Iran University of Science &Technology
3TH INTERNATIONAL CONFERENCE ON COMBINATORICS, CRYPTOGRAPHY AND COMPUTATION

Proceedings of the 3nd International Conference on Combinatorics, Cryptography and Computation (I4C2018)

A neurodynamic model for shortest path problem

Alireza Shojaeifard, Amin Mansoori[1], Ali Nakhaei Amrodi
Department of Mathematics and Statistics, Imam Hossein Comprehensive University, Tehran, Iran.
ashojaeifard@ihu.ac.ir; am.ma7676@yahoo.com; kpnakhaei@ihu.ac.ir

ABSTRACT

In this paper, a representation of a recurrent neural network to solve the shortest path (SP) problem is given. The motivation of the paper is to design a new effective one-layer structure recurrent neural network model for solving the SP. Moreover, we show that the proposed recurrent neural network is globally exponentially stable. Finally, the numerical examples are discussed to demonstrate the performance of our proposed approach.

KEYWORDS: Shortest path problem, recurrent neural network, linear optimization problem, Stability in the sense of *Lyapunov*.

1 INTRODUCTION

The shortest path problem (SPP) 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, economic, etc (see [1, 2, 3, 4]).

Let $G = (V, E)$ be a graph, where V is the set of vertices and E is the set of edges. A path between two nodes is an alternating sequence of vertices and edges starting and ending with the vertices. The distance (cost) of a path is the sum of the eights of the edges on the path. However, since there can be more than one path between any two vertices, the problem of finding a path with the minimum cost between two specified vertices makes sense. This is the so-called shortest path problem.

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. The main idea of the neural network approach for a mathematical programming problem is to establish an energy function (nonnegative) and a dynamic system which is a representation of an artificial neural network. An important requirement is that the energy function decreases monotonically as the

[1] Speaker

dynamic system approaches an equilibrium point. The main advantage of the neural network approach to optimization is that the nature of the dynamic solution procedure is inherently parallel and distributed.

Recently, neural networks for solving crisp and fuzzy optimization problem have been rather extensively studied and some important results have also been obtained [5, 6, 7]. For example, Xia and Wang [7] proposed a bi-projection neural network for solving a class of constrained quadratic optimization problems, Effati et al. [5] gave an efficient projection neural network for solving bilinear programming problems. Also, Eshaghnezhad et al. [6] gave a neurodynamic model for solving nonlinear pseudo-monotone projection equation.

Motivated by the above discussions, in the present paper, we design a new effective one-layer structure neural network model for solving the FSP problem. As far as we know, there is not an attempt for solving the FSP by recurrent neural network. As mentioned above, there are several neural network models to solve crisp mathematical programming problems. This methodology is going to study the possibility of extending the existing neural network model to solve the SP problem. Additionally, for showing the stability of the proposed neural network model, we give a Lyapunov function for dynamical system. As we mentioned, the main advantage of the neural network approach is that the nature of the dynamic solution procedure is inherently parallel and distributed.

The rest of the paper is organized as follows. Section 2 contains the SP problem. We present the neural network model in Section 3. In Section 4, the stability condition and global convergence for the proposed neural network are discussed. Illustrative examples are given in Section 5 to show the validity and applicability of our method. Finally, Section 6 states the conclusions.

2 SHORTEST PATH PROBLEM

In this section a mathematical formulation of the FSPP is presented.

Let us consider a directed network G = (V,E), where V is the set of vertices and E is the set of arcs. Each arc is denoted by an ordered pair (i,j), where $i, j \in V$. It is supposed that there is only one directed arc (i,j) from i to j. Let the node 1 be the source node and let us assume t as the destination node. We define a path $p_{\{ij\}}$ as a sequence $p_{\{ij\}} = \{i, (i, i_1), i_1, \ldots, i_k, (i_k, j), j\}$ of alternating nodes and arcs. The existence of at least one path p_{si} in G = (V,E) is assumed for every node i ∊ V\{s}.

Let $c_{\{ij\}}$ denotes the number associated with the arc (i,j), corresponding to the length necessary to traverse (i ,j) from i to j . In real problems , the length corresponds to the cost , the time , the distance, etc. Then, the SPP is formulated as the following linear programming problem:

$$\text{Min } f = \sum_{(i,j)\in E} c_{\{ij\}} x_{\{ij\}}$$

$$\text{s.t.} \quad \sum_j^r x_{\{ij\}} - \sum_j^r x_{\{ji\}} = \begin{cases} 1, & if\ i = 1 \\ 0, & if\ i \neq 1, r \\ -1, & if\ i = r \end{cases} \quad (i=1,\ldots, r) \tag{1}$$

$$x_{\{ij\}} = 0 \text{ or } 1, \quad \text{for } (i,j)\in E,$$

where r is the number of nodes. Also, $x_{\{ij\}}$ denotes the decision variable associated with the edge from vertices i to j as defined below:

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

Because of the total unimodality property of the constraint coefficient matrix defined in (1) [3], the integrality constraint in the shortest path problem formulation can be equivalently replaced with the non-negativity constraint, if the shortest path is unique. In other words, the optimal solutions of the equivalent linear programming problem are composed of zero and one integers if a unique optimum exists [3]. The equivalent linear programming problem based on the simplified edge path representation can be described as follows:

$$\text{Min } f = \sum_{i=1}^{r} \sum_{\substack{j=1 \\ j\neq i}} c_{\{ij\}} x_{\{ij\}}$$

s.t. $\sum_{\substack{k=1\\k\neq i}}^{r} x_{\{ik\}} - \sum_{\substack{l=2\\l\neq i}}^{r} x_{\{li\}} = \delta_{\{i1\}} - \delta_{\{ir\}}$ (i=1,…, r) (2)

$x_{\{ij\}} \geq 0, \quad i \neq j, i = 1, \dots, r-1, j = 1, \dots, r,$

where $\delta_{\{ij\}}$ 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 linear programming problem as follows:

Max $y_r - y_1$

s.t. $y_j - y\_i \leq c\_\{ij\}, i \neq j$ (i,j=1,…, r) (3)

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 [3]. 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 [3]: 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 $y_j - y_i = c_{\{ij\}}$, and $i, j = 1, 2, \dots, r$.

2- $x_{\{ij\}} = 0$ implies $y_j - y_i \leq c_{\{ij\}}$, and $i, j = 1,2, \dots, r$.

From the above analysis, it is seen that the shortest path problem formulation based on the edge path representation, is a linear programming problem. Thus, in next section we consider the general form of the linear programming problem and propose a recurrent neural network to solve it.

3 NEURAL NETWORK MODEL

Consider the following linear programming problem:

Min $c^T x$

s.t. $Ax = b$ (4)

$x \geq 0,$

where $x, c \epsilon R$, $A \epsilon R^{m\times n}$, and $b \in R^m$. Also, we can rewrite the problem (4) as follow:

Min $c^T x$

s.t. $Ax \leq b$ (5)

$-Ax \leq -b$

$x \geq 0.$

Finally, the problem (5) can be summarized as:

Min $c^T x$

s.t. $Dx \leq d,$ (٦)

where

$$D = \begin{bmatrix} A \\ -A \\ -I \end{bmatrix}, \quad d = \begin{bmatrix} b \\ -b \\ 0 \end{bmatrix}.$$

The KKT condition of the problem (5) can be constructed as the following:

$$KKT\ conditions: \begin{cases} y \geq 0, \ \ Dx \leq d, \\ y^T(Dx - d) = 0, \\ c + y^T A = 0. \end{cases} \quad (7)$$

where y is the Lagrange multiplier. By the KKT condition we see that (x^*, y^*) is an optimal solution (6), if and only if, satisfies the KKT conditions (7).

Now, our aim is to construct a continuous time dynamical system that will settle down to the KKT point of linear programming problem (6). At first we need some requirements.

The non-linear complementarity problem (NCP) consists in finding a vector $x \in R^n$ such that:

$x \geq 0, \ F(x) \geq 0, \ < x, F(x) >= 0,$

where $F: R^n \to R^n$ is a continuously differentiable function [8]. One of the most popular approaches for solving the non-linear complementarity problem, is to reformulate this problem as a system of non-linear equations (see [9]). A popular NCP function is the min function [10], which is defined as:

$$\varphi_{\min}(a, b) = \min\{a, b\}.$$

Also, the $\varphi_{\min}(a, b)$ can be shown with different representation as follow:

$\varphi_{\min}(a, b) = \min\{a, b\} = \frac{a+b-|a-b|}{2}$.

The perturbed min function is also given by:

$$\varphi_{\min}^{\varepsilon}(a, b) = \frac{a + b - |a - b + \varepsilon|}{2}, \qquad \varepsilon \to 0.$$

Now, applying the definition of NCP function, we can easily verify that the KKT conditions (6) are equivalent to the following unconstrained minimization problem:

$$\min \Psi(y) = \frac{1}{2}\|\phi(y)\|^2, \tag{7}$$

where

$$\phi(y) = \begin{bmatrix} c + y^T A \\ \varphi_{\min}^{\varepsilon}(y, -Dx + d) \end{bmatrix}$$

It is clear that $\Psi(y)$ is a merit function for the KKT conditions (6). The merit function $\Psi(y)$ in (7) is continuously differentiable for all y [10]. We may describe the neural network model corresponding to (4) and its dual by the following non-linear dynamical system:

$$\frac{dy}{dt} = -\gamma\nabla\Psi(y), \quad y(t_0) = y_0, \tag{8}$$

where $\gamma > 0$ is the convergence rate. An indication on how the neural network model (8) can be implemented on hardware is provided in Figure 1.

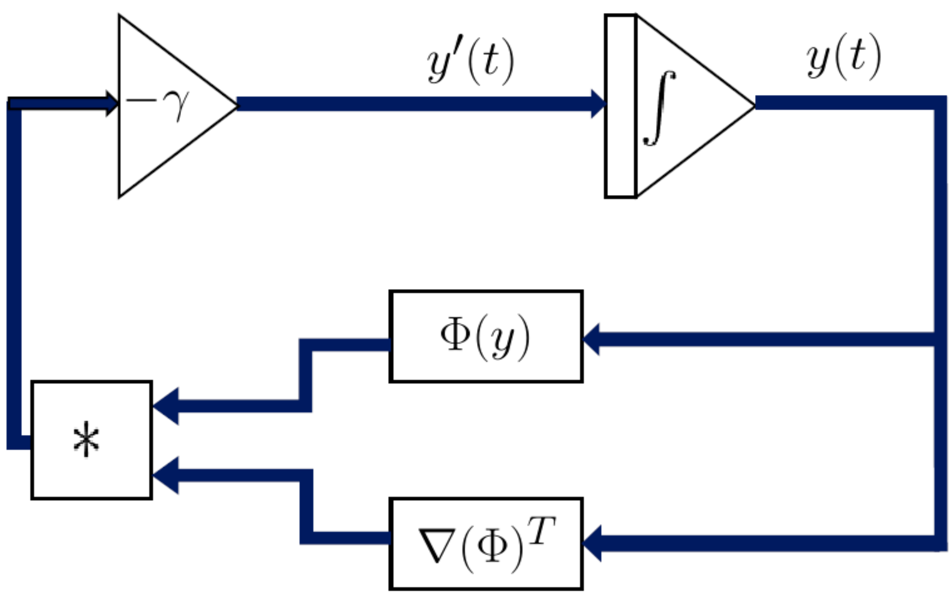

Figure. 1. *A simplified block diagram for the neural network model (8).*

4 STABILITY AND CONVERGENCE ANALYSIS

In this section, we show that the neural network model whose dynamics is described by the differential equation (8) is stable.

Theorem: $z^* = (x^*, y^*)$ is the equilibrium point of the neural network model (8), if and only if, x^* is a KKT point of the problem (4).

Lemma: Right hand side of (8) is Lipschitzian and differentiable.

Lemma: The equilibrium point of the proposed neural network model (8) is unique.

Theorem: The proposed neural network model (8) is globally stable in the sense of Lyapunov and is globally convergent to the optimal solution of the problem (4).

5 NUMERICAL EXAMPLE

In this section, we present an illustrative examples in order to demonstrate the efficiency, applicability, and the performance of the neural network model. The codes are developed using interpretive computer language MATLAB (ode45) and the calculations are implemented on a machine with Intel core 7 Duo processor 2 GHz and 4 GB RAM.

Example 1: Let us consider an acyclic network that possesses an arc with a negative cost in networks with such a characteristic as shown in Figure 1. Also, the arc lengths are displayed in Table 1.

Table 1. Arc lengths of the network in Example 1.

Source node	Destination node	Arc cost	Source node	Destination node	Arc cost
1	2	820	4	11	740
1	3	361	5	6	660
1	6	677	6	11	242
1	9	300	7	6	410
1	10	450	7	11	472
2	3	186	8	4	730
2	5	510	8	7	242
2	9	930	9	8	137
3	4	667	9	7	130
3	5	748	9	10	242
3	8	443	10	7	342
4	5	199	10	11	1310
4	6	340			

We apply the proposed neural network model in (8) to solve this program. Figure 2 displays the transient behavior of x(t) based on the neural network model (8) with random initial points and $\gamma = 4$.

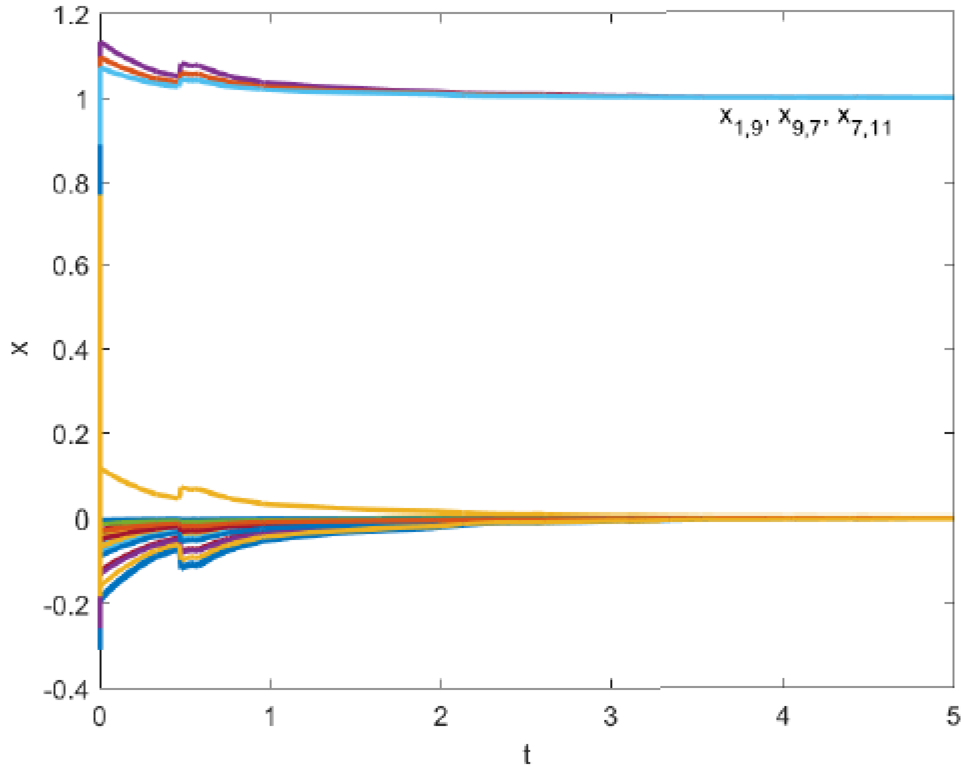

Figure. 2. Transient behaviors of the neural network (8) with random initial points and $\gamma = 4$.

6 CONCLUSION

In this paper, an efficient arti_cial neural network model for solving the non-linear programming problems with fuzzy parameters (FNLP) was proposed. We solved the proposed neural network model by an ODE. Here, we reformulated the KKT conditions of the FNLP into an efficient recurrent neural network model to solve the FNLP. Furthermore, in the neural network method, we have not this limitation that whether we choose the initial point from outside of the convergence region or not; of course, we obtain the unique solution of the problem. In fact, the neural network method does not depend on the initial point; this is because the model is globally convergent.

REFERENCES

[1] Ahuja, R. K., Magnanti, T. L., Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, New Jersey.

[2] Antonio, J. K., Huang, G. M., Tsai, W. K. (1992). A fast distributed shortest path algorithm for a class of hierar-chically clustered data networks, *IEEE Transactions on Computers* **41** (6), 710-724.

[3] Bazaraa, M. S, Shetty, C., Sherali, H. D. (1979). *Nonlinear Programming, Theory and Algorithms*, New York: John Wiley and Sons.

[4] 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.

[5] Effati, S., Mansoori, A., Eshaghnezhad, M. (2015). An efficient projection neural network for solving bilinear

programming problems, *Neurocomputing* **168**, 1188-1197.

[6] Eshaghnezhad, M., Effati, S., Mansoori, A. (2016). A Neurodynamic Model to Solve Nonlinear Pseudo-Monotone
Projection Equation and Its Applications, *IEEE Transactions on Cybernetics*, doi:10.1109/TCYB.2016.2611529.

[7] Xia, Y., Wang, J. (2015). A bi-projection neural network for solving constrained quadratic optimization problems,
IEEE Transactions on Neural Networks and Learning Systems **27** (2), 214-224.

[8] Miri, S. M., Effati, S. (2015). On Generalized Convexity of Nonlinear Complementarity Functions. Journal of Optimization Theory and Applications, 164 (2), 723-730.

[9] Kanzow, C. (1996). Nonlinear complementarity as unconstrained optimization. Journal of Optimization Theory and Applications, 88, 139-155.

[10] Pang, J. S. (1991). A B-differentiable equation based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems. Mathematical Programming, 51, 101-131.