Solving fuzzy shortest path problem with a new neural network model

Sohrab Effati [1], Morteza Pakdaman[2]
Department of mathematics, Tarbiat Moallem University of Sabzevar, Sabzevar, Iran

Abstract:
A new neural network model for solving fuzzy shortest path problem is proposed. We solved the neural network model with Euler method. This model can be applied for solving any fuzzy linear programming problem with fuzzy coefficients in objective function.

1. Model definition

A fuzzy shortest path problem can be formulated as follows:

$$\text{minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{m} \tilde{c}_{ij} x_{ij}$$

subject to

$$\sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} 1; & i=1 \\ 0; & i=2,3,\ldots,m-1 \\ -1; & i=m \end{cases}$$

$$x_{ij} \geq 0, \text{i, j} = 1,2,\ldots,\text{m}.$$

This problem is one of the important models in theory of networks. Many problems in applications such as transportation, communication and routing can be modeled as a shortest path problem (SPP). There are several traditional algorithms for solving this special linear programming problem such as simplex method (see [1]) or Dijkstra algorithm (See [2]). But in real world phenomena the arc lengths (c_{ij}'s) are not really known and so they aren't crisp. In this condition we can formulate a fuzzy shortest path problem (FSPP), where $\tilde{c}_{ij}$'s are (triangular or trapezoidal) fuzzy numbers . Yinzhen et al ([3]), solved FSPP with a neural network. Their method was based on penalization. Here we introduce a more simple fuzzy neural network as follows.

2. Solving Method

To solve FSPP, we introduce the following fuzzy optimization problem:

$$\min_{x} \quad \tilde{P}(x) = \tilde{c}x + \frac{k}{2}\|Ax-b\|_2^2 , \qquad x \geq 0. \tag{2}$$

[1] Effati@sttu.ac.ir
[2] Pakdaman.m@gmail.com

Where A is the matrix of technological coefficients for FSPP. (Note that here for simplicity and to save space, we skipped most fuzzy introductory requirements and just state some results) .

Theorem 2.1 (see [4]). Suppose $\tilde{f}$ is a fuzzy function with $\tilde{f}(x)[\alpha]=[\underline{f}(t,\alpha),\overline{f}(t,\alpha)],\ 0\le\alpha\le 1$ then if $\tilde{f}$ is differentiable (fuzzy differentiable), then $\underline{f}(t,\alpha),\overline{f}(t,\alpha)$ are differentiable functions and we have : $\tilde{f}'(x)[\alpha]=[\underline{f}'(t,\alpha),\overline{f}'(t,\alpha)],\ 0\le\alpha\le 1$.

Definition 2. 1 (see [4]). Let $\tilde{f}:\Omega\subseteq R^n\to E$ be a fuzzy mapping (E is the set of all fuzzy numbers), where Ω is an open subset of R^n. Let $(x_1,x_2,\ldots,x_n)\in\Omega$ and $\frac{\partial}{\partial x_i}, i=1,2,\ldots,n$ stands for the partial differentiation with respect to the i th variable x_i. Assume that for all $\alpha\in[0,1]$, $\underline{f}(t,\alpha),\overline{f}(t,\alpha)$ (the $\alpha-$cuts of f) have continuous partial derivatives. Define: $\frac{\tilde{\partial}\tilde{f}(x)}{\partial x_i}[\alpha]=[\frac{\partial\underline{f}(x,\alpha)}{\partial x_i},\frac{\partial\overline{f}(x,\alpha)}{\partial x_i}], i=1,2,\ldots,n;\alpha\in[0,1]$.
(3)

If for each *i=1,...,n*, (3), defines the $\alpha-$cuts of a fuzzy number, then we will say that $\tilde{f}$ is differentiable at x , and we write : $\tilde{\nabla}\tilde{f}(x)=(\frac{\tilde{\partial}\tilde{f}(x)}{\partial x_1},\frac{\tilde{\partial}\tilde{f}(x)}{\partial x_2},\ldots,\frac{\tilde{\partial}\tilde{f}(x)}{\partial x_n})$ .We call $\tilde{\nabla}\tilde{f}(x)$ the gradient of fuzzy function $\tilde{f}$ at x .

Theorem2. 2 (see [4]). Let $\tilde{f}$ be a differentiable fuzzy function at $x\in\Omega\subseteq R$ (Ω is an open set) if x is a point of local minimum then $\tilde{\nabla}\tilde{f}(x)=\tilde{0}$.

If we apply theorem 2.2 for problem (2), we should have: $\tilde{\nabla}\tilde{P}(x)=\tilde{0}$. Or we should have $\tilde{c}+k\sum_{i=1}^{m}a^{i^T}(a^i x-b_i)=\tilde{0}$. Now according this, we introduce the neural network model as:

$$\frac{\tilde{d}x}{dt}=-\tilde{c}-k\sum_{i=1}^{m}a^{i^T}(a^i x-b_i) \qquad (4)$$

We proved that this model is convergent to the optimal solution of FSPP. Note that here x is not fuzzy and $\frac{\tilde{d}}{dt}$ stands for fuzzy derivative of x . Now we can solve (4) with any numerical methods (for solving fuzzy differential equations). We solved this neural network model with Euler method.

3. Conclusions

In this extended abstract of our paper, we proposed a new neural network model for solving fuzzy shortest path problem. This method can be applied for solving any fuzzy linear programming problem with fuzzy coefficients in the objective function. Quickly convergence, simple computer programming and applicability for large networks are some advantages of this method.

References

[1]. M.S. Bazaraa, John J. Jarvis, H.D. Sherali, *Linear programming and network flows*, John Wiley and Sons, New York, (1992).

[2]. G.B Dantzig, M. N Thapa, *Linear programming*, Springer, New York, (1997).

[3]. Yinzhen Li, Mitsuo Gen and Kenichi Ida, *Solving fuzzy shortest path problems by neural network*, Computers ind. Engng, Vol. 31 (1996) 861-865.

[4]. M. Panigrahi, G. Panda, S. Nanda, *Convex fuzzy mapping with differentiability and its application in fuzzy optimization*, European Journal of Operational Research (Article in press).