Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 807656, 18 pages
doi:10.1155/2012/807656

Research Article

A Recurrent Neural Network for Nonlinear Fractional Programming

Quan-Ju Zhang[1] and Xiao Qing Lu[2]

[1] *Management Department, City College of Dongguan University of Technology, Dongguan 523106, China*
[2] *Financial Department, City College of Dongguan University of Technology, Dongguan 523106, China*

Correspondence should be addressed to Quan-Ju Zhang, zhangquanju@sina.com

Received 13 April 2012; Accepted 7 July 2012

Academic Editor: Hai L. Liu

This paper presents a novel recurrent time continuous neural network model which performs nonlinear fractional optimization subject to interval constraints on each of the optimization variables. The network is proved to be complete in the sense that the set of optima of the objective function to be minimized with interval constraints coincides with the set of equilibria of the neural network. It is also shown that the network is primal and globally convergent in the sense that its trajectory cannot escape from the feasible region and will converge to an exact optimal solution for any initial point being chosen in the feasible interval region. Simulation results are given to demonstrate further the global convergence and good performance of the proposing neural network for nonlinear fractional programming problems with interval constraints.

1. Introduction

Compared with the well-known applications of nonlinear programming to various branches of human activity, especially to economics, the applications of fractional programming are less known until now. Of course, the linearity of a problem makes it easier to tackle and hence contributes its wide recognition. However, it is certain that not all real-life economic problems can be described by linear models and hence are not likely applications of linear programming. Fractional programming is a nonlinear programming method that has known increasing exposure recently and its importance in solving concrete problems is steadily increasing. Moreover, it is known that the nonlinear optimization models describe practical problems much better than the linear optimization models do.

The fractional programming problems are particularly useful in the solution of economic problems in which various activities use certain resources in various proportions,

while the objective is to optimize a certain indicator, usually the most favorable return-on-allocation ratio subject to the constraint imposed on the availability of goods. The detailed descriptions of these models can be found in Charnes et al. [1], Patkar [2], and Mjelde [3]. Besides the economic applications, it was found that the fractional programming problems also appeared in other domains, such as physics, information theory, game theory, and others. Nonlinear fractional programming problems are, of course, the dominant ones for their much widely applications, see Stancu-Minasian [4] in details.

As it is known, conventional algorithms are time consuming in solving optimization problems with large-scale variables and so new parallel and distributed algorithms are more competent then. Artificial neural networks (RNNs) governed by a system of differential equations can be implemented physically by designated hardware with integrated circuits and an optimization process with different specific purposes could be conducted in a truly parallel way. An overview and paradigm descriptions of various neural network models for tackling a great deal of optimization problems can be found in the book by Cichocki and Unbehauen [5]. Unlike most numerical algorithms, neural network approach can handle, as described in Hopfield's seminal work [6, 7], optimization process in real-time on line and hence to be the top choice.

Neural network models for optimization problems have been investigated intensively since the pioneer work of Wang et al., see [8–14]. Wang et al. proposed several different neural network models for solving convex programming problems [8], linear programming [9, 10], which, proved to be globally convergent to the problem's exact solutions. Kennedy and Chua [11] developed a neural network model for solving nonlinear programming problems where a penalty parameter needed to tune in the optimization process and hence only approximate solutions were generated. Xia and Wang [12] gave a general neural network model designing methodology which put together many gradient-based network models for solving the convex programming problems under this framework with globally convergent stability. Neural networks for the quadratic optimization and nonlinear optimization with interval constraints were developed by Bouzerdorm and Pattison [13] and Liang and Wang [14], respectively.

All these neural networks can be classified into the following three types: (1) the gradient-based models [8–10] and its extension [12]; (2) the penalty-function-based model [11]; (3) the projection based models [13, 14]. Among them the first was proved to have the global convergence [12]; the third quasi-convergence [8–10] only when the optimization problems are convex programming problems. The second could only be demonstrated to have local convergence [11] and more unfortunately, it might fail to find exact solutions, see [15] for a numerical example. Because of this, the penalty-function-based model has little applications in practice. As it is known, nonlinear fractional programming does not belong to convex optimization problems [4] and how to construct a good performance neural network model to solve this optimization problem becomes a challenge now since. Motivated by this idea, a promising recurrent continous-time neural network model is going to be proposed in the present paper. The proposing RNN model has the following two most important features. (1) The model is complete in the sense that the set of optima of the nonlinear fractional programming with interval constraints coincides with the set of equilibria of the proposing RNN model. (2) The RNN model is invariant with respect to the problem's feasible set and has the global convergence property in the sense that all the trajectories of the proposing network converge to the exact solution set for any initial point starting at the feasible interval region. These two properties demonstrate that the proposing network model is quite suitable for solving nonlinear fractional programming problems with interval constraints.

Remains of the paper are organized as follows. Section 2 formulates the optimization problem and Section 3 describes the construction of the proposing RNN model. Complete property and global convergence of the proposing model are discussed in Sections 4 and 5, respectively. Section 6 gives some typical application areas of the fractional programming. Illustrative examples with computational results are reported in Section 7 to demonstrate further the good performance of the proposing RNN model in solving the interval-constrained nonlinear fractional programming problems. Finally, Section 8 is a conclusion remark which presents a summary of the main results of the paper.

2. Problem Formulation

The study of the nonlinear fractional programming with interval constraints is motivated by the study of the following linear fractional interval programming:

$$\min\left\{ f(x) = c^T x + \frac{c_0}{d^T x} + d_0 : a \le x \le b \right\}, \tag{2.1}$$

where:

(i) $f(x) = c^T x + c_0 / d^T x + d_0$,

(ii) c, d are n dimensional column vectors,

(iii) c_0, d_0 are scalars,

(iv) superscript T denotes the transpose operator,

(v) $x = (x_1, x_2, \ldots, x_n)^T \in R^n$ is the decision vector,

(vi) $a = (a_1, a_2, \ldots, a_n)^T \in R^n, b = (b_1, b_2, \ldots, b_n)^T \in R^n$ are constant vectors with $a_i \le b_i$ $(i = 1, 2, \ldots, n)$.

It is assumed that the denominator of the objective function $f(x)$ maintains a constant sign on an open set O which contains the interval constraints $W = \{x : a \le x \le b\}$, say positive, that is,

$$d^T x + d_0 > 0, \quad \forall x \in O \supseteq W, \tag{2.2}$$

and that the function $f(x)$ does not reduce to a linear function, that is, $d^T x + d_0 \neq$ constant on $O \supseteq W$ and c, d are linearly independent. If $x^* \in W$ and $f(x) \ge f(x^*)$ for any $x \in W$, then x^* is called an optimal solution to the problem (2.1). The set of all solutions to problem (2.1) is denoted by Ω^*, that is, $\Omega^* = \{x^* \in W \mid f(x) \ge f(x^*), \forall x \in W\}$.

Studies on linear fractional interval programming replaced the constraints in programming (2.1) with $a \le Ax \le b$ commenced in a series number of paper by Charnes et al. [16–18]. Charnse and Cooper, see [16], employed the change of variable method and developed a solution algorithm for this programming by the duality theory of linear programming. A little later, in [17, 18], Charnse et al. gave a different method which transformed the fractional interval problem into an equivalent problem like (2.1) by using the generalized inverse of A, and the explicit solutions were followed then. Also, Bühler [19] transformed the problem into another equivalent one of the same format, to which

he associated a linear parametric program used to obtain solution for the original interval programming problem.

Accordingly, constraints $a \leq Ax \leq b$ can always be transformed into $a \leq x \leq b$ by change of variable method without changing the programming's format, see [13] for quadratic programming and [17] for linear fractional interval programming. So, it is necessary to pay our attention on problem (2.1) only. As the existing studies on problem (2.1), see [16–19], focused on the classical method which is time consuming in optimization computational aspects, it is sure that the neural network method should be the top choice to meet the real-time computation requirement. To reach this goal, the present paper is to construct a RNN model that is available both for solving nonlinear fractional interval programming and for linear fractional interval programming problem (2.1) as well.

Consider the following more general nonlinear fractional programming problem:

$$\min\left\{F(x) = \frac{g(x)}{h(x)} : a \leq x \leq b\right\}, \tag{2.3}$$

where $g(x), h(x)$ are continuously differentiable function defined on an open convex set $O \subseteq R^n$ which contains the problem's feasible set $W = \{x \mid a \leq x \leq b\}$ and x, a, b the same as in problem (2.1), see the previous (v)-(vi). Similarly, we suppose the objective function's dominator $g(x)$ always keeps a constant sign, say $g(x) > 0$. As the most fractional programming problems arising in real-life world associate a kind of generalized convex properties, we suppose the objective function $F(x)$ to be pseudoconvex over O. There are several sufficient conditions for the function $F(x) = g(x)/h(x)$ being pseudoconvex, two of which, see [20], are (1) g is convex and $g \geq 0$, while h concave and $h > 0$; (2) g is convex and $g \leq 0$, while h is convex and $h > 0$. It is easy to see that the problem (2.1) is a special case of problem (2.3).

We are going to state the neural network model which can be employed for solving problem (2.3) and so for problem (2.1) as well. Details are described in the coming section.

3. The Neural Network Model

Consider the following single-layered recurrent neural network whose state variable x is described by the differential equation:

$$\frac{dx}{dt} = -x + f_W(x - \nabla F(x)), \tag{3.1}$$

where ∇ is the gradient operator and $f_W : R^n \to W$ is the projection operator defined by

$$f_W(x) = \arg\min_{w \in W} \|x - w\|. \tag{3.2}$$

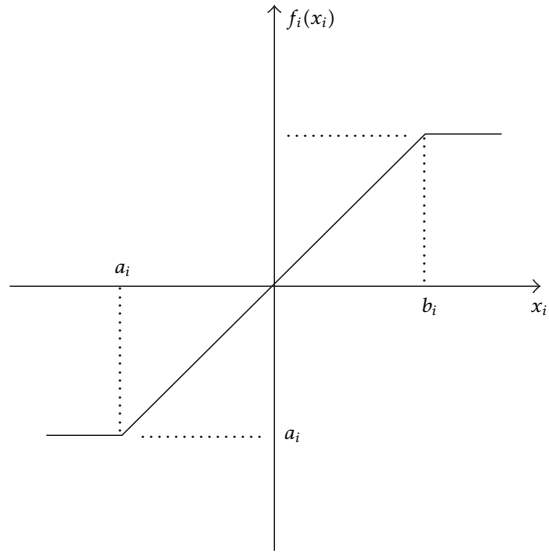

Figure 1: The activation function $f_i(x_i)$ of the neural network model (3.1).

For the interval constrained feasible set W, the operator f_W can be expressed explicitly as $f_W(x) = (f_{W_1}(x), \ldots, f_{W_n}(x))$ whose ith component is

$$f_{W_i}(x) \equiv f_{W_i}(x_i) = \begin{cases} a_i, & \\ x_i, & a_i \leq \\ b_i, & \end{cases} \begin{cases} x_i < a_i, \\ x_i \leq b_i, \\ x_i > b_i. \end{cases} \tag{3.3}$$

The activation function to one node of the neural network model (3.1) is the typical piece-wise linear $f_{W_i}(x_i)$ which is visibly illustrated in Figure 1.

To make a clear description of the proposed neural network model, we reformulate the compact matrix form (3.1) as the following component ones:

$$\frac{dx_i}{dt} = -x_i + f_{W_i}\left(x_i - \frac{\partial F(x)}{\partial x_i}\right), \quad i = 1, 2, \ldots, n. \tag{3.4}$$

When the RNN model (3.1) is employed to solve optimization problem (2.3), the initial state is required to be mapped into the feasible interval region W. That is, for any $x^0 = (x_1^0, x_2^0, \ldots, x_n^0) \in R^n$, the corresponding neural trajectory initial point should be chosen as $x(0) = f_W(x^0)$, or in the component form, $x_i(0) = f_{W_i}(x_i^0)$. The block functional diagram of the RNN model (3.4) is depicted in Figure 2.

Accordingly, the architecture of the proposed neural network model (3.4) is composed of n integrators, n processors for $F(x)$, $2n$ piece-wise linear activation functions, and $2n$ summers.

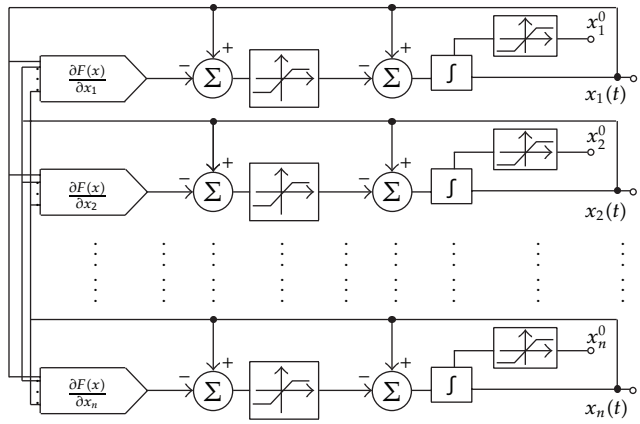

Figure 2: Functional block diagram of the neural network model (3.1).

Let the equilibrium state of the RNN model (3.1) be Ω^e which is defined by the following equation:

$$\Omega^e = \left\{ x^e \in R^n \mid x^e = f_W(x^e - \nabla F(x^e)) \right\} \subseteq W. \tag{3.5}$$

The relationship between the minimizer set Ω^* of problem (2.3) and the equilibrium set Ω^e is explored in the following section. It is guaranteed that the two sets coincide exactly and, this case is the most available expected one in the neural network model designs.

4. Complete Property

As proposed for binary-valued neural network model in [21], a neural network is said to be *regular* or *normal* if the set of minimizers of an energy function is a subset or superset of the set of the stable states of the neural network, respectively. If the two sets are the same, the neural network is said to be *complete*. The *regular* property implies the neural network's reliability and *normal* effectiveness, respectively, for the optimization process. *Complete* property means both reliability and effectiveness and it is the top choice in the neural network designing. Here, for the continuous-time RNN model (3.1), we say the model to be *regular*, *normal*, and *complete* respectively if three cases of $\Omega^* \subseteq \Omega^e$, $\Omega^e \subseteq \Omega^*$, and $\Omega^* = \Omega^e$ occur, respectively.

The complete property of the neural network (3.1) is stated in the following theorem.

Theorem 4.1. *The RNN model* (3.1) *is complete, that is,* $\Omega^* = \Omega^e$.

In order to prove Theorem 4.1, one needs the following lemmas.

Lemma 4.2. *Suppose that* x^* *is a solution to problem* (2.3)*, that is,*

$$F(x^*) = \min_{y \in W} F(y), \tag{4.1}$$

then x^* is a solution to the variational inequality:

$$x \in W : \left(\nabla F(x), y - x\right) \geq 0 \quad \text{for } y \in W. \tag{4.2}$$

Proof. See [22, Proposition 5.1]. □

Lemma 4.3. *Function $F(x) = c^T x + c_0 / d^T x + d_0$ defined in (2.3) is both pseudoconvex and pseudoconcave over W.*

Proof. See [20, Lemma 11.4.1]. □

Lemma 4.4. *Let $F(x) : R^n \to R$ be a differentiable pseudoconvex function on an open set $Y \subseteq R^n$, and $W \subseteq Y$ any given nonempty and convex set. Then x^* is an optimal solution to the problem of minimizing $F(x)$ subject to $x \in W$ if and only if $(x - x^*)^T \nabla F(x^*) \geq 0$ for all $x \in W$.*

Proof. See [4, Theorem 2.3.1(b)]. □

Now, we turn to the proof of Theorem 4.1: let $x^* = (x_1^*, x_2^*, \ldots, x_n^*)^T \in \Omega^*$, then $F(x) \geq F(x^*)$ for any $x \in W$ and hence, Lemma 4.2 means that x^* solves (4.2), that is,

$$x^* \in W : \left(y - x^*\right)^T \nabla F(x^*) \geq 0, \quad \forall y \in W, \tag{4.3}$$

which is equivalent to, see [23],

$$x^* = f_W(x^* - \nabla F(x^*)), \tag{4.4}$$

so, $x^* \in \Omega^e$. Thus, $\Omega^* \subseteq \Omega^e$.

Conversely, let $x^e = (x_1^e, x_2^e, \ldots, x_n^e)^T \in \Omega^e$, that is,

$$x^e = f_W(x^e - \nabla F(x^e)), \tag{4.5}$$

which, also see [23], means

$$x^e \in W : \left(y - x^e\right)^T \nabla F(x^e) \geq 0, \quad \forall y \in W. \tag{4.6}$$

Since the function $F(x)$ is pseudoconvex over W, see Lemma 4.3, it can be obtained by Lemma 4.4 that

$$F(x) \geq F(x^e), \quad \forall x \in W, \tag{4.7}$$

so, $x^e \in \Omega^*$. Thus, $\Omega^e \subseteq \Omega^*$. Therefore, the obtained results $\Omega^e \subseteq \Omega^*$ and $\Omega^* \subseteq \Omega^e$ lead the result of Theorem 4.1, $\Omega^* = \Omega^e$, to be true then.

5. Stability Analysis

First, it can be shown that the RNN model (3.1) has a solution trajectory which is global in the sense that its existence interval can be extended to ∞ on the right hand for any initial point in W.

The continuity of the right hand of (3.1) means, by Peano's local existence theorem, see [24], that there exists a solution $x(t; x^0)$ for $t \in [0, t_{\max})$ with any initial point $x_0 \in W$, here $t_{\max}$ is the maximal right hand point of the existence interval. The following lemma states that this $t_{\max}$ to be ∞.

Lemma 5.1. *The solution $x(t;\ x^0)$ of RNN model* (3.1) *with any initial point $x(0; x^0) = x^0 \in W$ is bounded and so, it can be extended to ∞.*

Proof. It is easy to check that the solution $x(t) = x(t; x^0)$ for $t \in [0, t_{\max})$ with initial condition $x(0; x^0) = x^0$ is given by

$$x(t) = e^{-t}x^0 + e^{-t}\int_0^t e^s f_W[x(s) - \nabla F(x(s))]ds. \tag{5.1}$$

Obviously, mapping f_W is bounded, that is $\|f_W\| \le K$ for some positive number $K > 0$, where $\|\cdot\|$ is the Euclidean 2 norm. It follows from (5.1) that

$$\begin{aligned} \|x(t)\| &\le e^{-t}\left\|x^0\right\| + e^{-t}K\int_0^t e^s ds \\ &\le e^{-t}\left\|x^0\right\| + K\left(1 - e^{-t}\right) \\ &\le \max\left\{\left\|x^0\right\|, K\right\}. \end{aligned} \tag{5.2}$$

Thus, solution $x(t)$ is bounded and so, by the extension theorem for ODEs, see [24], it can be concluded that $t_{\max} = \infty$ which completes the proof of this lemma.

Now, we are going to show another vital dynamical property which says the set W is positive invariant with respect to the RNN model (3.1). That is, any solution $x(t)$ starting from a point in W, for example, $x_0 \in W$, it will stay in W for all time t elapsing. Additionally, we can also prove that any solution starting from outside of W will either enter into the set W in finite time elapsing and hence stay in it for ever or approach it eventually. □

Theorem 5.2. *For the neural dynamical system* (3.1)*, the following two dynamical properties hold:*

(a) *W is a positive invariant set of the RNN model* (3.1)*;*

(b) *if $x_0 \notin W$, then, either $x(t)$ enters into W in finite time elapsing and hence stays in it for ever or $\rho(t) = \operatorname{dist}(x(t), W) \to 0$, as $t \to \infty$, where $\operatorname{dist}(x(t), W) = \inf_{y \in W}\|x - y\|$.*

Proof. Method to prove this theorem can be found in [14] and for the purpose of completeness and readability, here we give the whole proof as follows again.

Suppose that, for $i = 1, \ldots, n, W^i = \{x_i \in R \mid a_i \le x_i \le b_i\}$ and $x_i^0 = x_i(0;\ x^0) \in W^i$. We first prove that for all $i = 1, 2, \ldots, n$, the ith component $x_i(t) = x_i(t; x^0)$ belongs to W^i, that is, $x_i(t) \in W^i$ for all $t \ge 0$.

Let

$$t_i^* = \sup\left\{\widetilde{t} \mid x_i(t) \in W^i,\ \forall t \in \left[0, \widetilde{t}\right]\right\} \ge 0. \tag{5.3}$$

We show by a contradiction that $t_i^* = +\infty$. Suppose $t_i^* < \infty$, then $x_i(t) \in W^i$ for $t \in [0, t_i^*]$ and $x_i(t) \notin W^i$ for $t \in (t_i^*, t_i^* + \delta)$ where δ being a positive number. With no loss of generality, we assume that

$$x_i(t) < a_i, \quad \forall t \in \left(t_i^*, t_i^* + \delta\right). \tag{5.4}$$

The proof for $x_i(t) > b_i, \forall t \in (t_i^*, t_i^* + \delta)$ is similar. By the definition of f_{W_i}, the RNN model (3.4) and the assumption (5.4), it follows that

$$\frac{dx_i(t)}{dt} \ge -x_i(t) + a_i > 0, \quad \forall t \in \left(t_i^*, t_i^* + \delta\right). \tag{5.5}$$

So, $x_i(t)$ is strictly increasing in $t \in (t_i^*, t_i^* + \delta)$ and hence

$$x_i(t) > x_i\left(t_i^*\right), \quad \forall t \in \left(t_i^*, t_i^* + \delta\right). \tag{5.6}$$

Noting that $x_i(t) \in W$ for $t \in [0, t_i^*]$ and assumption (5.4) implies $x_i(t_i^*) = a_i$, so, by (5.6), we get

$$x_i(t) > a_i, \quad \forall t \in \left(t_i^*, t_i^* + \delta\right). \tag{5.7}$$

This is in contradiction with the assumption (5.4). So, $t_i^* = +\infty$, that is $x_i(t) \in W^i$ for all $t \ge 0$. This means W is positive invariant and hence (a) is guaranteed.

Second, for some i, suppose $x_i^0 = x_i(0;\ x^0) \notin W^i$. If there is a $t_i^* > 0$ such that $x(t_i^*) \in W^i$, then, according to (a), $x_i(t)$ will stay in W^i for all $t \ge t_i^*$. That is $x_i(t)$ will enter into W_i. Conversely, for all $t \ge 0$, suppose $x_i(t) \notin W^i$. Without loss of generality, we assume that $x_i(t) < a_i$. It can be guaranteed by a contradiction that $\sup\{x_i(t) \mid t \ge 0\} = a_i$. If it is not so, note that $x_i(t) < a_i$, then $\sup\{x_i(t) \mid t \ge 0\} = m < a_i$. It can be followed by (3.4) that

$$\frac{dx_i(t)}{dt} \ge -m + a_i = \delta > 0. \tag{5.8}$$

Integrating (5.8) gives us

$$x_i(t) \ge \delta t + x_i^0, \quad t > 0, \tag{5.9}$$

which is a contradiction because of $x_i(t) < a_i$. Thus, we obtain $\sup\{x_i(t) \mid t \ge 0\} = a_i$. This and the previous argument show that, for $x_0 \notin W$, either $x(t)$ enters into W in finite time and hence stays in it for ever or $\rho(t) = \operatorname{dist}(x(t), W) \to 0$, as $t \to \infty$.

We can now explore the global convergence of the neural network model (3.1). To proceed, we need an inequality result about the projection operator f_W and the definition of convergence for a neural network. □

Definition 5.3. Let $x(t)$ be a solution of system $\dot{x} = F(x)$. The system is said to be globally convergent to a set X with respect to set W if every solution $x(t)$ starting at W satisfies

$$\rho(x(t), X) \longrightarrow 0, \quad \text{as } t \longrightarrow \infty, \tag{5.10}$$

here $\rho(x(t), X) = \inf_{y \in X} \|x - y\|$ and $x(0) = x_0 \in W$.

Definition 5.4. The neural network (3.1) is said to be globally convergent to a set X with respect to set W if the corresponding dynamical system is so.

Lemma 5.5. *For all $v \in R^n$ and all $u \in W$*

$$\left(v - f_W(v)\right)^T \left(f_W(v) - u\right) \geq 0. \tag{5.11}$$

Proof. See [22, pp. 9-10]. □

Theorem 5.6. *The neural network* (3.1) *is globally convergent to the solution set Ω^* with respect to set W.*

Proof. From Lemma 5.5, we know that

$$\left(v - f_W(v)\right)^T \left(f_W(v) - u\right) \geq 0, \quad v \in R^n,\ u \in W. \tag{5.12}$$

Let $v = x - \nabla F(x)$ and $u = x$, then

$$\left(x - \nabla F(x) - f_W(x - \nabla F(x))\right)^T \left(f_W(x - \nabla F(x)) - x\right) \geq 0, \tag{5.13}$$

that is,

$$(\nabla F(x))^T \left\{f_W(x - \nabla F(x)) - x\right\} \leq -\left\|f_W(x - \nabla F(x)) - x\right\|^2. \tag{5.14}$$

Define an energy function $F(x)$, then, differentiating this function along the solution $x(t)$ of (3.1) gives us

$$\begin{aligned} \frac{dF(x(t))}{dt} &= (\nabla F(x))^T \frac{dx}{dt} \\ &= (\nabla F(x))^T \left\{f_W(x - \nabla F(x)) - x\right\}. \end{aligned} \tag{5.15}$$

According to (5.14), it follows that

$$\frac{dF(x(t))}{dt} \leq -\left\|f_W(x - \nabla F(x)) - x\right\|^2 \leq 0. \tag{5.16}$$

It means the energy of $F(x)$ is decreasing along any trajectory of (3.1). By Lemma 5.1, we know the solution $x(t)$ is bounded. So, $F(x)$ is a Liapunov function to system (3.1). Therefore, by LaSalle's invariant principle [25], it follows that all trajectories of (3.1) starting at W will converge to the largest invariant subset Σ of set E like

$$E = \left\{ x \mid \frac{dF}{dt} = 0 \right\}. \tag{5.17}$$

However, it can be guaranteed from (5.16) that $dF/dt = 0$ only if $f_W(x - \nabla F(x)) - x = 0$, which means that x must be an equilibrium of (3.1) or, $x \in \Omega$. Thus, Ω is the convergent set for all trajectories of neural network (3.1) starting at W. Noting that Theorem 4.1 tells us that $\Omega^* = \Omega$ and hence, Theorem 5.6 is proved to be true then.

Up to now, we have demonstrated that the proposed neural network (3.1) is a promising neural network model both in implementable construction sense and in theoretic convergence sense for solving nonlinear fractional programming problems and linear fractional programming problems with bound constraints. Certainly, it is also important to simulate the network's effectiveness by numerical experiment to test its performance in practice. In next section, we will focus our attention on handling illustrative examples to reach this goal. □

6. Typical Application Problems

This section contributes to some typical problems from various branches of human activity, especially in economics and engineering, that can be formulated as fractional programming. We choose three problems from information theory, optical processing of information and macroeconomic planning to identify the various applications of fractional programming.

6.1. Information Theory

For calculating maximum transmission rate in an information channel Meister and Oettli [26], Aggarwal and Sharma [27] employed the fractional programming described briefly as follows.

Consider a constant and discrete transmission channel with m input symbols and n output symbols, characterized by a transition matrix $P = (p_{ij}), i = 1, \ldots, m, p_{ij} \geq 0, \sum_i p_{ij} = 1$, where p_{ij} represents the probability of getting the symbol i at the output subject to the constraint that the input symbol was j. The probability distribution function of the inputs is denoted by $x = (x_j)$, and obviously, $x_j \geq 0, \sum_j x_j = 1$.

Define the transmission rate of the channel as:

$$T(x) = \frac{\sum_i \sum_j x_j p_{ij} \log\left(p_{ij} / \sum_k x_k p_{ik}\right)}{\sum_j t_j x_j}. \tag{6.1}$$

The relative capacity of the channel is defined by the maximum of $T(x)$, and we get the following fractional programming problem:

$$G = \max_{x \in X} \left\{ T(x) \mid x_j \geq 0, \sum_j x_j = 1 \right\}. \tag{6.2}$$

With the notations:

$$c_j = \sum_i p_{ij} \log p_{ij}, \quad c = (c_j), \qquad y_i = \sum_j p_{ij} x_j, \quad y = (y_j), \qquad z = (x_j, y_i), \tag{6.3}$$

problem (6.2) becomes

$$\max \left\{ T(z) = \frac{c'x - y' \log y}{t'x} \mid x_j \geq 0, \sum_j x_j = 1, y_i = \sum_j p_{ij} x_j \right\}. \tag{6.4}$$

6.2. Optical Processing of Information

In some physics problems, fractional programming can also be applied. In spectral filters for the detection of quadratic law for infrared radiation, the problem of maximizing the signal-to-noise ration appears. This means to maximize the filter function

$$\phi(x) = \frac{(a'x)^2}{x'Bx + \beta} \tag{6.5}$$

on the domain $S = \{x \in R^n, 0 \leq x_i \leq 1, i = 1, \ldots, n\}$ in which a and β are strict positive vector, and constant, respectively, B is a symmetric and positive definite matrix, $a'x$ represents the input signal, and $x'Bx + \beta$ represents the variance in the background signal. The domain of the feasible solutions S illustrates the fact that the filter cannot transmit more than 100% and less than 0% of the total energy. The optical filtering problems are very important in today's information technology, especially in coherent light applications, and optically based computers have already been built.

6.3. Macroeconomic Planning

One of the most significant applications of fractional programming is that of dynamic modeling of macroeconomic planning using the input-output method. Let $Y(t)$ be the national income created in year t. Obviously, $Y(t) = \sum_i Y_i(t)$. If we denote by $C_{ik}(t)$ the consumption, in branch k, of goods of type i (that were created in branch i) and by I_{ik} the part of the national income created in branch i and allocated to investment in branch k, then the following repartition equation applies to the national income created in branch i:

$$Y_i(t) = \sum_k (C_{ik}(t) + I_{ik}(t)). \tag{6.6}$$

The increase of the national income in branch k is function of the investment made in this branch

$$\Delta Y_i(t) = Y_k(t+1) - Y_k(t) = b_k I_k + b_{k0}, \tag{6.7}$$

where $I_k = \sum_i I_{ik}$.

In these conditions, the macroeconomic planning leads to maximize the increase rate of the national income:

$$\frac{\Delta Y}{Y} = \frac{\sum_k (b_k I_k(t) + b_{k0})}{\sum_i \sum_k (C_{ik}(t) + I_{ik}(t))} \tag{6.8}$$

subject to the constraints $C_k(t) \geq \max(\overline{C}_k, C_k(0))$, where $C_k(t) = \sum_i C_{ik}(t), I_k(0) \leq I_k(t) \leq I_k(\max)$, and $\overline{C}_k$ represents minimum consumption attributed to branch k whereas $I_k(\max)$ is the maximum level of investments for branch k.

7. Illustrative Examples

We give some computational examples as simulation experiment to show the proposed network's good performance.

Example 7.1. Consider the following linear fractional programming:

$$\begin{aligned} \min \quad & F(x) = \frac{x_1 + x_2 + 1}{2x_1 - x_2 + 3}, \\ \text{s.t.} \quad & 0 \leq x_1 \leq 2, \\ & 0 \leq x_2 \leq 2 \end{aligned} \tag{7.1}$$

This problem has an exact solution $x^* = [0,0]^T$ with the optimal value $F(x^*) = 1/3$. The gradient of $F(x)$ can be expressed as

$$\nabla F(x) = \begin{pmatrix} \dfrac{-3x_2 + 1}{(2x_1 - x_2 + 3)^2} \\ \dfrac{3x_1 + 4}{(2x_1 - x_2 + 3)^2} \end{pmatrix}. \tag{7.2}$$

Define

$$z_i = x_i - \nabla F_{x_i}, \quad i = 1,2 \tag{7.3}$$

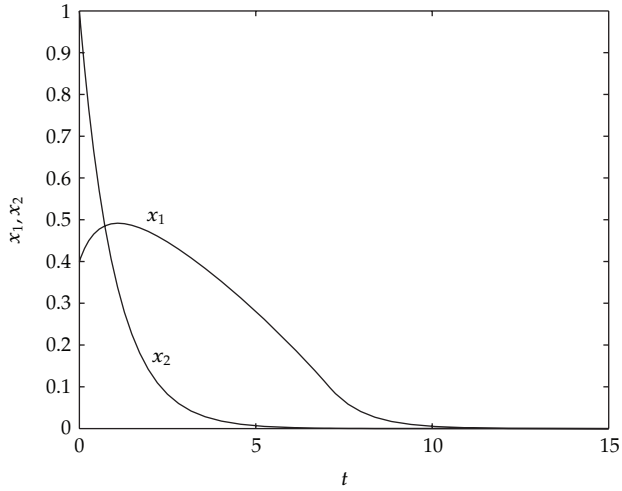

Figure 3: Transient behaviors of neural trajectories x_1, x_2 from the inside of W.

and pay attention to (7.2), we get

$$z_1 = \frac{4x_1^3 - 4x_1^2x_2 - 2x_1x_2^2 - 6x_1x_2 + 12x_1^2 + 9x_1 + 3x_2 - 1}{(2x_1 - x_2 + 3)^2},$$
$$z_2 = \frac{x_2^3 - 4x_2^2x_1 + 12x_2x_1 + 4x_2x_1^2 - 6x_2^2 + 9x_2 - 3x_1 - 4}{(2x_1 - x_2 + 3)^2}. \tag{7.4}$$

The dynamical systems are given by

$$\frac{dx_1}{dt} = \begin{cases} -x_1, & \begin{cases} z_1 < 0, \\ z_1 \le 2, \\ z_1 > 2, \end{cases} \\ -x_1 + z_1, & 0 \le \\ -x_1 + 2, & \end{cases}$$
$$\frac{dx_2}{dt} = \begin{cases} -x_2, & \begin{cases} z_2 < 0, \\ z_2 \le 2, \\ z_2 > 2. \end{cases} \\ -x_2 + z_2, & 0 \le \\ -x_2 + 2, & \end{cases} \tag{7.5}$$

Various combinations of (7.5) formulate the proposed neural network model (3.1) to this problem. Conducted on MATLAB 7.0., by ODE 23 solver, the simulation results are carried out and the transient behaviors of the neural trajectories x_1, x_2 starting at $x_0 = [0.4, 1]^T$, which is *in* the feasible region W, are shown in Figure 3. It can be seen visibly from the figure that the proposed neural network converges to the exact solution very soon.

Also, according to (b) of Theorem 5.2, the solution may be searched from outside of the feasible region. Figure 4 shows this by presenting how the solution of this problem is located by the proposed neural trajectories from the initial point $x_0 = [0.5, 3]^T$ which is *not in* W.

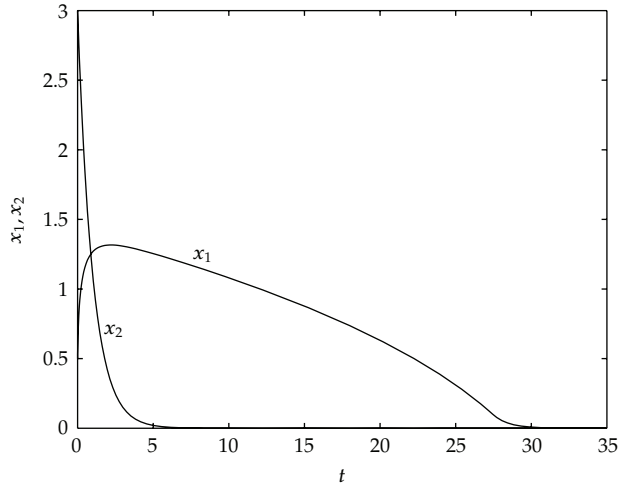

Figure 4: Transient behaviors of neural trajectories x_1, x_2 from the outside of W.

Example 7.2. Consider the following nonlinear fractional programming:

$$\begin{aligned} \min \quad & F(x) = \frac{x_1}{15 - x_1^2 - x_2^2}, \\ \text{s.t.} \quad & 1 \le x_1 \le 2, \\ & 1 \le x_2 \le 2. \end{aligned} \tag{7.6}$$

This problem has an exact solution $x^* = [1,1]^T$ with the optimal value $F(x^*) = 1/13$. The gradient of $F(x)$ can be expressed as

$$\nabla F(x) = \begin{pmatrix} \dfrac{15 + x_1^2 - x_2^2}{\left(15 - x_1^2 - x_2^2\right)^2} \\ \dfrac{2x_1x_2}{\left(15 - x_1^2 - x_2^2\right)^2} \end{pmatrix}. \tag{7.7}$$

Define

$$z_i = x_i - \nabla F_{x_i}, \quad i = 1,2 \tag{7.8}$$

and pay attention to (7.7), we get

$$\begin{aligned} z_1 &= \frac{x_1^5 + x_1x_2^4 - 30x_1^3 - 30x_1x_2^2 + 2x_1^3x_2^2 - x_1^2 + x_2^2 + 225x_1 - 15}{\left(15 - x_1^2 - x_2^2\right)^2}, \\ z_2 &= \frac{x_1^4x_2 + x_2^5 - 30x_1^2x_2 - 30x_2^3 + 2x_1^2x_2^3 + 225x_2 - 2x_1x_2}{\left(15 - x_1^2 - x_2^2\right)^2}. \end{aligned} \tag{7.9}$$

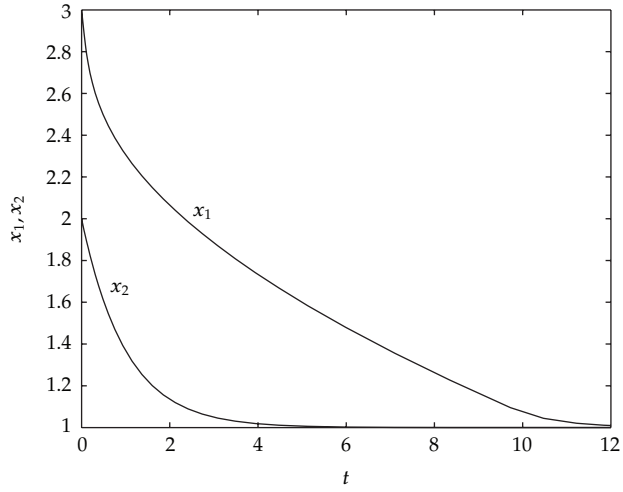

Figure 5: Transient behaviors of neural trajectories $x_1,\ x_2$ from the inside of W.

The dynamical systems are given by

$$\frac{dx_1}{dt} = \begin{cases} -x_1 + 1, & \\ -x_1 + z_1, & 1 \leq \\ -x_1 + 2, & \end{cases} \begin{cases} z_1 < 1, \\ z_1 \leq 2, \\ z_1 > 2, \end{cases}$$
$$\frac{dx_2}{dt} = \begin{cases} -x_2 + 1, & \\ -x_2 + z_2, & 1 \leq \\ -x_2 + 2, & \end{cases} \begin{cases} z_2 < 1, \\ z_2 \leq 2, \\ z_2 > 2. \end{cases} \tag{7.10}$$

Similarly, conducted on MATLAB 7.0., by ODE 23 solver, the transient behaviors of the neural trajectories $x_1,\ x_2$ from *inside* of the feasible region W, here $x_0 = (2, 3)$, are depicted in Figure 5 which shows the rapid convergence of the proposed neural network.

Figure 6 presents a trajectory from outside of W, here $(4, 3)$, it can be seen clearly from this that the solution of this problem is searched by the proposed neural trajectory soon.

8. Conclusions

In this paper, we have proposed a neural network model for solving nonlinear fractional programming problems with interval constraints. The network is governed by a system of differential equations with a projection method.

The stability of the proposed neural network has been demonstrated to have global convergence with respect to the problem's feasible set. As it is known, the existing neural network models with penalty function method for solving nonlinear programming problems may fail to find the exact solution of the problems. The new model has overcome this stability defect appearing in all penalty-function-based models.

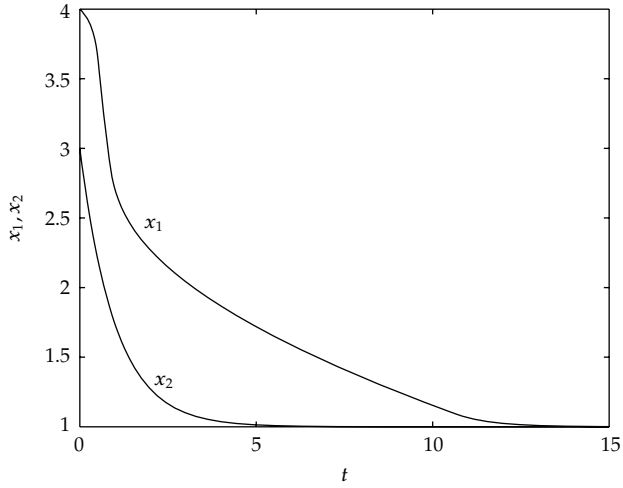

Figure 6: Transient behaviors of neural trajectories x_1, x_2 from the outside of W.

Certainly, the network presented here can perform well in the sense of real-time computation which, in the time elapsing sense, is also superior to the classical algorithms. Finally, numerical simulation results demonstrate further that the new model can act both effectively and reliably on the purpose of locating the involved problem's solutions.

Acknowledgment

The research is supported by Grant 2011108102056 from Science and Technology Bureau of Dong Guan, Guang Dong, China.

References

[1] A. Charnes, W. W. Cooper, and E. Rhodes, "Measuring the efficiency of decision making units," *European Journal of Operational Research*, vol. 2, no. 6, pp. 429–444, 1978.

[2] V. N. Patkar, "Fractional programming models for sharing of urban development responsabilities," *Nagarlok*, vol. 22, no. 1, pp. 88–94, 1990.

[3] K. M. Mjelde, "Fractional resource allocation with S-shaped return functions," *The Journal of the Operational Research Society*, vol. 34, no. 7, pp. 627–632, 1983.

[4] I. M. Stancu-Minasian, *Fractional Programming, Theory, Methods and Applications*, Kluwer Academic Publishers, Dodrecht, The Netherlands, 1992.

[5] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, John Wiley & Sons, New York, NY, USA, 1993.

[6] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 81, no. 10, pp. 3088–3092, 1984.

[7] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisons in optimization problems," *Biological Cybernetics*, vol. 52, no. 3, pp. 141–152, 1985.

[8] J. Wang, "A deterministic annealing neural network for convex programming," *Neural Networks*, vol. 7, no. 4, pp. 629–641, 1994.

[9] J. Wang and V. Chankong, "Recurrent neural networks for linear programming: analysis and design principles," *Computers and Operations Research*, vol. 19, no. 3-4, pp. 297–311, 1992.

[10] J. Wang, "Analysis and design of a recurrent neural network for linear programming," *IEEE Transactions on Circuits and Systems I*, vol. 40, no. 9, pp. 613–618, 1993.
[11] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Transaction on Circuits and Systems*, vol. 35, no. 5, pp. 554–562, 1988.
[12] Y. S. Xia and J. Wang, "A general methodology for designing globally convergent optimization neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1331–1343, 1998.
[13] A. Bouzerdoum and T. R. Pattison, "Neural network for quadratic optimization with bound constraints," *IEEE Transactions on Neural Networks*, vol. 4, no. 2, pp. 293–304, 1993.
[14] X. B. Liang and J. Wang, "A recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints," *IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1251–1262, 2000.
[15] Y. S. Xia, H. Leung, and J. Wang, "A projection neural network and its application to constrained optimization problems," *IEEE Transactions on Circuits and Systems I*, vol. 49, no. 4, pp. 447–458, 2002.
[16] A. Charnes and W. W. Cooper, "An explicit general solution in linear fractional programming," *Naval Research Logistics Quarterly*, vol. 20, pp. 449–467, 1973.
[17] A. Charnes, D. Granot, and F. Granot, "A note on explicit solution in linear fractional programming," *Naval Research Logistics Quarterly*, vol. 23, no. 1, pp. 161–167, 1976.
[18] A. Charnes, D. Granot, and F. Granot, "An algorithm for solving general fractional interval programming problems," *Naval Research Logistics Quarterly*, vol. 23, no. 1, pp. 67–84, 1976.
[19] W. Bühler, "A note on fractional interval programming," *Zeitschrift für Operations Research*, vol. 19, no. 1, pp. 29–36, 1975.
[20] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming, Theory and Algorithms*, John Wiley & Sons, New York, NY, USA, 1979.
[21] Z. B. Xu, G. Q. Hu, and C. P. Kwong, "Asymmetric Hopfield-type networks: theory and applications," *Neural Networks*, vol. 9, no. 3, pp. 483–501, 1996.
[22] D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, Academic Press, New York, NY, USA, 1980.
[23] B. C. Eaves, "On the basic theorem of complementarity," *Mathematical Programming*, vol. 1, no. 1, pp. 68–75, 1971.
[24] J. K. Hale, *Ordinary Differential Equations*, John Wiley & Sons, New York, NY, USA, 1993.
[25] J. P. LaSalle, "Stability theory for ordinary differential equations," *Journal of Differential Equations*, vol. 4, pp. 57–65, 1968.
[26] B. Meister and W. Oettli, "On the capacity of a discrete, constant channel," *Information and Control*, vol. 11, no. 3, pp. 341–351, 1967.
[27] S. P. Aggarwal and I. C. Sharma, "Maximization of the transmission rate of a discrete, constant channel," *Mathematical Methods of Operations Research*, vol. 14, no. 1, pp. 152–155, 1970.