,12]. One of the most popular approaches is to reformulate the NCP as an unconstrained minimization problem via a merit function; see [14,19–21]. A merit function is a function whose global minimizers coincide with the solutions of the NCP. The class of NCP-functions defined below is used to construct a merit function.

Definition 1.1. A function $\phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is called an NCP-function if it satisfies

$$\phi(a,b) = 0 \Longleftrightarrow a \geqslant 0, \quad b \geqslant 0, \quad ab = 0. \tag{2}$$

A popular NCP-function is the Fischer–Burmeister (FB) function [10,11], which is defined as

* Corresponding author. Tel.: +886 2 29325417; fax: +886 2 29332342.
E-mail addresses: jschen@math.ntnu.edu.tw (J.-S. Chen), chko@isu.edu.tw (C.-H. Ko), shhpan@scut.edu.cn (S. Pan).
[1] Member of Mathematics Division, National Center for Theoretical Sciences, Taipei Office. The author's work is partially supported by National Science Council of Taiwan.

0020-0255/$ - see front matter
doi:10.1016/j.ins.2009.11.014

$$\phi_{\mathrm{FB}}(a,b)=\sqrt{a^2+b^2}-(a+b). \tag{3}$$

The FB merit function $\psi_{\mathrm{FB}}:\mathbb{R}\times\mathbb{R}\to\mathbb{R}_+$ can be obtained by taking the square of ϕ_{FB}, i.e.,

$$\psi_{\mathrm{FB}}(a,b):=\frac{1}{2}|\phi_{\mathrm{FB}}(a,b)|^2. \tag{4}$$

In [1,3,4], we studied a family of NCP-functions that subsumes the FB function ϕ_{FB} as a special case. More specifically, we define $\phi_p:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ by

$$\phi_p(a,b):=\|(a,b)\|_p-(a+b), \tag{5}$$

where p is any fixed real number from $(1,+\infty)$ and $\|(a,b)\|_p$ denotes the p-norm of (a,b), i.e., $\|(a,b)\|_p=\sqrt[p]{|a|^p+|b|^p}$. In other words, in the function ϕ_p, we replace the 2-norm of (a,b) in the FB function ϕ_{FB} by a more general p-norm of (a,b). The function ϕ_p is still an NCP-function, as noted in Tseng's paper [29]. There has been no further study on this NCP-function, even for $p=3$, until recently [1,3,4]. Similar to ϕ_{FB}, the square of ϕ_p induces a nonnegative NCP-function $\psi_p:\mathbb{R}\times\mathbb{R}\to\mathbb{R}_+$:

$$\psi_p(a,b):=\frac{1}{2}|\phi_p(a,b)|^2. \tag{6}$$

The function ψ_p is continuously differentiable and it has some favorable properties; see [1,3,4]. Moreover, if we define the function $\Psi_p:\mathbb{R}^n\to\mathbb{R}_+$ by

$$\Psi_p(x):=\sum_{i=1}^{n}\psi_p(x_i,F_i(x))=\frac{1}{2}\|\Phi_p(x)\|^2, \tag{7}$$

where $\Phi_p:\mathbb{R}^n\to\mathbb{R}^n$ is a mapping given as

$$\Phi_p(x)=\begin{pmatrix}\phi_p(x_1,F_1(x))\\ \vdots\\ \phi_p(x_n,F_n(x))\end{pmatrix}, \tag{8}$$

then the NCP can be reformulated into the following smooth minimization problem:

$$\min_{x\in\mathbb{R}^n}\Psi_p(x). \tag{9}$$

Thus, $\Psi_p(x)$ in (7) is a smooth merit function for the NCP.

Effective gradient-type methods can be applied to the unconstrained smooth minimization problem (9). However, in many scientific and engineering applications, it is desirable to have a real-time solution of the NCP. Thus, traditional unconstrained optimization algorithms may not be suitable for real-time implementation because the computing time required for a solution greatly depends on the dimension and structure of the problem. One promising way to overcome this problem is to apply neural networks.

Neural networks for optimization were first introduced in the 1980s by Hopfield and Tank [16,28]. Since then, neural networks have been applied to various optimization problems, including linear programming, nonlinear programming, variational inequalities, and linear and nonlinear complementarity problems; see [6,8,7,15,17,18,22,24,31–35]. There have been many studies on neural-network approaches to real-world problems in some other fields, such as [26,27,36]. The main idea of the neural-network approach for optimization is to construct a nonnegative energy function and establish a dynamic system that represents an artificial neural network. The dynamic system is usually in the form of first order ordinary 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 using integrated circuits.

In this paper, we focus on a neural-network approach to the NCP. We utilize $\Psi_p(x)$ as the traditional energy function. As mentioned above, the NCP is equivalent to the unconstrained smooth minimization problem (9). Therefore, it is natural to adopt the following steepest descent-based neural network model for NCP:

$$\frac{dx(t)}{dt}=-\rho\nabla\Psi_p(x(t)),\quad x(0)=x_0, \tag{10}$$

where $\rho>0$ is a scaling factor. Most neural networks in the existing literature are projection-type ones based on other kinds of NCP-functions, such as natural residual function (e.g. [18,33]) and the regularized gap function (e.g. [6]). Recently, neural networks based on the FB function have been designed for linear and quadratic programming, and for nonlinear complementarity problems [8,24]. Our model is based on the generalized FB function, which is a generalization of the functions used in [8,24]. We show that the neural network (10) is Lyapunov stable, asymptotically stable, and exponentially stable. We observed in [2] that p has a great influence on the numerical performance of certain descent-type methods; a larger p yields a better convergence rate, whereas a smaller p often gives a better global convergence. Thus, whether such phenomena occur in our neural network model is also investigated.

Throughout this paper, $\mathbb{R}^n$ denotes the space of n-dimensional real column vectors and ^{T} denotes the transpose. For any differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, $\nabla f(x)$ means the gradient of f at x. For any differentiable mapping $F = (F_1, \ldots, F_m)^T : \mathbb{R}^n \to \mathbb{R}^m$, $\nabla F(x) = [\nabla F_1(x) \cdots \nabla F_m(x)] \in \mathbb{R}^{n\times m}$ denotes the transposed Jacobian of F at x. The p-norm of x is denoted by $\|x\|_p$ and the Euclidean norm of x is denoted by $\|x\|$. Besides, e_i is the n-dimensional vector whose i-th component is 1 and 0 elsewhere. Unless otherwise stated, we assume that p in the sequel is any fixed real number in $(1, +\infty)$ if not specified.

2. Preliminaries

In this section, we review some properties of ϕ_p and ψ_p, as well as materials of ordinary differential equations that will play an important role in the subsequent analysis. We start with some concepts for a nonlinear mapping.

Definition 2.1. Let $F = (F_1, \ldots, F_n)^T : \mathbb{R}^n \to \mathbb{R}^n$. Then, the mapping F is said to be

(a) monotone if $\langle x - y, F(x) - F(y)\rangle \geqslant 0$ for all $x, y \in \mathbb{R}^n$;
(b) strongly monotone with modulus $\mu > 0$ if $\langle x - y, F(x) - F(y)\rangle \geqslant \mu\|x - y\|^2$ for all $x, y \in \mathbb{R}^n$;
(c) an P_0-function if $\max_{\substack{1\leqslant i\leqslant n \\ x_i \neq y_i}}(x_i - y_i)(F_i(x) - F_i(y)) \geqslant 0$ for all $x, y \in \mathbb{R}^n$ and $x \neq y$;
(d) a uniform P-function with modulus $\kappa > 0$ if $\max_{1\leqslant i\leqslant n}(x_i - y_i)(F_i(x) - F_i(y)) \geqslant \kappa\|x - y\|^2$, for all $x, y \in \mathbb{R}^n$;
(e) Lipschitz continuous if there exists a constant $L > 0$ such that $\|F(x) - F(y)\| \leqslant L\|x - y\|$ for all $x, y \in \mathbb{R}^n$.

From Definition 2.1, the following one-sided implications can be obtained:

$$\begin{aligned} F \text{ is strongly monotone } &\Rightarrow F \text{ is a uniform} P\text{-function } \Rightarrow F \text{ is an } P_0 \text{ function; } \nabla F \text{ is positive semidefinite} \\ &\Rightarrow F \text{ is monotone } \Rightarrow F \text{ is an } P_0 \text{ function.}\end{aligned}$$

Nevertheless, we point out that F being a uniform P-function does not necessarily imply that F is monotone. The following two lemmas summarize some favorable properties of ϕ_p and ψ_p, respectively. Since their proofs can be found in [2–4], we here omit them.

Lemma 2.1. *Let $\phi_p : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be given by (5). Then, the following properties hold.*

(a) *ϕ_p is a positive homogeneous and sub-additive NCP-function.*
(b) *ϕ_p is Lipschitz continuous with $L = \sqrt{2} + 2^{(1/p-1/2)}$ for $1 < p < 2$, and $L = \sqrt{2} + 1$ for $p \geqslant 2$.*
(c) *ϕ_p is strongly semismooth.*
(d) *If $\{(a^k, b^k)\} \subseteq \mathbb{R} \times \mathbb{R}$ with $a^k \to -\infty$, or $b^k \to -\infty$, or $a^k \to \infty, b^k \to \infty$, then $|\phi_p(a^k, b^k)| \to \infty$ when $k \to \infty$.*
(e) *Given a point $(a, b) \in \mathbb{R} \times \mathbb{R}$, every element in the generalized gradient $\partial\phi_p(a, b)$ has the representation $(\xi - 1, \zeta - 1)$ with*

$$\xi = \frac{\mathrm{sgn}(a) \cdot |a|^{p-1}}{\|(a, b)\|_p^{p-1}} \quad \text{and} \quad \zeta = \frac{\mathrm{sgn}(b) \cdot |b|^{p-1}}{\|(a, b)\|_p^{p-1}}$$

for $(a, b) \neq (0, 0)$, where $\mathrm{sgn}(\cdot)$ represents the sign function; otherwise, ξ and ζ are real numbers that satisfy $|\xi|^{\frac{p}{p-1}} + |\zeta|^{\frac{p}{p-1}} \leqslant 1$.

Lemma 2.2. *Let ϕ_p and ψ_p be defined as in (5) and (6), respectively. Then,*

(a) *$\psi_p(a, b) \geqslant 0$ for all $a, b \in \mathbb{R}$ and ψ_p is an NCP-function, i.e., it satisfies (2).*
(b) *ψ_p is continuously differentiable everywhere. Moreover, $\nabla_a\psi_p(a, b) = \nabla_b\psi_p(a, b) = 0$ if $(a, b) = (0, 0)$; otherwise,*

$$\begin{aligned} \nabla_a\psi_p(a, b) &= \left(\frac{\mathrm{sgn}(a) \cdot |a|^{p-1}}{\|(a, b)\|_p^{p-1}} - 1\right)\phi_p(a, b), \\ \nabla_b\psi_p(a, b) &= \left(\frac{\mathrm{sgn}(b) \cdot |b|^{p-1}}{\|(a, b)\|_p^{p-1}} - 1\right)\phi_p(a, b). \end{aligned} \tag{11}$$

(c) *$\nabla_a\psi_p(a, b) \cdot \nabla_b\psi_p(a, b) \geqslant 0$ for all $a, b \in \mathbb{R}$. The inequality becomes an equality if and only if $\phi_p(a, b) = 0$.*
(d) *$\nabla_a\psi_p(a, b) = 0 \Longleftrightarrow \nabla_b\psi_p(a, b) = 0 \Longleftrightarrow \phi_p(a, b) = 0 \Longleftrightarrow \psi_p(a, b) = 0$.*
(e) *The gradient of ψ_p is Lipschitz continuous for $p \geqslant 2$, i.e., there exists $L > 0$ such that*

$$\|\nabla\psi_p(a, b) - \nabla\psi_p(c, d)\| \leqslant L\|(a, b) - (c, d)\| \quad \text{for all } (a, b), (c, d) \in \mathbb{R}^2 \text{ and } p \geqslant 2.$$

(f) *For all $a, b \in \mathbb{R}$, we have $(2 - 2^{1/p})\min\{a, b\} \leqslant |\phi_p(a, b)| \leqslant (2 + 2^{1/p})\min\{a, b\}$.*

Next, we recall some materials about first order differential equations (ODE):

$$\dot{x}(t) = H(x(t)), \quad x(t_0) = x_0 \in \mathbb{R}^n, \tag{12}$$

where $H : \mathbb{R}^n \to \mathbb{R}^n$ is a mapping. We also introduce three kinds of stability that will be discussed later. These materials can be found in ODE textbooks; see [25].

Definition 2.2. A point $x^* = x(t^*)$ is called an equilibrium point or a steady state of the dynamic system (12) if $H(x^*) = 0$. If there is a neighborhood $\Omega^* \subseteq \mathbb{R}^n$ of x^* such that $H(x^*) = 0$ and $H(x) \neq 0\ \forall x \in \Omega^* \setminus \{x^*\}$, then x^* is called an isolated equilibrium point.

Lemma 2.3. *Assume that $H : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous mapping. Then, for any $t_0 \geqslant 0$ and $x_0 \in \mathbb{R}^n$, there exists a local solution $x(t)$ for (12) with $t \in [t_0, \tau)$ for some $\tau > t_0$. If, in addition, H is locally Lipschitz continuous at x_0, then the solution is unique; if H is Lipschitz continuous in $\mathbb{R}^n$, then τ can be extended to ∞.*

If a local solution defined on $[t_0, \tau)$ cannot be extended to a local solution on a larger interval $[t_0, \tau_1), \tau_1 > \tau$, then it is called a maximal solution, and the interval $[t_0, \tau)$ is the maximal interval of existence. Clearly, any local solution has an extension to a maximal one. We denote $[t_0, \tau(x_0))$ by the maximal interval of existence associated with x_0.

Lemma 2.4. *Assume that $H : \mathbb{R}^n \to \mathbb{R}^n$ is continuous. If $x(t)$ with $t \in [t_0, \tau(x_0))$ is a maximal solution and $\tau(x_0) < \infty$, then $\lim_{t \uparrow \tau(x_0)} \|x(t)\| = \infty$.*

Definition 2.3. Stability in the sense of LyapunovLet $x(t)$ be a solution for (12). An isolated equilibrium point x^* is Lyapunov stable if for any $x_0 = x(t_0)$ and any $\varepsilon > 0$, there exists a $\delta > 0$ such that $\|x(t) - x^*\| < \varepsilon$ for all $t \geqslant t_0$ and $\|x(t_0) - x^*\| < \delta$.

Definition 2.4 (*Asymptotic stability*). An isolated equilibrium point x^* is said to be asymptotically stable if in addition to being Lyapunov stable, it has the property that $x(t) \to x^*$ as $t \to \infty$ for all $\|x(t_0) - x^*\| < \delta$.

Definition 2.5 (*Lyapunov function*). Let $\Omega \subseteq \mathbb{R}^n$ be an open neighborhood of $\bar{x}$. A continuously differentiable function $W : \mathbb{R}^n \to \mathbb{R}$ is said to be a Lyapunov function at the state $\bar{x}$ over the set Ω for Eq. (12) if

$$\begin{cases} W(\bar{x}) = 0, \quad W(x) > 0, \quad \forall x \in \Omega \setminus \{\bar{x}\}. \\ \frac{dW(x(t))}{dt} = \nabla W(x(t))^T H(x(t)) \leqslant 0, \quad \forall x \in \Omega. \end{cases} \tag{13}$$

Lemma 2.5

(a) An isolated equilibrium point x^* is Lyapunov stable if there exists a Lyapunov function over some neighborhood Ω^* of x^*.
(b) An isolated equilibrium point x^* is asymptotically stable if there is a Lyapunov function over some neighborhood Ω^* of x^* such that $\frac{dW(x(t))}{dt} < 0$ for all $x \in \Omega^* \setminus \{x^*\}$.

Definition 2.6 (*Exponential stability*). An isolated equilibrium point x^* is exponentially stable if there exists a $\delta > 0$ such that arbitrary point $x(t)$ of (10) with the initial condition $x(t_0) = x_0$ and $\|x(t_0) - x^*\| < \delta$ is well-defined on $[0, +\infty)$ and satisfies

$$\|x(t) - x^*\|_2 \leqslant ce^{-\omega t}\|x(t_0) - x^*\| \quad \forall t \geqslant t_0,$$

where $c > 0$ and $\omega > 0$ are constants independent of the initial point.

3. Neural network model

We now discuss properties of the neural network model introduced in (10). First, from Lemma 2.2(a), we obtain the following result.

Proposition 3.1. *Let $\Psi_p : \mathbb{R}^n \to \mathbb{R}_+$ be defined as in (7). Then, $\Psi_p(x) \geqslant 0$ for all $x \in \mathbb{R}^n$ and $\Psi_p(x) = 0$ if and only if x solves the NCP.*

Proposition 3.2. *Let $\Psi_p : \mathbb{R}^n \to \mathbb{R}_+$ be given by (7). Then, the following results hold.*

(a) The function Ψ_p is continuously differentiable everywhere with

$$\nabla \Psi_p(x) = V^T \Phi_p(x) \quad \text{for any } V \in \partial \Phi_p(x) \tag{14}$$

or

$$\nabla \Psi_p(x) = \nabla_a \psi_p(x, F(x)) + \nabla F(x) \nabla_b \psi_p(x, F(x)) \tag{15}$$

with

$$\nabla_a \psi_p(x, F(x)) := \left[\nabla_a \psi_p(x_1, F_1(x)), \ldots, \nabla_a \psi_p(x_n, F_n(x))\right]^T,$$
$$\nabla_b \psi_p(x, F(x)) := \left[\nabla_b \psi_p(x_1, F_1(x)), \ldots, \nabla_b \psi_p(x_n, F_n(x))\right]^T.$$

(b) *If F is an P_0-function, then every stationary point of (9) is a global minimizer of $\Psi_p(x)$, and it consequently solves the NCP.*
(c) *If F is a uniform P-function, then the level sets $\mathscr{L}(\Psi_p, \gamma) := \{x \in \mathbb{R}^n | \Psi_p(x) \leqslant \gamma\}$ are bounded for all $\gamma \in \mathbb{R}$.*
(d) $\Psi_p(x(t))$ is nonincreasing with respect to t.

Proof. The first equality in (a) follows from Lemma 2.2(c) and [5, Theorem 2.6.6]. The second one follows from the chain rule. Part (b) is the result of [3, Proposition 3.4], and part (c) is the result of [4, Proposition 3.5]. It remains to show part (d). By the definition of $\Psi_p(x)$ and (10), it is not difficult to compute

$$\frac{d\Psi_p(x(t))}{dt} = \nabla \Psi_p(x(t))^T \frac{dx(t)}{dt} = \nabla \Psi_p(x(t))^T\left(-\rho \nabla \Psi_p(x(t))\right) = -\rho \|\nabla \Psi_p(x(t))\|^2 \leqslant 0. \tag{16}$$

Therefore, $\Psi_p(x(t))$ is a monotonically decreasing function with respect to t. □

Proposition 3.2(a) provides two ways to compute $\nabla \Psi_p(x)$, which is needed in the network (10). One is to use formula (14), for which we give an algorithm (see Algorithm 3.1 below), to evaluate an element $V \in \partial \Phi_p(x)$. The other is to adopt formula (15).

Algorithm 3.1. The procedure to evaluate an element $V \in \partial \Phi_p(x)$

(S.0) Let $x \in \mathbb{R}^n$ be given, and let V_i denote the i-th row of a matrix $V \in \mathbb{R}^{n \times n}$.
(S.1) Set $I(x) := \{i \in \{1, 2, \ldots, n\} |\ x_i = F_i(x) = 0\}$.
(S.2) Set $z \in \mathbb{R}^n$ such that $z_i = 0$ for $i \notin I(x)$, and $z_i = 1$ for $i \in I(x)$.
(S.3) For $i \in I(x)$, let $u_i = [|z_i|^{\frac{p}{p-1}} + |\nabla F_i(x)^T z|^{\frac{p}{p-1}}]^{\frac{p-1}{p}}$, and

$$V_i = \left(\frac{z_i}{u_i} - 1\right) e_i^T + \left(\frac{\nabla F_i(x)^T z}{u_i} - 1\right) \nabla F_i(x)^T.$$

(S.4) For $i \notin I(x)$, set

$$V_i = \left(\frac{\operatorname{sgn}(x_i) \cdot |x_i|^{p-1}}{\|(x_i, F_i(x))\|_p^{p-1}} - 1\right) e_i^T + \left(\frac{\operatorname{sgn}(F_i(x)) \cdot |F_i(x)|^{p-1}}{\|(x_i, F_i(x))\|_p^{p-1}} - 1\right) \nabla F_i(x)^T.$$

The above procedure is a traditional way of obtaining $\nabla \Psi_p(x(t))$. For example, the neural network in [24] uses (14) and a similar algorithm to evaluate an element of $V \in \partial \Phi_{\mathrm{FB}}(x)$. We propose a simpler way of obtaining $\nabla \Psi_p(x(t))$ which is to compute $\nabla \Psi_p(x(t))$ by using formula (15) rather than formula (14). Formula (15) also provides an indication on how the neural network (10) can be implemented on hardware; see Fig. 1.

To close this section, we claim that Ψ_p provides a global error bound for the solution of the NCP. This result is important and will be used to analyze the influence of p on the convergence rate of the trajectory $x(t)$ of the neural network (10) in the next section.

Proposition 3.3. Suppose F is a uniform P-function with modulus $\kappa > 0$ and Lipschitz continuous with constant $L > 0$. Then, the NCP has a unique solution x^*, and

$$\|x - x^*\|^2 \leqslant \frac{4L^2}{\kappa^2 (2 - 2^{1/p})^2} \Psi_p(x) \quad \forall x \in \mathbb{R}^n.$$

Proof. Since F is a uniform P-function, by Proposition 3.2(c), there exists a global minimizer of $\Psi_p(x)$ which says the NCP has a solution. Assume that the NCP has two different solutions x^* and y^*, then by Definition 2.1(d) we have

$$\kappa \|x^* - y^*\|^2 \leqslant \max_{1 \leqslant i \leqslant m} (x_i^* - y_i^*)(F_i(x^*) - F_i(y^*)) = \max_{1 \leqslant i \leqslant m} \left\{-x_i^* F_i(y^*) - y_i^* F_i(x^*)\right\} \leqslant 0,$$

where the equality is due to the fact that $x_i^* F_i(x^*) = y_i^* F_i(y^*) = 0$ for $i = 1, 2, \ldots, n$ (note that x^* and y^* are the solutions to the NCP), and the last inequality holds since $x^*, y^* \geqslant 0$ and $F(x^*), F(y^*) \geqslant 0$. This leads to a contradiction. Hence, the NCP has a unique solution.

For any $x \in \mathbb{R}^n$, let $r(x) := (r_1(x), \ldots, r_n(x))^T$ with $r_i(x) = \min\{x_i, F_i(x)\}$ for $i = 1, \ldots, n$. Since F is Lipschitz continuous with constant $L > 0$, by [21, Lemma 7.4] we have

$$(x_i - x_i^*)(F_i(x) - F_i(x^*)) \leqslant 2L |r_i(x)| \|x - x^*\|,$$

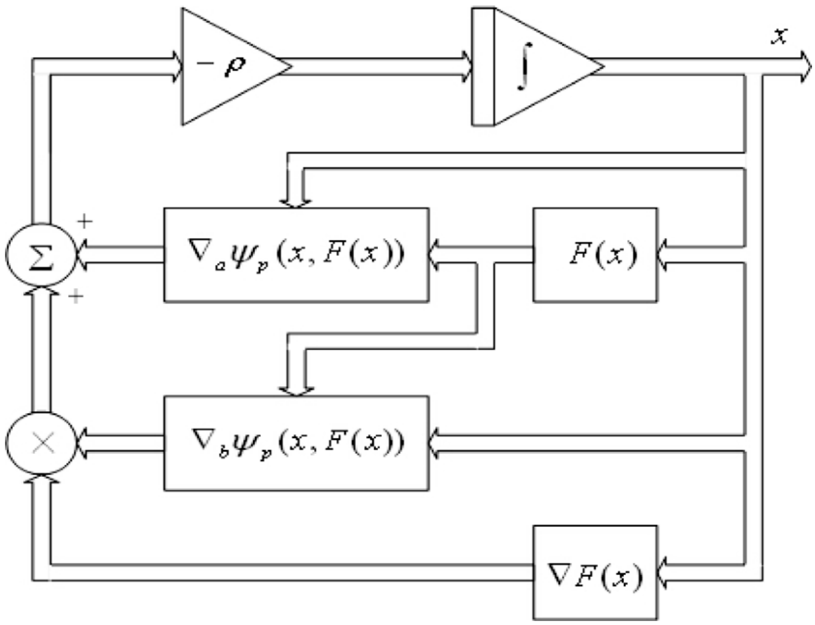

Fig. 1. A simplified block diagram for the neural network (10).

for all $x \in \mathbb{R}^n$ and $i = 1, 2, \ldots, n$. On the other hand, since F is a uniform P-function with modulus $\kappa > 0$, from Definition 2.1(d) it follows that

$$\kappa \|x - x^*\|^2 \leqslant \max_{1 \leqslant i \leqslant n} (x_i - x_i^*)(F_i(x) - F_i(x^*))$$

for any $x \in \mathbb{R}^n$. Combining the last two equations yields

$$\|x - x^*\| \leqslant (2L/\kappa) \max_{1 \leqslant i \leqslant n} |r_i(x)| \quad \forall x \in \mathbb{R}^n.$$

This together with Lemma 2.2(f) implies

$$\|x - x^*\| \leqslant \frac{2L}{\kappa(2 - 2^{1/p})} \max_{1 \leqslant i \leqslant n} |\phi_p(x_i, F_i(x))| \leqslant \frac{2L}{\kappa(2 - 2^{1/p})} \|\Phi_p(x)\|.$$

Consequently, we obtain the desired result. □

4. Convergence and stability of the trajectory

This section focuses on issues of convergence and stability of the neural network (10). We analyze the behavior of the solution trajectory of (10) including the existence and convergence, and establish three kinds of stability for an isolated equilibrium point. We first state the relationships between an equilibrium point of (10) and a solution to the NCP.

Proposition 4.1

(a) Every solution to the NCP is an equilibrium point of (10).
(b) If F is an P_0-function, then every equilibrium point of (10) is a solution to the NCP.

Proof

(a) Suppose that x is a solution to the NCP. Then, from Proposition 3.1, it is clear that $\Phi_p(x) = 0$. Using Lemma 2.2(d) and (15), we then have $\nabla \Psi_p(x) = 0$. This, by Definition 2.2, shows that x is an equilibrium point of (10).
(b) This is a direct consequence of Proposition 3.2(b). □

The following proposition establishes the existence of the solution trajectory of (10).

Proposition 4.2. *For any fixed $p \geqslant 2$, the following hold.*

(a) For any initial state $x_0 = x(t_0)$, there exists exactly one maximal solution $x(t)$ with $t \in [t_0, \tau(x_0))$ for the neural network (10).
(b) If the level set $\mathscr{L}(x_0) = \{x \in \mathbb{R}^n | \Psi_p(x) \leqslant \Psi_p(x_0)\}$ is bounded or F is Lipschitz continuous, then $\tau(x_0) = +\infty$.

Proof

(a) Since F is continuously differentiable, $\nabla F(x)$ is continuous, and therefore, $\nabla F(x)$ is bounded on a local compact neighborhood of x. On the other hand, $\nabla_a \psi_p$ and $\nabla_b \psi_p$ are Lipschitz continuous by Lemma 2.2(e). These two facts together with formula (15) show that $\nabla \Psi_p(x)$ is locally Lipschitz continuous. Thus, applying Lemma 2.3 leads to the desired result.

(b) We proceed the arguments by the two cases as shown below.

Case (i): The level set $\mathscr{L}(x_0)$ is bounded. We prove the result by contradiction. Suppose $\tau(x_0) < \infty$. Then, by Lemma 2.4, $\lim_{t\uparrow\tau(x_0)} \|x(t)\| = \infty$. Let $\mathscr{L}^c(x_0) := \mathbb{R}^n \setminus \mathscr{L}(x_0)$ and

$$\tau_0 := \inf\{s \geqslant 0 | s < \tau(x_0), x(s) \in \mathscr{L}^c(x_0)\} < \infty.$$

We know that $x(\tau_0)$ lies on the boundary of $\mathscr{L}(x_0)$ and $\mathscr{L}^c(x_0)$. Moreover, $\mathscr{L}(x_0)$ is compact since it is bounded by assumption and it is also closed because of the continuity of $\Psi_p(x)$. Therefore, we have $x(\tau_0) \in \mathscr{L}(x_0)$ and $\tau_0 < \tau(x_0)$, implying that

$$\Psi_p(x(s)) > \Psi_p(x_0) > \Psi_p(x(\tau_0)) \quad \text{for some } s \in (\tau_0, \tau(x_0)). \tag{17}$$

However, Proposition 3.2(d) says that $\Psi_p(x(\cdot))$ is nonincreasing on $[t_0, \tau(x_0))$, which contradicts (17). This completes the proof of Case (i).

Case (ii): F is Lipschitz continuous. From the proof of part (a), we know that $\nabla \Psi_p(x)$ is Lipschitz continuous. Thus, by Lemma 2.3, we have $\tau(x_0) = \infty$. $\square$

Next, we investigate the convergence of the solution trajectory of (10).

Theorem 4.1

(a) Let $x(t)$ with $t \in [t_0, \tau(x_0))$ be the unique maximal solution to (10). If $\tau(x_0) = \infty$ and $\{x(t)\}$ is bounded, then $\lim_{t\to\infty} \nabla \Psi_p(x(t)) = 0$.

(b) If F is strongly monotone or a uniform P-function, then $\mathscr{L}(x_0)$ is bounded and every accumulation point of the trajectory $x(t)$ is a solution to the NCP.

Proof. With Proposition 3.2 (b) and (d) and Proposition 4.2, the arguments are exactly the same as those for [24, Corollary 4.3]. Thus, we omit them. $\square$

From Proposition 4.1 (a), every solution x^* to the NCP is an equilibrium point of the neural network (10). If, in addition, x^* is an isolated equilibrium point of (10), then we can show that x^* is not only Lyapunov stable but also asymptotically stable.

Theorem 4.2. *Let x^* be an isolated equilibrium point of the neural network (10). Then, x^* is Lyapunov stable for (10), and furthermore, it is asymptotically stable.*

Proof. Since x^* is a solution to the NCP, $\Psi_p(x^*) = 0$. In addition, since x^* is an isolated equilibrium point of (10), there exists a neighborhood $\Omega^* \subseteq \mathbb{R}^n$ of x^* such that

$$\nabla \Psi_p(x^*) = 0, \quad \text{and} \quad \nabla \Psi_p(x) \neq 0 \quad \forall x \in \Omega^* \setminus \{x^*\}.$$

Next, we argue that $\Psi_p(x)$ is indeed a Lyapunov function at x^* over the set Ω^* for (10) by showing that the conditions in (13) are satisfied. First, notice that $\Psi_p(x) \geqslant 0$. Suppose that there is an $\bar{x} \in \Omega^* \setminus \{x^*\}$ such that $\Psi_p(\bar{x}) = 0$. Then, by formula (15) and Lemma 2.2(d), we have $\nabla \Psi(\bar{x}) = 0$, i.e., $\bar{x}$ is also an equilibrium point of (10), which clearly contradicts the assumption that x^* is an isolated equilibrium point in Ω^*. Thus, we prove that $\Psi_p(x) > 0$ for any $x \in \Omega^* \setminus \{x^*\}$. This together with (16) shows that the conditions in (13) are satisfied, and hence $\Psi_p(x)$ is a Lyapunov function at x^* over the set Ω^* for (10). Therefore, x^* is Lyapunov stable by Lemma 2.5(a).

Now, we show that x^* is asymptotically stable. Since x^* is isolated, from (16) we have

$$\frac{d\Psi_p(x(t))}{dt} < 0, \quad \forall\, x(t) \in \Omega^* \setminus \{x^*\}.$$

This, by Lemma 2.5(b), implies that x^* is asymptotically stable. $\square$

Furthermore, using the same arguments we can prove that the neural network (10) is also exponentially stable if x^* is a *regular solution* to the NCP. Recall that x^* is a regular solution to the NCP if every element $V \in \partial \Phi_p(x^*)$ is nonsingular.

Theorem 4.3. *If x^* is a regular solution of the NCP, then it is* exponentially stable.

Remark 4.1

(a) Using arguments similar to those used in Proposition 3.2 of [13], we can prove that x^* is regular if $\nabla F_{\alpha\alpha}$ is nonsingular and the Schur complement of $\nabla F_{\alpha\alpha}$ in

$$\begin{pmatrix} \nabla F_{\alpha\alpha}(x^*) & \nabla F_{\alpha\beta}(x^*) \\ \nabla F_{\beta\alpha}(x^*) & \nabla F_{\beta\beta}(x^*) \end{pmatrix}$$

is an P-matrix, where $\alpha := \{i | x_i^* > 0\}$ and $\beta := \{i | x_i^* = F_i(x^*) = 0\}$. Clearly, if ∇F is positive definite, then the conditions hold true.

(b) From Definition 2.6, if an isolated equilibrium point x^* is exponentially stable, then there exists a $\delta > 0$ such that $x(t)$ with $x_0 = (t_0)$, and $\|x(t_0) - x^*\| < \delta$ satisfies

$$\|x(t) - x^*\| \leqslant ce^{-\omega t}\|x(t_0) - x^*\| \quad \forall t \geqslant t_0,$$

which together with Proposition 3.3 implies that

$$\|x(t) - x^*\| \leqslant \frac{2cL}{\kappa(2 - 2^{1/p})}\sqrt{\Psi_p(x_0)}e^{-\omega t} \quad \forall t \geqslant t_0. \tag{18}$$

Since the strong monotonicity of F implies that F is a uniform P-function and that ∇F is positive definite, from (18) we obtain that the neural network (10) can yield a trajectory with an exponential convergence rate under the condition that F is strongly monotone and Lipschitz continuous.

(c) We observe from (18) that, when p increases, the coefficient of $e^{-\omega t}$ in the right hand side term becomes smaller, which in turn implies that a larger p yields a better convergence rate. This agrees with the result obtained by [2] for a descent-type method based on Ψ_p. In addition, from (18) we notice that the energy of the initial state, i.e., $\Psi_p(x_0)$ also has an influence on the convergence rate. A higher initial energy will lead to a worse convergence rate.

5. Simulation results

In this section, we test four well-known nonlinear complementarity problems by our neural network model (10). For each test problem, we also compare the numerical performance of the proposed neural network with various values of p and various initial states $x(t_0)$. The test instances are described below.

Example 5.1 [31, Example 2]. Consider the NCP, where $F : \mathbb{R}^5 \to \mathbb{R}^5$ is given by

$$F(x) = \begin{pmatrix} x_1 + x_2x_3x_4x_5/50 \\ x_2 + x_1x_3x_4x_5/50 - 3 \\ x_3 + x_1x_2x_4x_5/50 - 1 \\ x_4 + x_1x_2x_3x_5/50 + 1/2 \\ x_5 + x_1x_2x_3x_4/50 \end{pmatrix}.$$

The NCP has only one solution $x^* = (0, 3, 1, 0, 0)$.

Example 5.2 [30, Watson]. Consider the NCP, where $F : \mathbb{R}^5 \to \mathbb{R}^5$ is given by

$$F(x) = 2\exp\left(\sum_{i=1}^{5}(x_i - i + 2)^2\right)\begin{pmatrix} x_1 + 1 \\ x_2 \\ x_3 - 1 \\ x_4 - 2 \\ x_5 - 3 \end{pmatrix}.$$

Note that F is not a P_0-function on $\mathbb{R}^n$. The solution to this problem is $x^* = (0, 0, 1, 2, 3)$.

Example 5.3 [23, Kojima–Shindo]. Consider the NCP, where $F : \mathbb{R}^4 \to \mathbb{R}^4$ is given by

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

This is a non-degenerate NCP and the solution is $x^* = (\sqrt{6}/2, 0, 0, 1/2)$.

Example 5.4 [23, Kojima–Shindo]. Consider the NCP, where $F:\mathbb{R}^4 \to \mathbb{R}^4$ is given by

$$F(x) = \begin{pmatrix} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_1 + x_2^2 + 10x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 9x_4 - 9 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{pmatrix}.$$

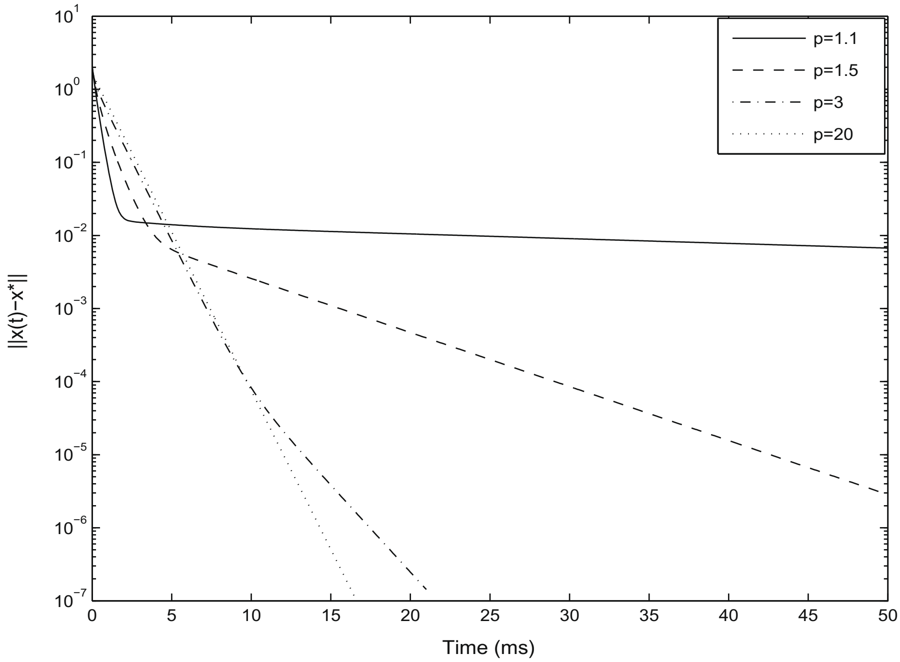

Fig. 2. Convergence behavior of the error $\|x(t) - x^*\|$ in Example 5.1 with given x_0.

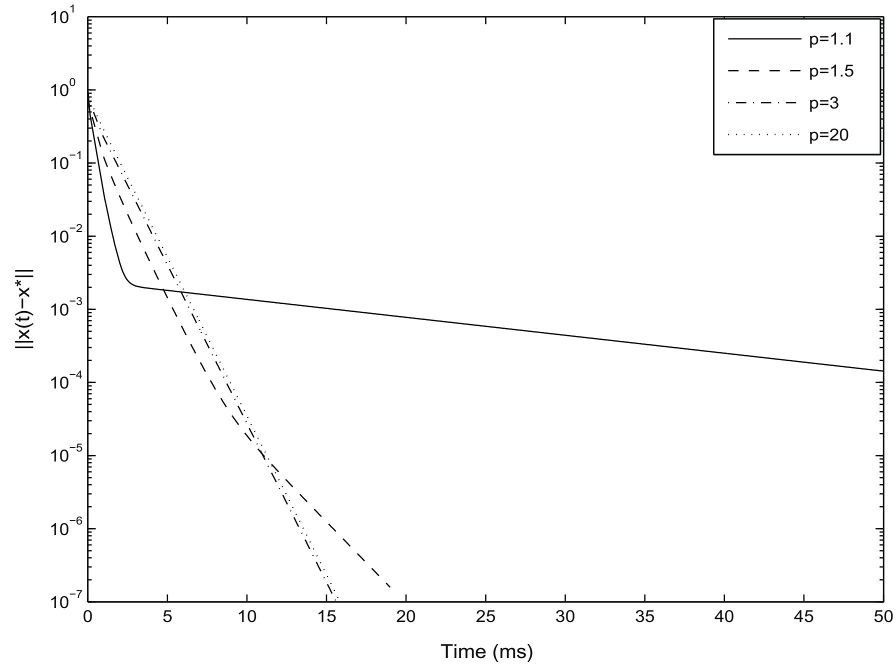

Fig. 3. Convergence behavior of the error $\|x(t) - x^*\|$ in Example 5.2 with given x_0.

This is a degenerate NCP and has two solutions $x^* = (\sqrt{6}/2, 0, 0, 1/2)$ and $x^* = (1, 0, 3, 0)$.

The numerical implementation is coded by Matlab 7.0 and the ordinary differential equation solver adopted is *ode23*, which uses an Runge–Kutta (2,3) formula. We first test the influence of the parameter p on the value of $\|x(t) - x^*\|$. Figs. 2–5 in the appendix describe how $\|x(t) - x^*\|$ varies with p for these instances with the initial states $x_0 = (10^{-2}, 1, 0.5, 10^{-2}, 10^{-2})^T, x_0 = (10^{-2}, 10^{-2}, 0.5, 0.5, 0.5)^T, x_0 = (2, 10^{-2}, 10^{-2}, 0.1)^T$, and $x_0 = (10^{-3}, 10^{-3}, 10^{-3}, 10^{-3})^T$,

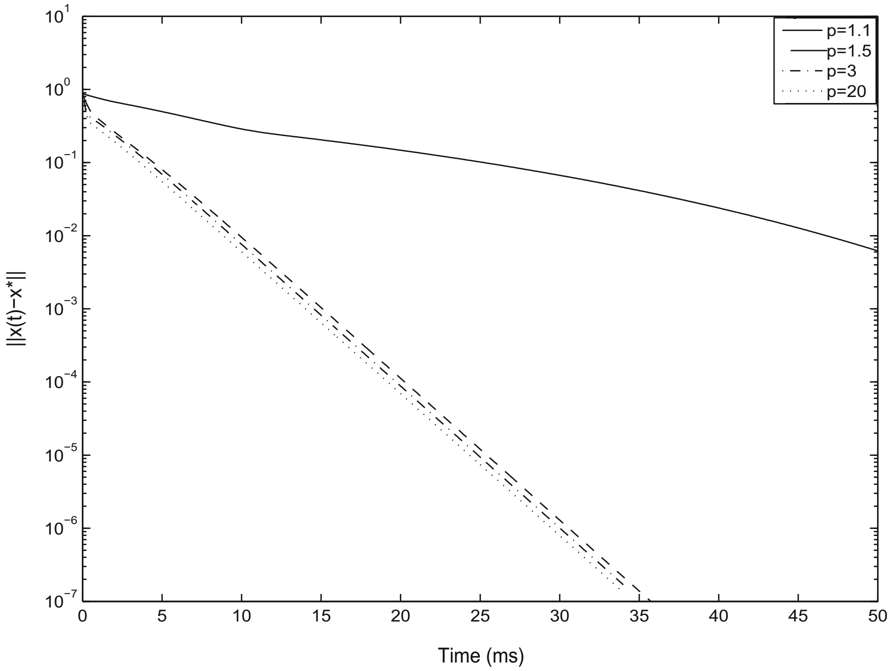

Fig. 4. Convergence behavior of the error $\|x(t) - x^*\|$ in Example 5.3 with given x_0.

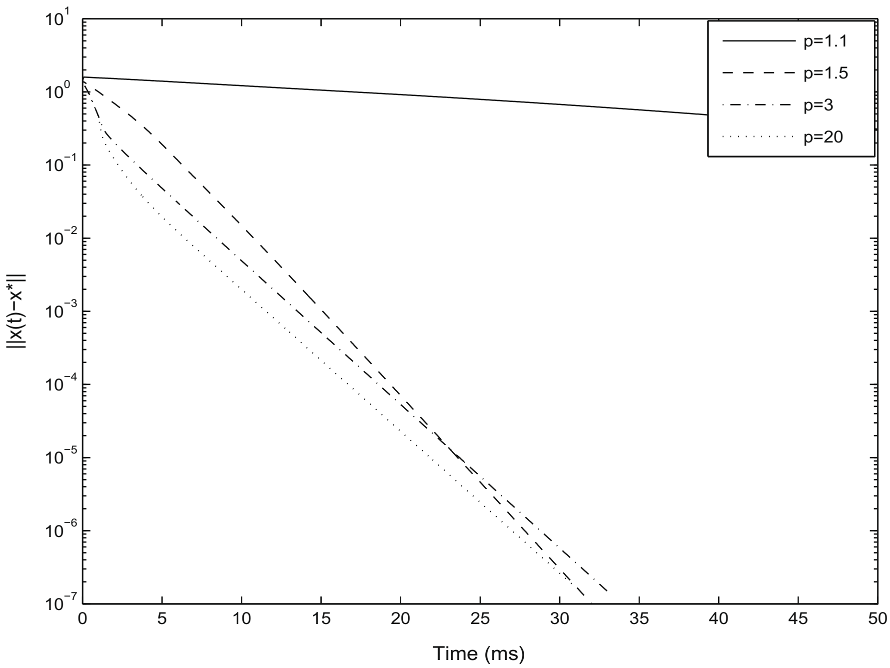

Fig. 5. Convergence behavior of the error $\|x(t) - x^*\|$ in Example 5.4 with given x_0.

respectively. In the tests, the design parameter ρ in the neural network (10) is set to be 1000. From Figs. 2–5, we see that, when $p = 1.1$, the neural network (10) generates the slowest decrease of $\|x(t) - x^*\|$ for all test instances, whereas when $p = 20$ it generates the fastest decrease of $\|x(t) - x^*\|$. This verifies the analysis of Remark 4.1(c). We should emphasize that the conclusion in Remark 4.1(c) requires the initial state x_0 to be sufficiently close to an equilibrium point. If this condition is not satisfied, we cannot draw such conclusion; see Fig. 6.

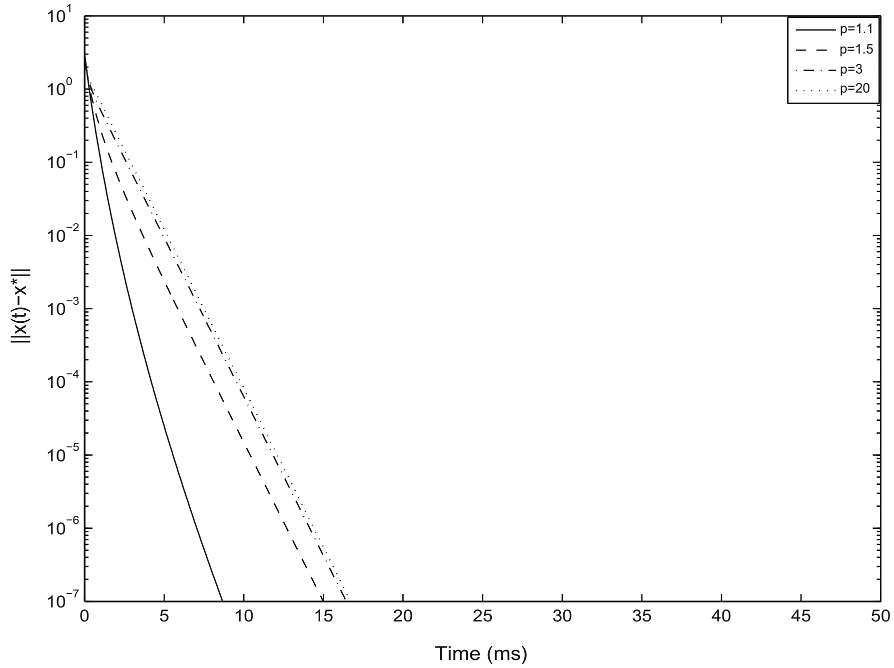

Fig. 6. Convergence behavior of $\|x(t) - x^*\|$ in Example 5.1 with $x_0 = [0, 0, 0, 0, 0]$.

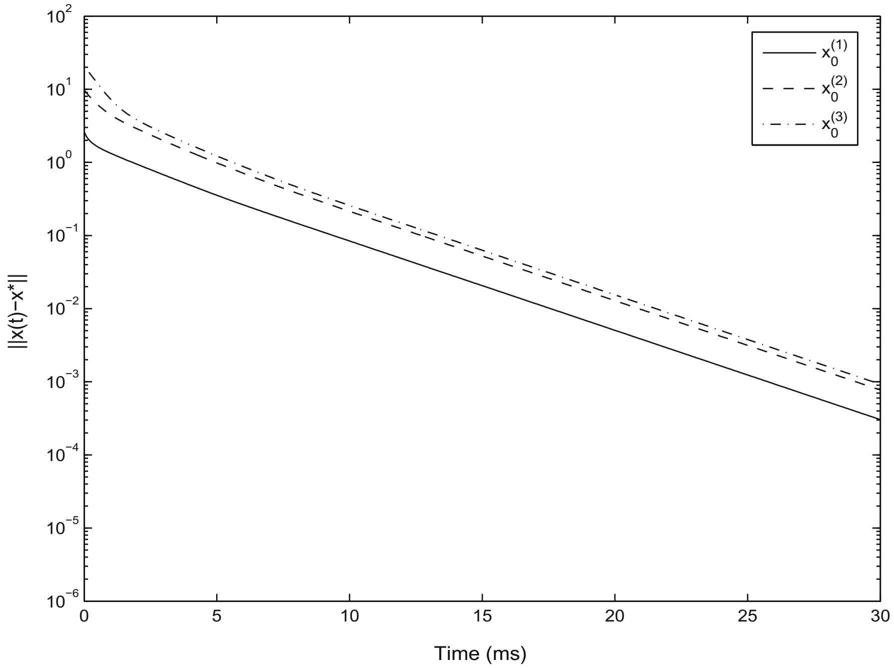

Fig. 7. Convergence behavior of $\|x(t) - x^*\|$ in Example 5.1 with three different initial points $x_0^{(1)}$, $x_0^{(2)}$, and $x_0^{(3)}$ $(p = 1.8)$.

Example 5.1 shows how the value of $\|x(t) - x^*\|$ varies with initial state x_0. Fig. 7 describes the convergence behavior of $\|x(t) - x^*\|$ with initial states $x_0^{(1)} = (1,1,1,1,1)^T$, $x_0^{(2)} = (5,5,5,5,5)^T$, and $x_0^{(3)} = (10,10,10,10,10)^T$. Notice that the initial energies corresponding to these three states are $\Psi_p(x_0^{(1)}) = 5.814$, $\Psi_p(x_0^{(2)}) = 39.367$, and $\Psi_p(x_0^{(3)}) = 226.333$, respectively.

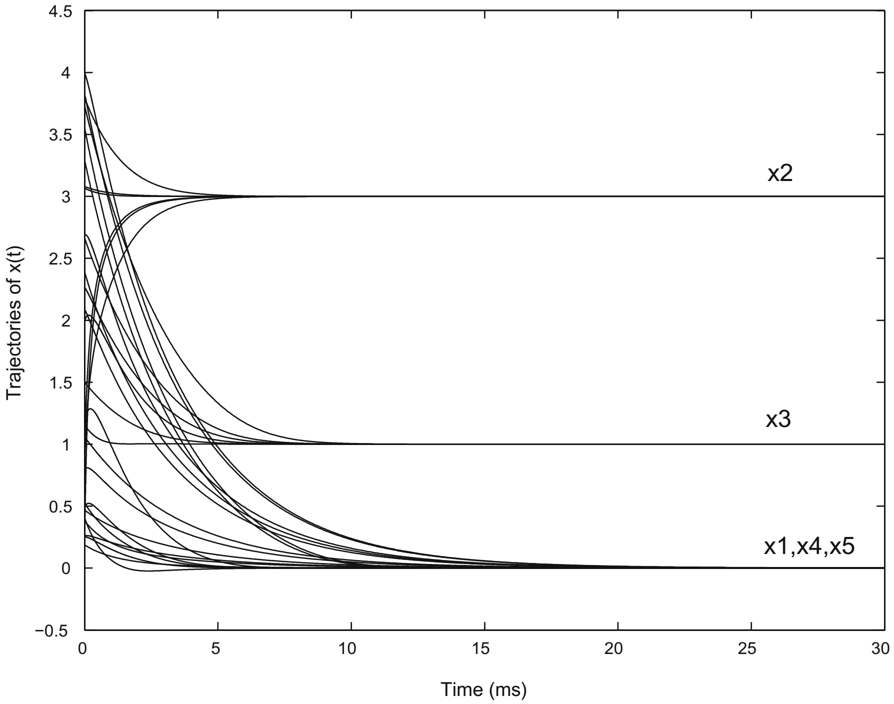

Fig. 8. Transient behavior of $x(t)$ of the neural network with 6 random initial points and $p = 1.8$ in Example 5.1.

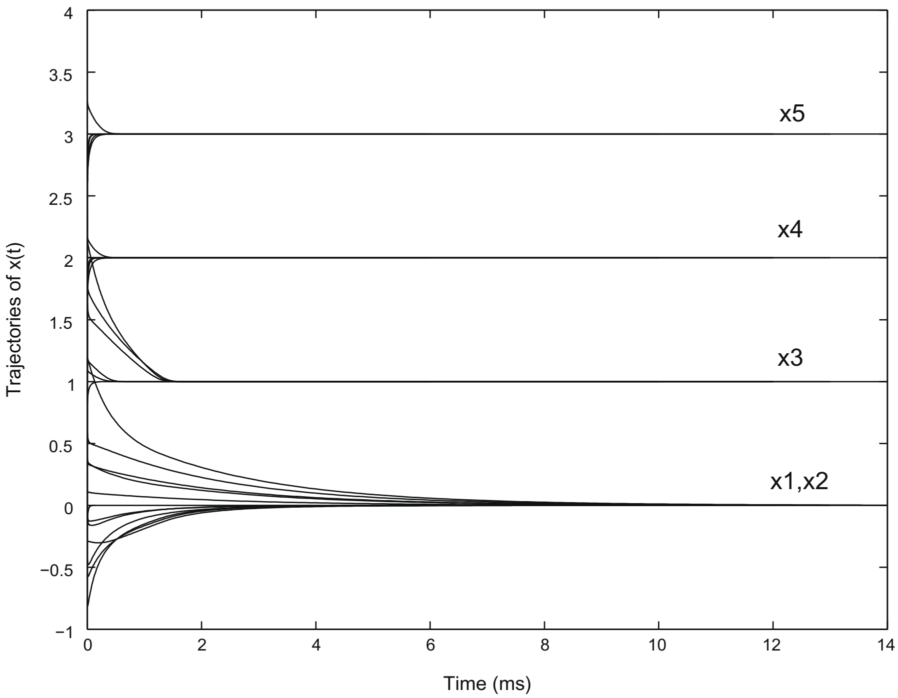

Fig. 9. Transient behavior of $x(t)$ of the neural network with 6 random initial points and $p = 1.4$ in Example 5.2.

In the tests, we choose $p = 1.8$ and $\rho = 1000$. Fig. 7, shows that a larger initial energy yields a slower decrease of the error $\|x(t) - x^*\|$ if the initial state is close to the solution of the NCP. This agrees with the analysis in Remark 4.1(c).

The convergence behavior of $x(t)$ from several initial states with a fixed p and $\rho = 1000$ for each example is shown in Figs. 8–12. The transient behavior of $x(t)$ for Example 5.4 is depicted in Figs. 11 and 12 since there are two solutions for this prob-

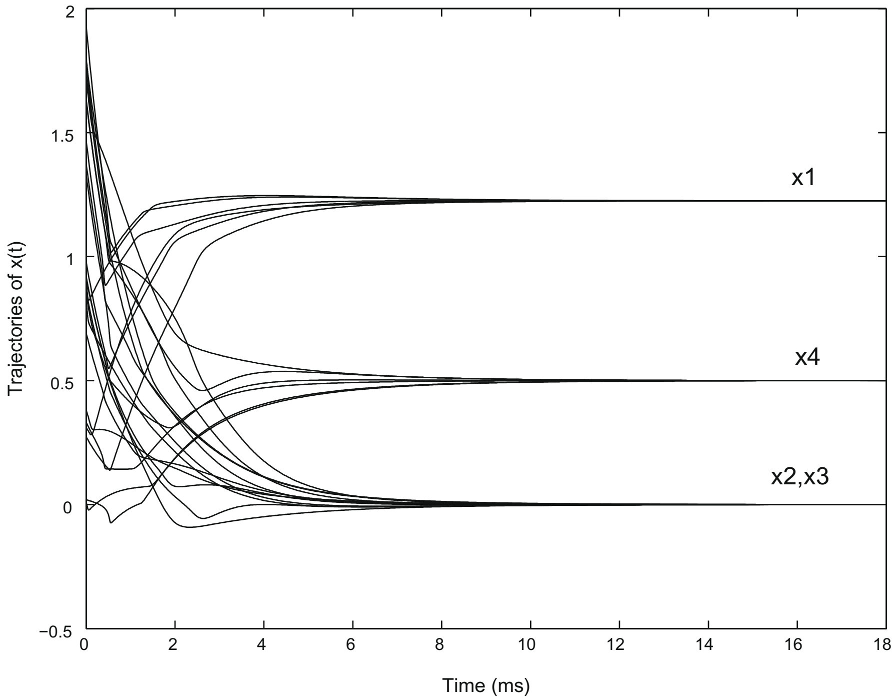

Fig. 10. Transient behavior of $x(t)$ of the neural network with 6 random initial points and $p = 2.4$ in Example 5.3.

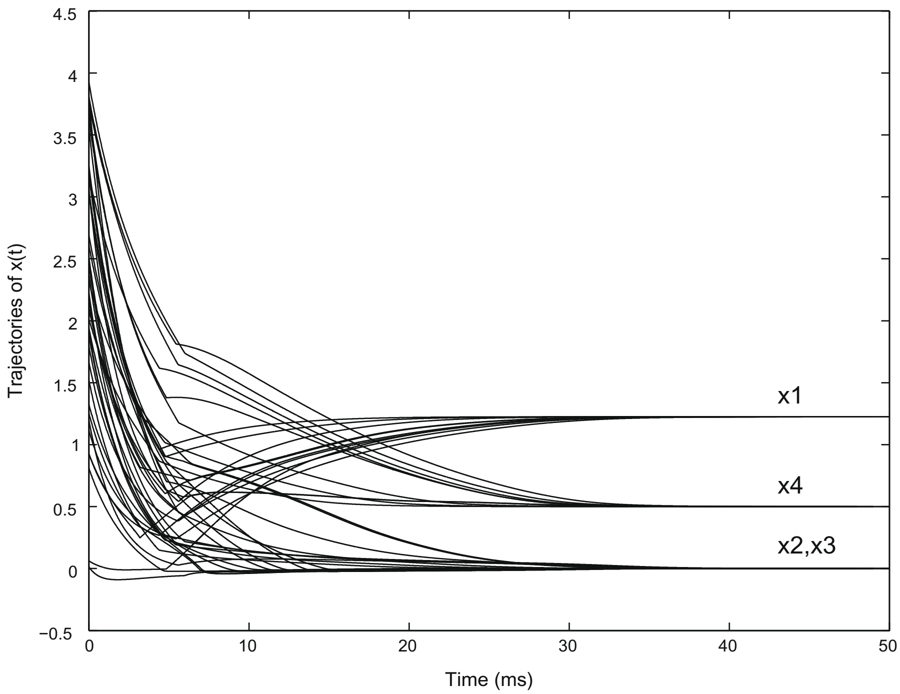

Fig. 11. Transient behavior of $x(t)$ of the neural network with 9 random initial points and $p = 1.2$ in Example 5.4.

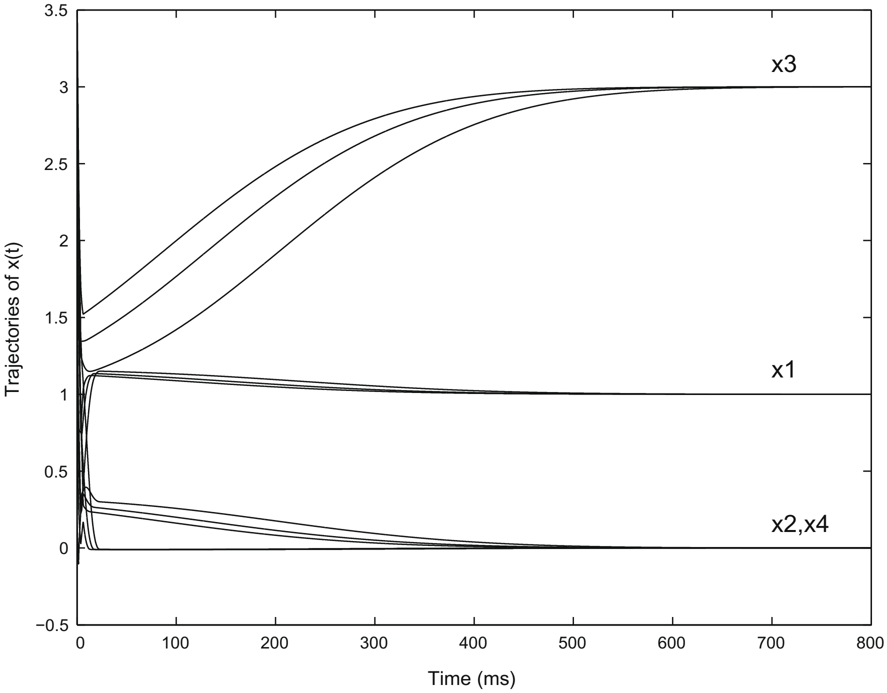

Fig. 12. Transient behavior of $x(t)$ of the neural network with 3 random initial points and $p = 1.2$ in Example 5.4.

lem. More specifically, we test 12 random initial points for the NCP, 9 of which converge to $(\sqrt{6}/2, 0, 0, 1/2)$; the remaining 3 converge to $(1, 0, 3, 0)$. When finding the solution trajectory $x(t)$, we employ $\|\nabla \Psi_p(x(t))\| \leqslant 10^{-5}$ as the stopping criterion.

To sum up, the neural network (10) is a better alternative for the network based on the FB function ϕ_{FB} if an appropriate p is chosen. Based on the analysis of Remark 4.1(c) and the above numerical simulations, we see that, to obtain a better convergence rate of the trajectory $x(t)$, the parameter p cannot be set too small. In addition, we should emphasize that the initial state $x(t_0)$ has a great influence on the convergence behavior of $\|x(t) - x^*\|$.

To end this section, we answer a natural question: are there advantages of our proposed neural network compared to the existing ones? To answer this, we summarize what we have observed from numerical experiments and theoretical results as below.

- We compare our neural network model with some existing models which also work for NCP, for instance, the ones used in [6,31,32]. At first glance, the neural network models based on projection in [6,31,32] look having lower complexity. However, we observe that the difference of the numerical performance is very marginal by testing MCPLIB benchmark problems.
- Our proposed model seems having better properties from theoretical view. Note that there requires monotonicity (strong monotonicity) of F to guarantee the Lyapunov stability (exponential stability) of the neural network models used in [6,31,32]. In contrast, such conditions are not needed for our neural network model. In fact, it can be verified that all F's are non-monotone in previous examples except Example 5.2 (by checking the positive semi-definiteness of their Jacobian matrices).
- For the following special NCP:

$$x = (x_1, x_2, x_3) \geqslant 0, \quad F(x) = (x_1, -x_2, -x_3) \geqslant 0, \quad \langle x, F(x) \rangle = x_1^2 - x_2^2 - x_3^2 = 0,$$

  it is easy to verify that the unique solution is $(0, 0, 0)$ which can be solved easily by our neural network model. But, the solution trajectory diverges by using the model in [31].
- Changing initial points may not having much effect for our neural network model, whereas it does for other existing models. For instance, choosing $x_0 = (12, -12, 12, -12, 12)$ as the initial point in Example 5.1 causes the divergence of solution trajectory solved by the neural network model used in [31], while it does not affect anything by our neural network model.

6. Conclusions

In this paper, we have studied a (class of) neural network based on the generalized FB function ϕ_p defined as in (5). We establish the Lyapunov stability, the asymptotic stability, and the exponential stability for the neural network. In addition, we also analyze the influence of the parameter p on the convergence rate of the trajectory (or the local convergence behavior of the error $\|x(t) - x^*\|$) and obtain that a larger p leads to a better convergence rate. This agrees with the result obtained by [2] for a descent-type method based on ϕ_p, which also indicates how to choose a suitable p in practice. Numerical experiments verify the obtained theoretical results. The advantages of our proposed neural network compared to other existing neural networks are reported as well. One future topic is to modify the proposed neural network model for various optimization problems and establish its related stability accordingly.

Appendix

See Figs. 2–12.

References

[1] J.-S. Chen, The semismooth-related properties of a merit function and a descent method for the nonlinear complementarity problem, Journal of Global Optimization 36 (2006) 565–580.
[2] J.-S. Chen, H.-T. Gao, S.-H. Pan, A derivative-free R-linearly convergent algorithm based on the generalized Fischer–Burmeister merit function, Journal of Computational and Applied Mathematics, submitted for publication.
[3] J.-S. Chen, S.-H. Pan, A family of NCP functions and a descent method for the nonlinear complementarity problem, Computational Optimization and Applications 40 (2008) 389–404.
[4] J.-S. Chen, S.-H. Pan, A regularization semismooth Newton method based on the generalized Fischer–Burmeister function for P_0-NCPs, Journal of Computational and Applied Mathematics 220 (2008) 464–479.
[5] F.H. Clarke, Optimization and Nonsmooth Analysis, Wiley, New York, 1983.
[6] C. Dang, Y. Leung, X. Gao, K. Chen, Neural networks for nonlinear and mixed complementarity problems and their applications, Neural Networks 17 (2004) 271–283.
[7] S. Effati, A. Ghomashi, A.R. Nazemi, Application of projection neural network in solving convex programming problems, Applied Mathematics and Computation 188 (2007) 1103–1114.
[8] S. Effati, A.R. Nazemi, Neural network and its application for solving linear and quadratic programming problems, Applied Mathematics and Computation 172 (2006) 305–331.
[9] M.C. Ferris, O.L. Mangasarian, J.-S. Pang (Eds.), Complementarity: Applications, Algorithms and Extensions, Kluwer Academic Publishers, Dordrecht, 2001.
[10] A. Fischer, A special Newton-type optimization methods, Optimization 24 (1992) 269–284.
[11] A. Fischer, Solution of the monotone complementarity problem with locally Lipschitzian functions, Mathematical Programming 76 (1997) 513–532.
[12] F. Facchinei, J.-S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, vols. I and II, Springer-Verlag, New York, 2003.
[13] F. Facchinei, J. Soares, A new merit function for nonlinear complementarity problems and a related algorithm, SIAM Journal on Optimization 7 (1997) 225–247.
[14] C. Geiger, C. Kanzow, On the resolution of monotone complementarity problems, Computational Optimization and Applications 5 (1996) 155–173.
[15] Q. Han, L.-Z. Liao, H. Qi, L. Qi, Stability analysis of gradient-based neural networks for optimization problems, Journal of Global Optimization 19 (2001) 363–381.
[16] J.J. Hopfield, D.W. Tank, Neural computation of decision in optimization problems, Biological Cybernetics 52 (1985) 141–152.
[17] X. Hu, J. Wang, Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network, IEEE Transactions on Neural Networks 17 (2006) 1487–1499.
[18] X. Hu, J. Wang, A recurrent neural network for solving a class of general variational inequalities, IEEE Transactions on Systems, Man, and Cybernetics – B 37 (2007) 528–539.
[19] H. Jiang, Unconstrained minimization approaches to nonlinear complementarity problems, Journal of Global Optimization 9 (1996) 169–181.
[20] C. Kanzow, Nonlinear complementarity as unconstrained optimization, Journal of Optimization Theory and Applications 88 (1996) 139–155.
[21] C. Kanzow, M. Fukushima, Equivalence of the generalized complementarity problem to differentiable unconstrained minimization, Journal of Optimization Theory and Applications 90 (1996) 581–603.
[22] M.P. Kennedy, L.O. Chua, Neural network for nonlinear programming, IEEE Transaction on Circuits and Systems 35 (1988) 554–562.
[23] M. Kojima, S. Shindo, Extensions of Newton and quasi-Newton methods to systems of PC^1 equations, Journal of Operations Research Society of Japan 29 (1986) 352–374.
[24] L.-Z. Liao, H. Qi, L. Qi, Solving nonlinear complementarity problems with neural networks: a reformulation method approach, Journal of Computational and Applied Mathematics 131 (2001) 342–359.
[25] R.K. Miller, A.N. Michel, Ordinary Differential Equations, Academic Press, 1982.
[26] S.-K. Oh, W. Pedrycz, S.-B. Roh, Genetically optimized fuzzy polynomial neural networks with fuzzy set-based polynomial neurons, Information Sciences 176 (2006) 3490–3519.
[27] A. Shortt, J. Keating, L. Monlinier, C. Pannell, Optical implementation of the Kak neural network, Information Sciences 171 (2005) 273–287.
[28] D.W. Tank, J.J. Hopfield, Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit, IEEE Transactions on Circuits and Systems 33 (1986) 533–541.
[29] P. Tseng, Global behaviour of a class of merit functions for the nonlinear complementarity problem, Journal of Optimization Theory and Applications 89 (1996) 17–37.
[30] L.T. Watson, Solving the nonlinear complementarity problem by a homotopy method, SIAM Journal on Control and Optimization 17 (1979) 36–46.
[31] Y. Xia, H. Leung, J. Wang, A projection neural network and its application to constrained optimization problems, IEEE Transactions on Circuits and Systems – I 49 (2002) 447–458.
[32] Y. Xia, H. Leung, J. Wang, A general projection neural network for solving monotone variational inequalities and related optimization problems, IEEE Transactions on Neural Networks 15 (2004) 318–328.
[33] Y. Xia, H. Leung, J. Wang, A recurrent neural network for solving nonlinear convex programs subject to linear constraints, IEEE Transactions on Neural Networks 16 (2005) 379–386.
[34] M. Yashtini, A. Malek, Solving complementarity and variational inequalities problems using neural networks, Applied Mathematics and Computation 190 (2007) 216–230.
[35] S.H. Zak, V. Upatising, S. Hui, Solving linear programming problems with neural networks: a comparative study, IEEE Transactions on Neural Networks 6 (1995) 94–104.
[36] G. Zhang, A neural network ensemble method with jittered training data for time series forecasting, Information Sciences 177 (2007) 5329–5340.