Real Time Traffic Control: A Soft Computing Approach

D. TEODOROVIĆ [a] J. POPOVIĆ [b] P. LUČIĆ [c]
[a]Department of Civil and Environmental Engineering
Virginia Polytechnic Institute and State University, Northern Virginia Center
7054 Haycock Road, Falls Church, VA 22043-2311
USA
http://filebox.vt.edu/users/duteodor/
[b]Faculty of Transport and Traffic Engineering
University of Belgrade
Vojvode Stepe 305, 11000 Belgrade
Serbia and Montenegro

[c]CSSI, Inc.
400 Virginia Avenue SW, Suite 710, Washington DC,
USA

Abstract: - This paper describes Soft Computing approach to modeling time-dependent (dynamic, real time) transportation phenomenon characterized by uncertainty. The proposed "intelligent" control systems that are based on a combination of fuzzy logic (or neural networks) and mathematical programming (or heuristic) techniques make "on line" control decisions of the highest quality. In the first step of the proposed model, the best control strategies are developed off line for many different traffic patterns. These strategies are developed using mathematical programming or heuristic approach. In the second step, learning from the best strategies, fuzzy rule base is created from numerical data (or neural network is trained). Applications of the systems are considered for the stochastic vehicle routing, and real-time traffic control at the isolated intersection.

Key-Words: - Uncertainty Modeling, Fuzzy Sets, Neural Networks, Transportation, Traffic

1 Introduction

Most complex traffic and transportation engineering problems are characterized by *uncertain* transportation supply, demand, and/or cost patterns [1]. This means that there are limitless number of potential situations in traffic and transportation that request adequate control and action. Every potential *situation* that can happen in the transportation system requests adequate *decision* and *action*. Most of these decisions must be made in *real time*. Some of these decisions involve human decision-makers whereas others involve automatic control mechanisms triggered by computer hardware. The initial assumption in this paper is that it is possible to develop a new type of control system that makes on-line decisions of a high quality. In other words, this paper assumes that it is possible to develop the control systems that will recognize different situations and make the appropriate real- time decision without knowing the functional relationships between individual variables. Intelligent traffic and transportation control systems should be able to generalize, adapt, and learn based on new knowledge and new information. The concept proposed in this paper is general and it can be applied to a broad class of real-time engineering control problems that are characterized by uncertainty. The paper is organized as follows: the proposed control system is presented in section 2. Successful examples of the proposed control concept in solving complex traffic and transportation problems are shown in section 3. Section 4 presents the concluding remarks and further research orientations.

2 Soft Computing Real-Time Control

Various traffic patterns constantly occur in transportation systems. Traffic control mechanisms constantly respond to new traffic patterns through different control decisions and actions. Specific traffic conditions must be encountered and recognized by traffic control mechanisms. This means that the good traffic control mechanisms should have the *ability to distinguish* one traffic scenario from another. The number of different traffic scenarios is practically limitless. Traffic control mechanism should be capable to "fight" with the traffic conditions it has encountered in the past, as well as with the unknown traffic conditions that appear in the transportation system for the first time. Consequently, triggering the specific action in real time (green light extension at the intersection, flight cancellation in the case of serious airline schedule disturbance, assignment of particular cab to specific request, etc) depends on the recognized "traffic scenario".

2.1 Creating The Traffic Scenarios Database

We can be familiar with some traffic situations that have encountered in the past and be prepared for them with adequate response. On the other hand, we must be able to find adequate response for the traffic situations that we are facing for the first time. We can create the traffic scenarios database of the considered transportation system by collecting data and/or by simulation. Depending on the context of the problem, this means that we are able to "predict" moments of time in which different events will happen. For example, in context of isolated intersection on-line control problem, this means that we are able to predict exact time of the arrival of each vehicle on each approach during certain time period. By simulation we can create different traffic patterns during considered time period. The greater the database, the better the expected performances of future traffic control system.

2.2. Creating The Best Control Actions for Known Traffic Scenarios

Let us first try to develop the best control actions for known traffic scenarios. Known traffic scenario (generated by simulation) give us full information about future events. In the case of perfect prediction we must be able to make optimal decisions. For known traffic scenarios, we should have adequate control actions. Let us denote by (P_1) **t**he problem of discovering action_1 for given scenario_1. For known scenario_1, depending on the studied transportation phenomenon, the problem (P_1) could be solved (off-line) using linear programming, nonlinear programming, dynamic programming, multi-objective programming, or by using some metaheuristic algorithms (genetic algorithms, simulated annealing technique, taboo search). The problem of discovering action_2 for given scenario_2 is denoted by (P_2). In this way, for given set of m scenarios, *Traffic Scenarios Set* $=\{scenario_1, scenario_2, \ldots, scenario_n\}$, the set of actions, *Actions Set* $= \{action_1, action_2, \ldots, action_n\}$ is produced after solving the corresponding problems $P_1, P_2, \ldots P_n$. We can get the optimal solution (or "good" solution) for every considered traffic scenario.

2.3. Creating The Intelligent Transportation Control Systems

When creating the intelligent transportation control systems we use artificial neural networks, or fuzzy logic techniques. We create fuzzy rule base from numerical examples ("Traffic Scenario- Best Control Action" Database). There are few different methods for generating fuzzy rule base from numerical data [2], [3]. Theoretical results reached during the past several years have indicated that fuzzy logic systems are universal approximators and this explains why fuzzy logic systems are so successful in engineering applications [3]. Feedforward neural networks also approximate unknown functions, that is, they can be considered as universal approximators. The theorem proved by Hornik et al. and Cybenko states that a multilayered feedforward neural network with one hidden layer can approximate any continuous function up to a desired degree of accuracy provided it contains a sufficient number of nodes in the hidden layer [4],[5]. The proposed system that makes on-line decisions of a high quality is capable to recognize different situations, to generalize, to adapt, to learn and to make the appropriate decision without knowing the functional relationships between individual variables. The proposed approach for creating the Intelligent Transportation Control Systems could be formulated through the following steps:

Step 1: Using simulation, generate many different traffic scenarios.

Step 2: Formulate considered problem and find the optimal solution or sub-optimal solution for each generated traffic scenario using optimization techniques or heuristic algorithms. Create the "Traffic Scenario-Best Control Action" Database.

Step 3: Based on the "Traffic Scenario-Best Control Action" Database resulted from Steps 1 and 2, create the Intelligent Transportation Control System.

The following question is very important: Is the proposed system capable to find "good" solution for the unknown traffic "scenario? To properly answer this question we must ask ourselves the following: What is the *ideal control strategy* for *unknown traffic scenario*? How can we create the ideal control strategy for unknown traffic scenario? The answer is very simple. We use the same techniques (optimization techniques, and/or heuristic algorithms) that we used to create "Traffic Scenario-Best Control Action" Database. We consider our system as "good enough" if it is capable to produce control strategies "similar" to ideal control strategies in the case of unknown traffic scenarios. (In some cases it can happen that our system produces the ideal control strategy).

3. Successful Examples Of The Proposed Approach In Solving Complex Traffic Problems

3.1 Intelligent isolated intersection

Consider an isolated "T" intersection consisting of two one-way streets as shown in Fig. 1. In this paper we will not take into consideration the whole set of engineering details like detector placement, calculation of the minimum and maximum green times, yellow and all-red times, and pedestrian requirements. The detectors provide real time information on the numbers of incoming vehicles, stopped vehicles, and the total vehicle waiting time (delay) on each approach.

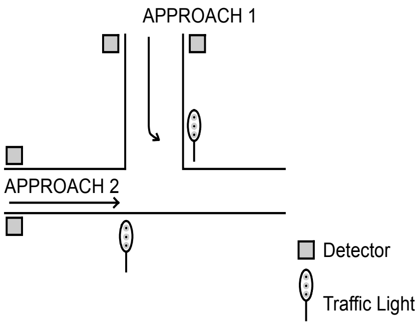

Fig.1 - "T" intersection consisting of two one-way streets

This information is updated in short time intervals. Based on this information, a set of rules is applied to control the signal phase for the next time interval. The decision is either to continue or to terminate the current signal phase [6],[7]. The question is how to build the rules so that they satisfy the following objectives of signal control: (1) to minimize the total number of stopped vehicles S, and (2) to minimize the total delay D over a given time period (0, T). In other words, our performance index ("cost" or "penalty function") should represent some weighted combination of stops and delays. For example, the performance function could read as follows:

$$F = w_1 S + w_2 D \qquad (1)$$

where:

w_1 - the weight (the importance) given to the total number of stopped vehicles; w_2 - the weight (the importance) given to the total delay; $w_1 + w_2 = 1$.

The terms S and D are added with weights of w_1 and w_2. This enables multi-criteria sensitivity analysis and generation of a great number of different control strategies depending on chosen criteria weights (importance). Consider just one of the approaches of Fig. 1. Let 1 denote the situation when the signal phase on the approach in question is green, and, 0 the situation when the approach in question is red. Then over the period (T), each small time interval may be designated either 0 or 1, and the chain of the numbers such as the following indicates the pattern of signal phase change over T:

010111100001111………11000011110000011l.

This sequence represents how the signal phase changed during time T. We use genetic algorithms to develop the optimum sequence of signal phases assuming that the future traffic conditions at the intersection are known [8]. Many different hypothetical traffic scenarios are generated, and for each scenario, the corresponding best solution consisting of a string of 1's and 0's is developed using Genetic Algorithm. This set of solutions constitutes the "Traffic Scenario – Best Control Strategy" Database for the intersection. This database is the starting point for creating the intelligent control system. We generated fuzzy rule base using Wang-Mendel's method [2]. Typical fuzzy rule in the fuzzy rule base is, for the example, the following one:

If *the total number of approaching vehicles is SMALL, and if the total number of vehicles waiting in the other approach is LARGE, and if time elapsed since the last phase change is VERY LONG*

Then *the time length until the next phase change is VERY SHORT*

Because the Genetic Algorithm result was the retrospectively derived best solution for a given traffic pattern, the performance associated with it is considered as the target or reference for evaluation. The criterion used to compare the two cases ("intelligent" transportation control result vs. ideal control result (Genetic Algorithm result) is the performance index defined in relation (1). The vehicle arrivals are assumed to follow the Poisson process. Thirty-two patterns are generated with each pattern lasting for 10 minutes (600 seconds). The headway between two successive vehicles is not less than 1.5 seconds. The size of the small time interval at which control decisions are made is 6 seconds. The best decision at each small time interval is developed using the genetic algorithm. The specific values of weights between the minimum total delay (w_1) and minimum total number of stopped vehicles (w_2) are as follows: $w_1 = 0, 0.2,...,1$; $w_2 = 1, 0.8,...,0$. Thus, for a given traffic pattern, six best solutions, corresponding to each weight combination, are developed. Fig. 2 shows the number of stopped vehicles for the traffic arrival patterns that *are not previously used*. The comparison was made between the results obtained using the "intelligent" system and those obtained using genetic algorithm (The Genetic Algorithm results represent number of stopped vehicles values attainable when the *future is known (ideally predicted*). Bearing this fact in mind, as well as the fact that the "intelligent" system operates in an on-line regime in conditions of *uncertainty*, it can be concluded that good results would be achieved using the intelligent system. In this figure (as in the case of the total delay), most points line up along the 45-degree line.

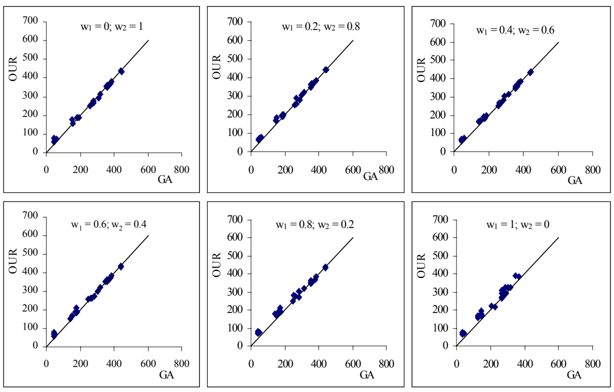

Fig. 2 – Total number of stopped vehicles: Comparison between the ideal control strategies and the control strategies produced by the developed intelligent system

This indicates that intelligent transportation control results and Genetic Algorithm results (ideal control decisions) are very similar and that the rules from the proposed method can yield solutions close to the best solution.

3.2 Intelligent Vehicle Routing System

Let us assume that there are n nodes in the network to be served (Fig. 3). We also assume that vehicles of the same size provide service. We denote vehicle capacity by C. Vehicles set out from depot, serve a number of nodes, and on completion of their service, return to the depot. The classical vehicle routing problem consists of finding the set of routes that minimizes transport costs. We assume that the demand at each node is only approximately known. Such demand can be represented by a probability density function or in the case of subjective estimate by the appropriate fuzzy number. Without loss of generality, in this paper we assume that demand D_i at any node i ($i = 1,2,\ldots,n$) is represented by the Normal distribution with mean μ_i and standard deviation σ_i. The problem of routing vehicles in the case of stochastic demand at nodes is known as the *stochastic vehicle routing problem* [9]. The basic characteristic of the stochastic vehicle routing problem is that the real value of demand at a node is only known when the vehicle reaches the node. Due to the uncertainty of demand at the nodes, a vehicle might not be able to service a node once it arrives there due to an insufficient capacity. Such situation is known as a "*route failure*". In the case of "route failure" different actions need to be applied. We assume that in such situations the vehicle returns to the depot, empties what it has picked up thus far, returns to the node where it had a "failure," and continues service along the rest of the planned route (Fig. 3). Obviously, when evaluating the planned route, the additional distance that the vehicle makes due to "failure" arising in some nodes along the route must be taken into consideration. The problem (P) logically arises of designing such a set of routes, which will result in the least total sum of planned route lengths and additional distance covered by vehicles due to failure. The problem (P) is solved in this paper many times for different scenarios (known the random demand values at all nodes). In order to solve problem (P), we first solved corresponding Traveling Salesman Problem using various heuristic algorithms. In this way, we created "Giant vehicle route". In the next step, we "walked" along the created giant route and we have decided when to finish with one vehicle route and when to start with the next vehicle route. These decisions were easily made since we knew demand at every node and vehicle capacity.
After serving the first k nodes, the available capacity of vehicle B_k will equal:

$$B_k = C - \sum_{i=1}^{k} D_i \qquad (2)$$

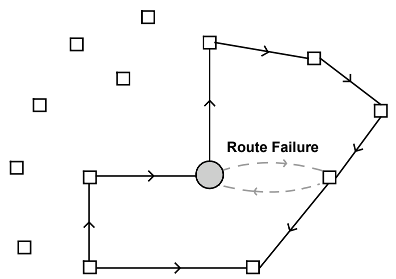

Fig. 3 - "Failure" at a node of the planned route

In the case of stochastic vehicle routing problem, the "strength" of our preference for the vehicle to serve the next node after it has served k nodes depends on the available capacity B_k, as well as on expected demand in the next node. We can expect that at a certain time point the route will have "small," "medium," or "big" number of nodes. We will denote by n_k the expected number of new nodes in the route after vehicle already has served k nodes. The linguistic expressions "small number of new nodes," "medium number of new nodes," and "big number of new nodes" can be represented by corresponding fuzzy sets. Available capacity can also be subjectively estimated, for example, as "small," "medium, and "large." Let us denote respectively by X_1, X_2 and X_3 the following variables:

$$X_{1k} = \frac{B_k}{C}; \quad X_{2k} = \frac{\mu_{k+1}}{C}; \quad X_{3k} = \frac{o_{k+1}}{C} \qquad (3)$$

The first variable represents relative available capacity after serving the first k nodes. The second variable represents relative expected demand in the next node, while the third one describes relative variability of the demand in the next node. The typical rule in the approximate reasoning algorithm to determine the expected number of the new nodes in the route can be the following one:

If *Relative available capacity is LARGE and Relative expected demand in the next node is SMALL and Relative variability of the demand in the next node is SMALL*

Then *Expected number of the new nodes in the route is BIG*

We can see that the antecedent of the rules contains remaining vehicle capacity and the expected demand in the next node. The consequence contains the expected number of new nodes in the route. In this paper the fuzzy rule base is generated from numerical examples using the procedure proposed by Wang and Mendel [2]. For known available capacity B_k that remains after serving k nodes, and for known characteristics of the demand in the next node it is possible to use the approximate reasoning rules to determine the expected number of new nodes in the route. We are now able to answer the following question: should we send the vehicle to the next node or return it to the depot after completing service to k nodes? Let the expected number of the new nodes in the route equal n_k^*. Based on this value, a decision must be made whether to send the vehicle to the next node or return it to the depot. The vehicle should be sent to the next node if the case when $n_k^* \geq 1$. In the opposite case, when $n_k^* < 1$, the vehicle should be returned to the depot. The generated fuzzy rule base enables "on line" developing of the vehicle routes. The vehicle routes are created in the following way:

Step 1: Using some heuristic algorithm solve the Traveling Salesman Problem.

Step 2: First include the depot in a route. Then include nodes in the route in the same order as they appear in the Traveling Salesman Route. Before deciding to include a node into the route, first use generated fuzzy rule base to calculate the expected number of new nodes in the route. If the calculated expected number of the new nodes is greater than or equal to one, include the node in the route. Otherwise, this node becomes a first point of the new vehicle route. Finish with the algorithm when all nodes are included in the routes.

The developed model was tested on well-known TSP benchmark problems [10]. We have compared the results obtained by the proposed process above with the a priori known solution. Maximum and average relative error values are given in Table1.

Table 1- Maximum and average relative error values

Problem	Maximum relative error [%]	Average relative error [%]
Eil51	5.83	0.607
Berlin52	5.98	1.44
St70	4.05	1.63
Pr76	5.83	1.75
Kroa100	5.75	2.42
Eil101	5.44	2.18
Tsp225	4.70	1.49
A280	3.48	1.26
Pcb442	4.32	1.15
Pr1002	2.38	0.93

Each performed numerical experiment is represented by the following two solutions: (a) the solution obtained when future demand is known in advance; (b) the solution obtained by the proposed "Intelligent" system that makes real-time decisions. The obtained solution pair is shown in Fig. 4 (Example: Eil101).

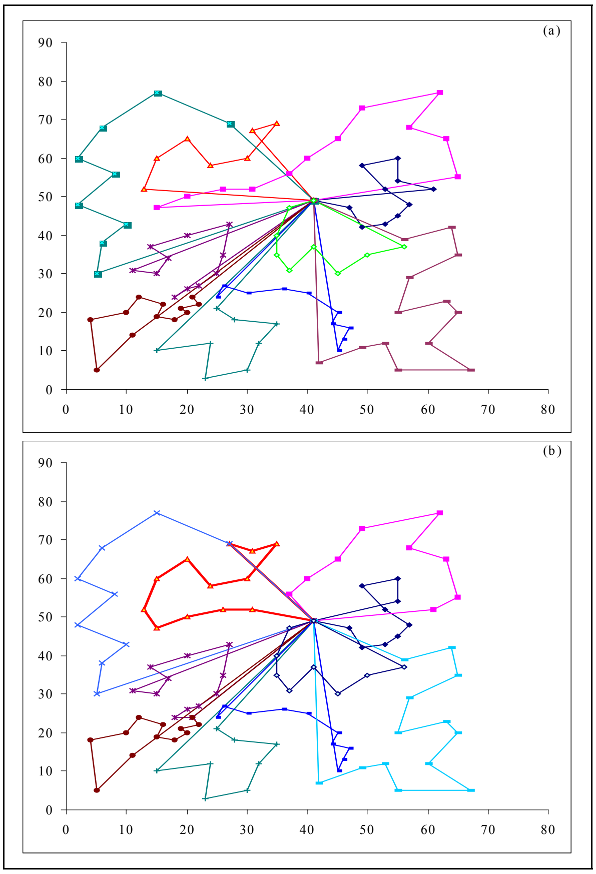

Fig. 4 - Example Eil101; (a) the solution obtained when future demand is known in advance; (b) the solution obtained by the proposed "Intelligent" system

4. CONCLUSIONS

In this paper, an "intelligent" traffic and transportation control systems are proposed. The proposed process learns from the best solutions obtained assuming that the future situations are known. Combining many solutions, a set of rules is developed. All pairs ("traffic scenario-appropriate set of decisions") were used to produce a fuzzy rule base. Evaluating the performance of the fuzzy rules developed by this process is also noble. Because the best solution is known for a particular pattern, the performance of the proposed rules can easily be checked against the result of the best solution. Many tests show that the outcome of the proposed rules is nearly equal to the best solution. The proposed system has the possibility to learn from examples, which means that it is adaptable. There are numerous transportation and logistic problems where this research could apply. The proposed concept is especially important for research activities whose unified themes are uncertainty (randomness, stochasticity, fuziness, ...) and time-dependence (dynamic, real-time).

References:

[1] D. Teodorović, K. Vukadinović, *Traffic Control and Transport Planning: A Fuzzy Sets and Neural Networks Approach* Boston/Dordrecht/London, Kluwer Academic Publishers, 1998.

[2] L-X.Wang, & J. Mendel, Generating Fuzzy Rules by Learning from Examples, *IEEE Transactions on Systems, Man and Cybernetics*, *22* (6), 1992, 1414-1427.

[3] J. Mendel, Fuzzy Logic Systems for Engineering: A Tutorial, *Proceedings of the IEEE, 83*, 1995, 345-377.

[4] K. Hornik, M. Stinchcombe, & H. White, Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, *2*(5), 1989, 359-366.

[5] G. Cybenko, Approximation by Superpositions of a Sigmoidal Function, *Mathematics of Control, Signals, and System*, *2*, 1989, 303-314.

[6] C. Pappis, & E. Mamdani, A Fuzzy Controller for a Traffic Junction, *IEEE Transactions on Systems, Man and Cybernetics, SMC-7*, 1977, 707-717.

[7] D. Teodorović, Invited Review: Fuzzy Sets Theory Applications in Traffic and Transportation, *European Journal of Operational Research*, *74* (3), 1994, 379 - 390.

[8] J. Holland, *Adaptation in Natural and Artificial Systems* (Ann Arbor, MI: University of Michigan Press, 1975).

[9] M. Gendreau, G. Laporte, & R.Seguin, R., Stochastic Vehicle Routing, *European Journal of Operational Research*, *88* (1), 1996, 3-12.

[10]**http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/tsp/**