# An expert system for network control problems and its applications in large scale network design under uncertainty

Mehdi Ghatee, Mehdi Hashemi
*Laboratory of Network & Optimization Research Center (NORC),*
*Department of Computer Science, Amirkabir University of Technology, Tehran, Iran.*
*Emails: {ghatee,hashemi}@aut.ac.ir, URLs: math-cs.aut.ac.ir/∼ghatee, www.aut.ac.ir/hashemi*

**Abstract**

This paper describes an expert system to find the control parameters in order to optimize the performance function of network when the network users are intelligent and we have granular information about network. Because of intelligent agents, a traffic assignment model is applied to predict the link flows. The uncertainty of granular information of network is also captured with fuzzy sets. We obtain equilibrium flows from a traffic assignment model with fuzzy costs. The performance function is concluded from a simulation scheme considering three criteria: network congestion, traveling time and social attachment. To optimize the performance function with respect to control variables, we use particle swarm optimization (PSO) approach. Through optimization process in the case of large scale networks, a lot of evaluation of the performance function is necessary which is computationally heavy. Thus a multilayer perceptron is used as a metamodel to predict the system behavior when the control parameters are varying. Both of the components of the proposed expert system, metamodel and the optimizer can be implemented in parallel manner, thus it is possible to find near optimal control parameters in large scale networks. Such method can be pursued to deduce the congestion through urban network when the links are controlled by cost instruments using RFID technology or camera with signal processing. The place of such algorithm in network design is also investigated.

**Keywords**: *Simulation-Optimization; Meta-heuristic Algorithms; Metamodel; Path Enumeration; Granular Information*

## 1 Introduction

In this paper we consider a network with intelligent agents. Such agents have a perception about optimality and they imply to use traffic assignment models to predict their behavior. These predictions are used to enhance the performance of network and to satisfy the users' demand. On the other hand traffic assignment models are dependent on control parameters. Ghatee and Hashemi[4] surveyed on some references on this topic up to 2005.
The most noticeable method in this area is simulation-optimization. This technique includes a metamodel and a metaheuristic algorithm. Metamodel is used to imitate the behavior of a real system or simulation results; while metaheuristic algorithm considers metamodel to find a reasonable near optimal solution without any guarantee about closeness to optimality. For example, Ceylan and Bell [2] implemented a genetic algorithm on the results of a simulation software. Metamodel was also be used in traffic control by Allsop and Charlesworth [1] in which a spline interpolation was proposed fitting the results of traffic assignment when the signal times are varying. Regression, time series, artificial neural network and RBF are the most famous techniques as metamodel. In this paper a neural network is utilized as metamodel. On the other hand, the uncertainty of real environment is a challengeable concept in this area. Stochastic

traffic assignment models [8] have been used for this aim in a wide variety of applications. Fuzzy extension of these models [6] can be used to contribute the stochastic ones in capturing the real environment, i.e., because of granular information of a real system, stochastic models are not sufficient for uncertainty handling and imprecision can be illustrated with fuzzy sets. To study the origin of fuzziness in traffic assignment models the interested reader may be referred to [5].
In what follows, we consider the problem of network control when the intelligent users reveal fuzziness in behavior. We first study a fuzzy traffic assignment model to predict the link flows. The performance function of network is defined with respect to equilibrium flows and control parameters. We use a multilayer perceptron neural network as metamodel to imitate the value of performance function when the control parameters are varying. On this metamodel, a particle swarm optimization (PSO) algorithm is used to find optimal control parameters for network links. We show how this expert system can be applied in urban networks when the networks are controlled by RFID technology or camera with signal processing. The parallel implementation of the proposed system provides opportunity to define real time control in urban network.

# 2 Some Preliminaries

In what follows, we briefly present some necessary concepts.

## 2.1 Granular Information

Granular information can be captured by fuzzy numbers efficiently. One of the simplest types of fuzzy sets is $LR$ fuzzy number in which a number around $a$ can be shown with $\tilde{a} = (a, a^L, a^R)$ in which $a^L$and $a^R$are the left and the right spreads, respectively. The membership function of this problem is as follows:

$$\mu_{\tilde{a}}(x) = \begin{cases} L(\frac{a-x}{a^L}), & x < a, \\ R(\frac{x-a}{a^R}), & x \geq a, \end{cases}$$

in which $L$ is a non-increasing left continuous function that $L(0) = 1$. The function $R$ is defined similarly.

## 2.2 Metamodel

When a functional form exists to imitate the behavior of system, it is reasonable to consider this functional form instead of original system in calculation processes. This functional form which can be presented as a polynomial, finite or infinite series, neural network etc, is called metamodel. Although metamodels are created on the base of simulation results, it is possible to find them considering real model. For a deep study we can offer [12]. Artificial neural networks have recently gained attention as fast and flexible devices to macroscopic modeling, simulation, and optimization. In this study, among a lot of schemes of artificial neural network as metamodel, multilayer perceptron is employed. It is trained with backpropagation, delta-bar-delta, extended delta-bar-delta, quick propagation, and Levenberg-Marquardt algorithms [3]. It has been proved that a multilayer perceptron, with enough number of neurons in hidden layer, can be used to predict each integerable function with respect to a given tolerance, see e.g., [3]. The results of the multilayer perceptron trained with the backpropagation algorithm is used in present paper which is in very good agreement with the results available in the experimental data. We follow an implementation of such neural network done with Martin and can be downloaded in [13].

## 2.3 PSO Metaheuristic

Metaheuristics are refereed to famous and wide-usage heuristic algorithms which cannot provide any guarantee about convergence, however they provide satisfying solutions usually near optimal. In this area, particle swarm optimization (PSO) has been followed by many researchers [9]. PSO algorithm was proposed by Kennedy and Eberhart [7]. The PSO is inspired by the behavior of bird flocking and fish schooling. A large number of birds/fishes flock synchronously, change direction suddenly, and scatter and regroup together. Each individual, called a particle, benefits from the experience of its own and that of the other members of the swarm during the search for food. The PSO models the social dynamics of flocks of birds and serves as an optimizer for nonlinear functions. The general principles for the PSO algorithm are stated as follows, see also [11].

Given an optimization function $f(P)$ where $P$ is a vector of $n$ real-valued random variables, the PSO initializes a swarm of particles, each of which is represented as $P_i = (p_{i,1}, p_{i,2}, ..., p_{i,n})$, $i = 1, 2, ..., K$, where $K$ is the swarm size. Thus, each particle is randomly positioned in the n-dimensional real number space and is a candidate solution to the optimization function. In PSO, particle $i$ remembers the best position it visited so far, referred to as $pbest_i$ and the best position by its neighbors. There are two versions for keeping the neighbors' best position, namely *lbest* and *gbest*. In the local version, each particle keeps track of the best position *lbest* attained by its local neighboring particles. For the global version, the best position *gbest* is determined by any particles in the entire swarm. Hence, the *gbest* model is a special case of the *lbest* model. The PSO is an evolutionary computation algorithm and in each generation, particle $i$ adjusts its velocity $v_{i,j}$ and position $p_{i,j}$ through each dimension $j$ by referring to, with random multipliers, the personal best position ($pbest_{i,j}$) and the swarm's best position ($gbest_j$, if the global version is adopted) using the following equations:

$$v_{i,j} = v_{i,j} + c_1 r_1 (pbest_{i,j} - p_{i,j}) + c_2 r_2 (gbest_j - p_{i,j}), \tag{1}$$

and

$$p_{i,j} = p_{i,j} + v_{i,j}, \tag{2}$$

where $c_1$ and $c_2$ are the acceleration constants and $r_1$ and $r_2$ are random real numbers drawn uniformly from $U(0, 1)$. Thus the particle flies through potential solutions toward $pbest_i$ and *gbest* in a navigated way while still exploring new areas by the stochastic mechanism to escape from local optima. The particle's velocity on each dimension is set restricted by a maximum velocity $v_{max}$, which controls the maximum travel distance at each iteration to avoid this particle flying past good solutions. The PSO algorithm is terminated with a maximal number of generations or the best particle position of the entire swarm cannot be improved further after a sufficiently large number of generations. In present paper we utilize an implementation which can be downloaded in [14].

## 2.4 RFID Technology for Urban Control

Since the traffic congestion is a severe problem in many modern cities around the world, the traffic managers dictate some limitation for vehicles whose owners want to enter in central business domain (CBD). For implementing such limitation, radio-frequency identification (RFID) technology may be used. RFID is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders. An RFID tag is incorporated into a car for the purpose of identification and tracking using radio waves [10]. In the applied problem of current paper, we assume RFID technology is sued for traffic control. However, this assumption does not limit the theoretic concept of the paper, e.g., when camera and signal processing is used to control CBD, analogous scheme can be pursued.

## 2.5 Contribution of Paper

In this paper we present an expert system for an urban network to find tax of each vehicle which should be paid when it enters urban links. We assume the network is controlled by RFID technology. The taxes have been determined online in order to optimize performance function of network regarding to congestion, traveling time and social attachment. The utility function, considering these objective functions, is imitated with a multilayer perceptron as metamodel. The optimal taxes are obtained by using PSO algorithm on metamodel. The proposed expert system can be used in congested urban network to control efficiently and dynamically. When due to implementation cost or public security, it is no possible to utilize RFID, control with camera and signal processing may be applied. Then the contribution of this paper can be followed for off-line control.

### 2.6 Adjunction of the travel cost and control cost

It is possible to join the multi-costs of a link into a single cost using weighting parameters. For example, the problem of this paper considers construction cost, traveling cost and control parameters. Assume the following adjunction cost for link $a$ :

$$\tilde{c}_a = w_{trav}(l_a, l_a^L, l_a^R) + w_{count} b_a + w_{const} f_a,$$

in which $(l_a, l_a^L, l_a^R)$, $b_a$ and $f_a$ are traveling cost, control parameter and construction cost, respectively. $w_{trav}$, $w_{count}$ and $w_{const}$ are corresponding importance weighting parameters.

# 3 Fuzzy traffic assignment

There are two important concepts related with understanding traffic in transport systems:

- The transport demand between places
- The transport supply between places establishing a set of paths between places that are generating and attracting movements

A fundamental concept is how traffic is distributed in a transportation network when we know its structure, capacity and the spatial demand. Such question may be answered by a traffic assignment model [8]. Because of granular information, the demand, capacity and cost cannot to be known precisely. For example, when image processing is used to find the capacity, the capacity can be captured with fuzzy sets more efficiently. The traveling time is also presented under link statues including light, normal, congested and accident. These linguistic concepts are exhibited with fuzzy numbers. The origin of fuzzy demand in traffic assignment is discussed in [5] while the fuzzy cost in the mentioned problem has been defended by Henn [6]. In the latter reference, possibility measure was used to provide a Logit model for traffic assignment which cannot be implemented in all of the cases. In what follows, we present another model which can be always utilized. Assume $\tilde{c}_a = (c_a^c, c_a^L, c_a^R)$ represents the traveling cost of link considering the granular information and simulation results. The following fuzzy traffic assignment model can be stated:

$$\min \tilde{Z}(f) = \sum_{r,s,p} \tilde{c}_p^{rs} f_p^{rs} + \theta \sum_{a\in A} \int_0^{x_a} I_a(w)dw \tag{3}$$

s.t.

$$\begin{aligned} &\textstyle\sum_{r,s,p} f_p^{rs} = d_{rs} \\ &\textstyle\sum_{r,s,p} f_p^{rs} \delta_a^{rs,p} = x_a \\ &f_p^{rs} \geq 0 \end{aligned}$$

The second term of the objective function (2) shows the link impedance, i.e., when the flow increases, the increasing function $I_a(w)$ increases. Since (3) is a minimization problem, in the optimal state, we face with minimal possible congestion. For example, a famous impedance function which presented by the US Bureau of Public Roads [15] is as bellow:

$$I_a(w) = t_a(1 + 2(\frac{x_a}{u_a})^3) \tag{4}$$

in which $u_a$ and $t_a$ are the capacity and the free travel time of link $a$,respectively. The objective function can be stated as bellow:

$$\tilde{Z}(f) = (\sum_{r,s,p} c_p^c f_p^{rs} + \theta \sum_{a\in A} \int_0^{x_a} I_a(w)dw, \tag{5}$$

$$\sum_{r,s,p} c_p^L f_p^{rs}, \sum_{r,s,p} c_p^R f_p^{rs})$$

By minimizing the most possible case, minimizing the pessimistic area and maximizing the most optimistic area, we have:

$$\min \sum_{r,s,p} c_p^c f_p^{rs} + \theta \sum_{a\in A} \int_0^{x_a} I_a(w)dw \tag{6}$$

$$\max \sum_{r,s,p} c_p^L f_p^{rs}$$

$$\min \sum_{r,s,p} c_p^R f_p^{rs}$$

s.t.

$$\sum_{r,s,p} f_p^{rs} = d_{rs}$$

$$\sum_{r,s,p} f_p^{rs} \delta_a^{rs,p} = x_a$$

$$f_p^{rs} \geq 0$$

This multi-objective problem is transformed as bellow when we employ the weighting parameters $\lambda_1$,$\lambda_2$ and $\lambda_3$:

$$\min Z(f) = \sum_{r,s,p} (\lambda_1 c_p^c - \lambda_2 c_p^L + \lambda_3 c_p^R) f_p^{rs} \tag{7}$$

$$+ \theta \sum_{a\in A} \int_0^{x_a} I_a(w)dw$$

s.t.

$$\sum_{r,s,p} f_p^{rs} = d_{rs}$$

$$\sum_{r,s,p} f_p^{rs} \delta_a^{rs,p} = x_a$$

$$f_p^{rs} \geq 0$$

**Theorem 3.1** *Assume*$\lambda_1 c_p^c - \lambda_2 c_p^L + \lambda_3 c_p^R \geq 0$ *for each path p. Let all of the paths convey flow in optimal assignment. The exact solution of model (7) is as follows:*

$$f_p^{rs} = \frac{e^{-\theta(\lambda_1 c_p^c - \lambda_2 c_p^L + \lambda_3 c_p^R)}}{\sum_{q\in P^{rs}} e^{-\theta(\lambda_1 c_q^c - \lambda_2 c_q^L + \lambda_3 c_q^R)}} d_{rs} \tag{8}$$

**Proof.** By the assumptions, it is easy to prove that the objective function of model (7) is convex and the constraints are also convex. Thus this model is convex programming problem. Thus the solution of Lagrangian problem is optimal. By some computation, one can show that (8) is the unique solution of Lagrangian problem of model (7). This completes the proof.
The possibility of using path $p$ is given with

$$Pos_p^{rs} = \frac{e^{-\theta(\lambda_1 c_p^c - \lambda_2 c_p^L + \lambda_3 c_p^R)}}{\sum_{q\in P^{rs}} e^{-\theta(\lambda_1 c_q^c - \lambda_2 c_q^L + \lambda_3 c_q^R)}} \tag{9}$$

and the equilibrium possible flow through path $p$ is stated

$$f_p^{rs} = Pos_p^{rs} d_{rs}. \tag{10}$$

Such flows may be referred as Logit flow in comparison with stochastic variants. We can now present the following algorithm:

**Algorithm 3.1** ***Fuzzy Traffic Assignment***
*Input: Incident matrix A, Demand matrix* $D = (d_{rs})$, *Fuzzy costs* $(\tilde{c}_a) = ((c_a^c, c_a^L, c_a^R))$, *threshold* $\varepsilon > 0$.
*Output: Logit flows.*

1. *For each link a set* $h_a^{old} = \infty$ *and* $h_a^{new} = 1$
2. *While* $max_a \mid h_a^{new} - h_a^{old} \mid < \varepsilon$ *repeat:*
3. *Update the fuzzy costs considering flow links* $h_a^{new}$.
4. *Find non-dominated paths between each couple of origin-destination with respect to fuzzy costs.*
5. *Find the path flows applying*
$$f_p^{rs} = \frac{e^{-\theta(\lambda_1 c_p^c - \lambda_2 c_p^L + \lambda_3 c_p^R)}}{\sum_{q \in P^{rs}} e^{-\theta(\lambda_1 c_q^c - \lambda_2 c_q^L + \lambda_3 c_q^R)}} d_{rs}$$
6. *Set* $h_a^{old} = h_a^{new}$ *and*
$$h_a^{new} = \sum_{r,s,p} f_p^{rs} \delta_a^p$$
7. *Go to step 2.*
8. *Return* $h_a^{new}$ *as equilibrium possible flows.*

# 4 Performance index of urban network

To improve the quality of service in urban network, the following three objective functions may be taken into account:

- Maximizing the usage of all links to decrease network congestion
- Minimizing the taxes in order to decrease social attachments
- Minimizing the traveling costs

The first and the second ones can be given with:

$$\min O_1 = \sum_{(i,j)} \frac{x_{i,j}}{u_{i,j} - x_{i,j}} \tag{11}$$

$$\min O_2 = \sum_{(i,j)} b_{i,j} x_{i,j}$$

To capture the third goal, we use the following average amount:

$$\min O_3 = \frac{1}{N} \sum_{i=1,\ldots,N} \sum_{l_{i,j} \in \tilde{l}_{i,j}} c_{i,j} x_{i,j} \tag{12}$$

where $N$ is the number of simulation tests and $c_{i,j} \in \tilde{c}_{i,j}$ is a real scenario corresponding to fuzzy link cost. It is important to implement computerized software to produce such scenarios with respect to degrees of possibility and the shape of the right and the left functions. When the importance weights of the presented objective functions are $k_1$, $k_2$ and $k_3$, one can optimize the following goal to improve the performance index of urban network:

$$O = k_1 O_1 + k_2 O_2 + k_3 O_3 \tag{13}$$

To predict the above objective as a function with respect to control parameters, we use a multilayer perceptron whose inputs are control parameters and whose output is the value of objective function (13) with respect to given control parameter. After training process, the metamodel should be validated considering some new data.

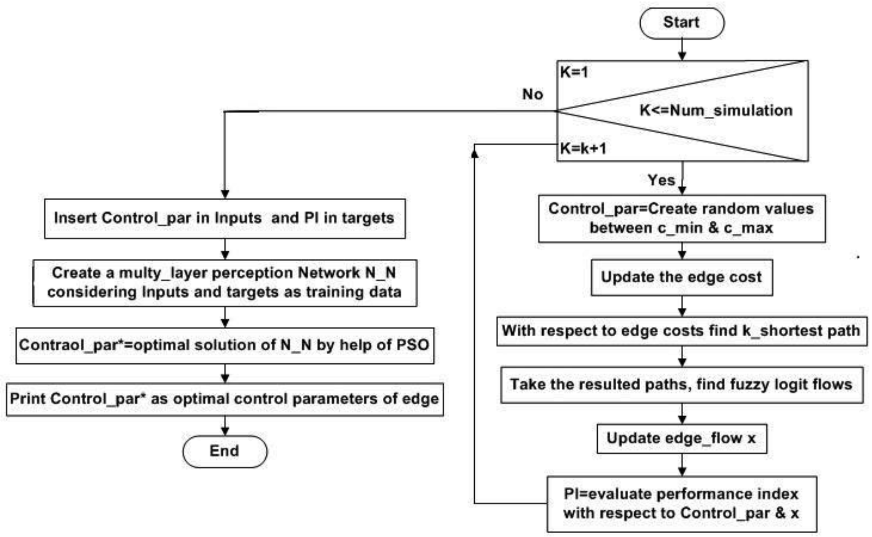


Figure 1: The flowchart of an expert system which can find the optimal control parameter.

# 5 An expert system

In this section an expert system is proposed to introduce the optimal control parameters when a PSO algorithm is done on a neural network which illustrates the behavior of performance index of an urban network. The flowchart of this intelligent scheme is depicted in Figure 1.
This system iterates $K\_simulation$. In each simulation, a random setting of control parameters between the meaningful bounds are constructed. For this setting, the equilibrium possible flow through different paths and then the performance index of network will be obtained. Such results are used to train the multilayer perceptron. Then a PSO algorithm is used on metamodel to obtain optimal control parameters. In the next section, it is reveal that such scheme can be shortened the computation time.

# 6 Numerical experiments

In this section we examine the performance of our expert system to find optimal taxes. We implement our software with Matlab code, version 7 on a computer with following properties: $Intel® \; Core^{TM} 2 \; Duo \; CPU \; T5780$ @ 2.00 $GHz$, and 0.99 $GHz \; of \; RAM$. Consider the network which was previously used in [5] and depicted in Figure 2 and corresponding data in Table 1. In addition to fuzzy costs for this network assume construction costs of links are assumed between 1000 and 9000 units. Also control parameters are limited between 10 and 100 units. The important weights of social attachments, congestion and traveling times are stated as 3, 6 and 7 respectively. To provide absorbing paths, we also use K-shortest path algorithm analogues to [5].
To evaluate the performance index, for each fuzzy link cost, 100 scenarios have been simulated. The importance weights of center, left and right spreads are considered as 0.3. The left and right shape functions are also $L(x) = 1- \mid x \mid$ and $R(x) = 1 - x^3$.
To create a metamodel to imitate the behavior of performance index with respect to the different control parameters, we use a 3-layer perceptron with 169, 10 and 1 neurons in the input, hidden and output layers, respectively. In order to provide training data, 400 simulation experiments with different control parameters are constructed. Then, control parameters are uniformly constructed between the lower and upper bounds of control parameters. The performance indices considering such control parameters are

determined. These couples of data are used to train the 3-layers perceptron network. After, training process some new data are used to check the robustness of neural network in order to predict the network responses. When the responses do not agreed with reality, it is necessary to construct more simulation experiments. It is important to note that other distribution functions with given average and standard deviation can be used to construct simulation experiments.
The time of producing such training data is 187.109 seconds, i.e., each evaluation of performance index consumes 0.468 second. The neural network also consumes 8.407 seconds to fit with respect to these data. Therefore, the sum of construction time of this metamodel is 195.516. The trend of descending the error of neural network in predicting the system responses, is depicted in Figure 3. This figure reveals that between $2000^{th}$ and $2500^{th}$ iterations, the convergence is happened. On the provided metamodel, we use PSO algorithm to find optimal control parameters. After 18 iterations and by consuming 0.907 second the reasonable control parameters have been found. Through PSO algorithm the metamodel has been loaded 4318 times. This means without using metamodel, according to 0.468 second per each evaluation, we need 2020.824 second in order to find same result. Thus, by using metamodel, we save 1825 second in computation time. The trend of descending the utility function of PSO through PSO algorithm is presented in Figure 4. The optimal control parameters are also given in Table 1.

**Table (1): Data of network depicted in Figure (2).**

| Tail node | Head node | Link Flow | Link Cost |
|---|---|---|---|
| 1 | 3 | 159 | 22 |
| 2 | 3 | 109 | 62 |
| 2 | 8 | 59 | 82 |
| 3 | 1 | 54 | 44 |
| 3 | 2 | 59 | 64 |
| 3 | 4 | 130 | 10 |
| 3 | 7 | 141 | 52 |
| 4 | 3 | 58 | 48 |
| 4 | 6 | 130 | 51 |
| 5 | 6 | 68 | 45 |
| 6 | 4 | 58 | 53 |
| 6 | 5 | 107 | 80 |
| 6 | 7 | 97 | 83 |
| 6 | 12 | 102 | 32 |
| 7 | 3 | 57 | 61 |
| 7 | 6 | 95 | 44 |
| 7 | 8 | 39 | 100 |
| 7 | 11 | 82 | 100 |
| 8 | 2 | 109 | 25 |
| 8 | 7 | 95 | 95 |
| 8 | 9 | 58 | 50 |
| 8 | 10 | 0 | 77 |
| 9 | 8 | 164 | 69 |
| 10 | 8 | 0 | 25 |
| 11 | 7 | 82 | 86 |
| 11 | 12 | 71 | 43 |
| 11 | 13 | 138 | 46 |
| 12 | 6 | 71 | 62 |
| 12 | 11 | 102 | 59 |
| 13 | 11 | 107 | 84 |

This experiment concludes that:

1. By increasing the number of nodes of hidden network, we could not find any improvement.

2. By increasing the number of training data, the metamodel could not be improved and in some cases the results become unstable.

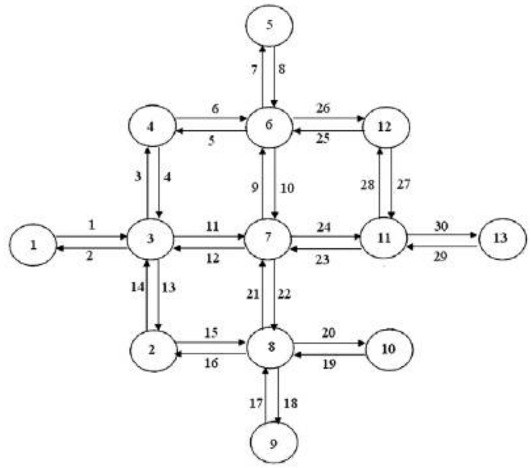


Figure 2: A semi-urban network with 13 nodes and 30 links.

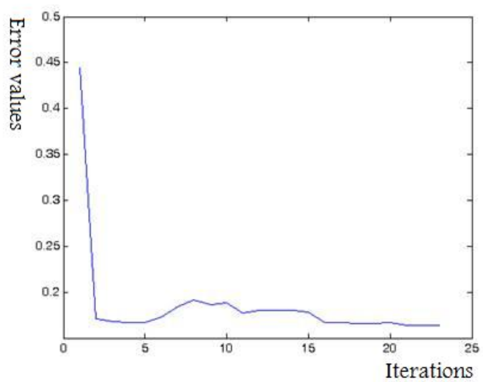


Figure 3: The error through training process of neural network with 400 experiment data.

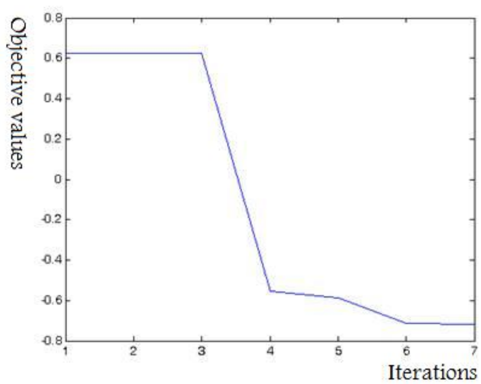


Figure 4: The improvement of utility function of PSO algorithm.

3. The trend of errors through training iterations of neural network was not always decreasing. Thus, it is necessary to check the trend of error and if it is not acceptable, another loading is necessary.

4. Because the neural network can be saved, it is no necessary to create metamodel every time and so this expert system can be applied for online control in urban networks.

5. The PSO algorithm always converges.

# 7 Conclusion

This paper provides an expert system in order to find the control parameters in a given network. This system can capture real and fuzzy quantities simultaneously. Since this scheme uses a neural network as metamodel, the evaluation of performance index is essentially easy and savable. PSO algorithm is also used to find satisfying control parameters. This scheme can be pursued to present optimal design of network when control process is noticed by decision maker in designing phase. RFID technology as a feasible instrument for implementing this expert system is widely used in large cities in future and this means such expert system has an important role in urban control. By the way, such expert system can be done when camera and signal processing is used for network control.

# References


[1] R.E. Allsop, J.A. Charlesworth, *Traffic in a signal controlled road network: An example of different signal timings including different routing*, Traffic Engineering and Control, 1977, pp. 262–264.

[2] H. Ceylan, M.G.H. Bell, *Traffic signal optimization based on genetic algorithm approach, including drivers routing*, Transportation research part B **38** (2004) pp. 329–342.

[3] A. Chichocki, R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, John Wiley & Sons, Chichester, 1993.

[4] M. Ghatee, S.M. Hashemi, *Descent direction algorithm with multicommodity flow problem for signal optimization and traffic assignment jointly*, Applied Mathematics and Computation **188** (2007) pp. 555–566.

[5] M. Ghatee, S.M. Hashemi, *Traffic assignment model with fuzzy level of travel demand: An efficient algorithm based on quasi-Logit formulas*, European Journal of Operational Research **194** (2009) pp. 432–451.

[6] V. Henn, *Fuzzy route choices model for traffic assignment*, Fuzzy Sets and Systems **116** (2000) pp. 77–101.

[7] J. Kennedy, R.C. Eberhart, *Particle swarm optimization*, Proceedings of the IEEE International Conference on Neural Networks, **IV** (1995) pp. 1942–1948.

[8] Y. Sheffi, *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice-Hall, USA, 1985.

[9] A. I. F. Vaz, L. N. Vicente, *Fuzzy route choices model for traffic assignment*, Journal of Global Optimization **39** (2007) pp. 197–219.

[10] W. Wen, *A dynamic and automatic traffic light control expert system for solving the road congestion problem*, Expert Systems with Applications **34** (2008) pp. 2370–2381.

[11] P.Y. Yin, *A discrete particle swarm algorithm for optimal polygonal approximation of digital curves*, Journal of Visual Communication & Image Representation **15** (2004) pp. 241–260.

[12] C.W. Zobel, K.B. Keeling, *Neural network-based simulation metamodels for predicting probability distributions*, Computers and Industrial Engineering **54** (2008) pp. 879–888.

[13] A site for downloading a neural network code: *www.dia.fi.upm.es/∼jamartin/download.htm.*

[14] A site for downloading a particle swarm optimization code: *www.norg.uminho.pt/aivaz/pswarm.*

[15] US Bureau of Public Roads, 1964. *Traffic Assignment Manual,* US Department of Commerce, Washington, DC.