

On-line Signal-Plan Generation for Centralized Control Using Neural Networks

Rajeev Kumar Saraf¹ and Douglas H. Fisher²

1 Introduction

Traffic congestion has become one of the most noticeable problems of urban society. It is widely acknowledged that simply building new roads will not eliminate the congestion problem. Consequently, there is considerable emphasis on better utilizing available resources. In particular, well coordinated on-line traffic control systems promise to increase traffic capacity.

There are three main strategies used for traffic control. A *fixed-time* strategy is used when the traffic control, calculated for average conditions, is independent of the time and of the actual characteristics of the traffic flows. *Signal plan selection-based* strategy selects an appropriate plan (i.e., cycle length, offsets and phase splits) from a library of precalculated plans, where this selection depends on time-of-day and/or day-of-week, or it may operate in a traffic-responsive mode that is based on the flow measurements from traffic detectors. Finally, a *signal plan generation-based* control generates a new signal plan on-line based on traffic flow characteristics. Plan generation differs from plan selection, in that the former does not select from a predefined library of plans. Rather, plans are generated in real-time by some procedure.

These basic strategies can be implemented under different computer-controlled

¹ Department of Civil Engineering, Vanderbilt University, Nashville TN

² Department of Computer Science, Vanderbilt University, Nashville TN

architectures. A *centralized computer control* uses a central computer to carry out the control tasks and data processing requirements. In contrast, depending on the size of the network to be controlled, a computer hierarchy may be equipped with each computer controlling a small group of intersections. This is a *hierarchically or partially distributed computer controlled* system. A typical hierarchy has three or more levels. These levels could correspond to a central computing system, microcomputers installed in spatially distributed, unmanned subcenters, and local microcomputers at the individual intersections [1]. Finally, *radically distributed computer control* provides each intersection with its own digital data processing unit. There are no control centers and the task of the microcomputer at each intersection is to solve the control problems at that intersection.

Two major dimensions along which traffic control systems vary have been described: the traffic control strategy and the computer architecture on which the strategy is implemented. Table 1 illustrates the various systems and research efforts that fall within each cell of this two-dimensional view. Our work has focused on developing traffic-responsive plan-generation systems for centralized and radically-distributed environments, though this article limits discussion to centralized control. Our approach uses adaptive neural network learning algorithms. In particular, we train a neural network to approximate the recommendations of TRANSYT-7F, a well-known plan generation system for centralized control. Theoretically, TRANSYT-7F could be used dynamically to find the "best" plan for a given situation, but the use of TRANSYT-7F creates serious problems in a real-time environment because of its extensive computational requirements. In contrast, neural network-based learning systems promise advantages in terms of computational cost. After sufficient 'training', the neural networks can rapidly generate

Table 1 State-of-the-art review in traffic control

Computer Control Architecture	Traffic Control Strategy		
	Fixed-Time	Traffic-Responsive Plan Selection and Modification	Traffic-Responsive Plan Generation
Centralized Control	TRANSYT [2] THEMES [3] SIGOP III [4] MITROP [5] FORCAST[6] CLC-Combination Method[7] UTCS 1-GC [8] SIGRID SIGMA [9]	FLEXIPROG [10] EQUISTAT [11] Ambrosino et al in [12] SAGE [13]	DYNAMIC PLAN generation [14] PLIDENT [15] ASCOT[16] RTOP [17] UTCS 2-GC and 3-GC [8] Tokyo Multi-criterion method [18] SCOOT [19] Band transformations method [20] Incremental Coordination using TRANSYT-7F [21] CALIFE [22] Neural Network based control [23]
Hierarchical Distributed Control			PRODYN-H [24] UTOPIA [25] Drouin et al. in [26] Drenick and McShane in [27] Lin et al. in [28] Lee and Saridis in [29] Greenberg et al. in [30] SCATS [31]
Radically Distributed Control			Heydecker in [32] Bell in [33] OPAC [34] PRODYN-D [35] OFSET [36] Connectionist model [37] Iba [38]

new signal timing plans in response to changing conditions.

In sum, our approach uses TRANSYT-7F to generate signal plan recommendations off-line for varying traffic conditions. These plans, together with the corresponding traffic conditions, are used to train a neural network. Once trained, the neural network can be used on-line to generate plans dynamically. This paper investigates the plausibility of this approach. Specifically, we evaluate the ability of a trained neural network to approximate TRANSYT-7F's recommendations, evaluate the recommendations of each system in terms

of a performance index measure, and provide some experimentally-supported speculation on neural-network advantages along other dimensions such as transition cost³ (see [39]).

2 Learning to Control a Centralized Traffic System

In a centralized plan-generation system, a single central computer controls the network by generating signal timing plans for real-time, demand-responsive control. TRANSYT is the most commonly used software for (off-line) signal plan generation. The current version of TRANSYT is marketed as TRANSYT-7F in the U.S.A. TRANSYT-7F uses a hill-climbing optimization technique to produce a set of optimal signal control parameters, such as cycle length, phase lengths and offsets.

This section describes TRANSYT-7F in some detail. We then describe a neural network approach for learning TRANSYT-7F's situation-plan mapping. The result of learning is an on-line traffic control system that produces signal plans in a way that "mimics" TRANSYT-7F. For illustrative purposes, the applicability of the neural network model for optimization of phase splits for an idealized, isolated one-way intersection is demonstrated. Section 3 explores the application of neural networks for control of a real arterial road network.

2.1 Signal Plan Generator: TRANSYT-7F

TRANSYT-7F is a computer model to optimize the linking and timing of traffic signals in a network. TRANSYT-7F calculates a performance index (PI) based on a

³ Elsewhere, we describe a similarly-intended approach for a radically-distributed environment using OPAC to train a neural network [39].

deterministic, macroscopic traffic flow model. We assume PI is a linear combination of measures of effectiveness (MOEs) such as delays and stops. TRANSYT-7F uses an iterative, gradient search optimization procedure to find signal timings that (locally) optimize the PI. The signal settings corresponding to the best PI found is output as TRANSYT-7F's signal plan.

2.1.1 Input

TRANSYT-7F assumes a road network and traffic flow data as input. Traffic flow data consists of traffic flow estimates for all movements for every link in the network. Saturation flows are required for each separately identified stop-line. Link cruise speeds need to be known in order to calculate platoon dispersion. Since TRANSYT-7F only permits a fixed-phase sequence mode of operation, the user must specify the number of phases and their sequence. The performance criteria are largely based on stops and delays and the user-specified weighting factors applied to each criterion. Traffic flow data are the only variables that vary with time unless a change in policy is implemented, such as adding new lanes or changing the speed limit.

2.1.2 Signal Timing Optimization

The objective function for optimization is called the Performance Index (PI). TRANSYT-7F tries to find a signal plan that minimizes the value of the PI. For the purpose of this research, PI is assumed to be a linear combination of delay and stops [40]. There are three main decision variables in a signal plan: system-wide cycle length, relative offsets and phase lengths at each of the intersections. TRANSYT-7F uses a two-step optimization process to arrive at the values of each of the decision variables. In the first step, TRANSYT-7F carries out a "quick" hill-climbing optimization to evaluate

cycle lengths over a user specified range of values; the user also specifies that only values at fixed increments apart within this range should be examined. The system selects the cycle length with the best PI among those examined as the system-wide cycle length. In this "quick" optimization process, TRANSYT-7F optimizes offsets and phase lengths for each cycle length, but not as extensively as in the next step.

In the second step, after the "best" cycle length is selected, TRANSYT-7F carries out a "normal" hill-climbing optimization to first arrive at a set of offset and phase lengths. In this step, TRANSYT-7F goes through an iterative process of optimizing offsets and phase lengths one after another for each of the nodes. At the beginning of the optimization process, TRANSYT-7F calculates the initial phase lengths by equalizing the degree of saturation on conflicting critical movements. The set of initial timings represent one point in the solution space of the optimization problem. The hill-climbing approach uses this set of signal timings as the starting point in the search for the best solution. The search moves in a direction that reduces the value of PI compared to the PI value of the current point, and is continued until the PI value cannot be reduced any further by moving in any direction. The time taken to find the signal timings corresponding to the best PI value would depend on the amount that the current signal timing can be changed during each iteration of the search process. A large step size would require less execution time than a small step size, but the former is more prone to coming to rest at local minima. To overcome these problems, the process is initially started with large step sizes and then the step size is reduced as search progresses.

Note that the hill-climbing optimization is a heuristic approach that does not guarantee a global optimum. Moreover, the cycle length evaluation step only evaluates

a certain number of cycle lengths based on the cycle length range and the cycle increment specified. As such, TRANSYT-7F selects the best cycle length among the cycle lengths that were evaluated. In general, there may be other cycle lengths in the range that are better than the "best" cycle length computed by TRANSYT-7F.

2.1.3 Output

The output of TRANSYT-7F is a network-wide signal plan that can be implemented to control the traffic flow through the network. The signal plan consists of a network-wide cycle length, phase splits for all the phases at each of the signalized intersections, and offsets for all the intersections with respect to the master intersection. Signal plans are generated off-line for different traffic flow situations based on the historical data and stored in a plan library for future retrieval. The switching from one plan to another is based on the time-of-day (TOD) and/or day-of-week (DOW), or on the actual traffic flow measurements from detectors at key locations in the network. A real-time constraint makes TRANSYT-7F's ability for on-line plan generation problematic.

Instead, an approach that employs neural nets to learn TRANSYT-7F's Input/Output function is proposed. The neural network may then be used to rapidly generate new signal timing plans in response to changing conditions. Before elaborating the proposed traffic control strategy, the neural net model that has been adopted in this research is briefly described.

2.2 Neural Network Model

Artificial neural networks are biologically-inspired computational models that make decisions based on acquired knowledge. Network computation is performed by a mesh

of non-linear, computational units and inter-connections that resemble biological nerve nets. The basic processing units of neural networks are called artificial neurons or nodes.

Artificial neurons perform as summing and nonlinear mapping junctions. Each neuron accepts a set of inputs, each of which may represent the output of a 'preceding' neuron or may be directly supplied by the external environment. Each input is multiplied by a corresponding weight and a weighted sum, NET, is computed. This sum is then passed through a nonlinear mapping function to determine the activation level or output, OUT, of the neuron. Figure 1 shows an artificial neuron that uses a sigmoidal nonlinearity. The sigmoid function is mathematically expressed as,

$$OUT = \frac{1}{(1 + e^{-NET})} \quad (1)$$

Neurons usually operate in parallel and are often organized in layers. The neurons in the neural network are of three types: (1) those that receive input from the external

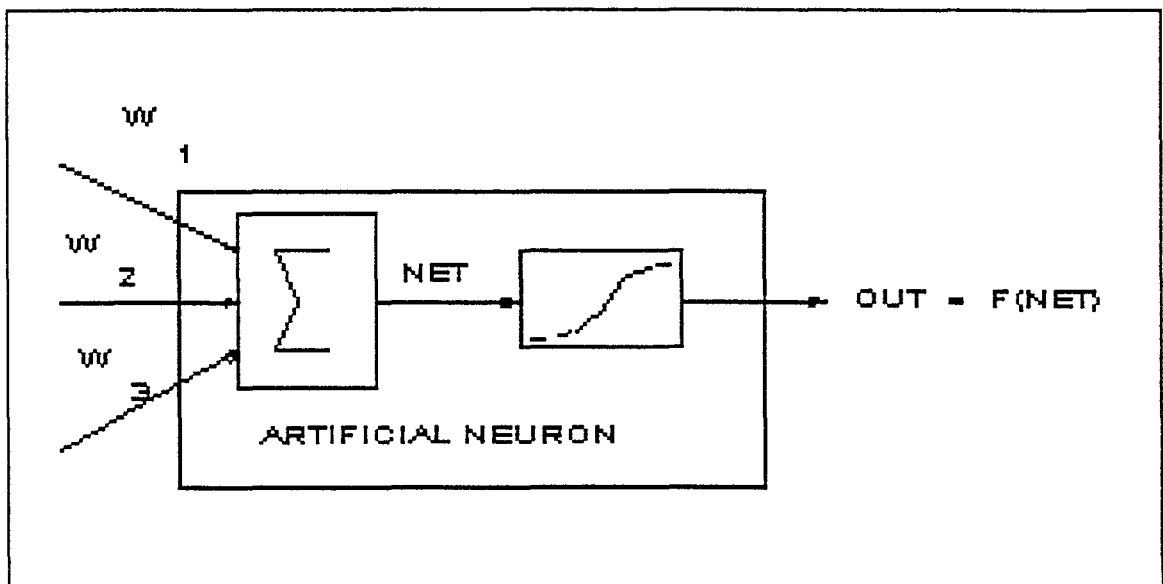


Figure 1 Artificial Neuron

environment (i.e., input units), (2) those that provide output to the environment (i.e., output units) and, (3) those that are internal to the model (i.e., hidden units). Neurons are arranged into layers: an input layer, an output layer, and zero or more hidden layers. For this research, only three layered nets as shown in Figure 2 are employed. The networks that are used in this research are also feed-forward networks, meaning that activation is directed strictly from the input layer, through the hidden layer, to the output layer. Neurons in each layer have a connection with neurons in the adjacent layers whose strength is expressed by a numerical weight.

Neural network models are specified by the net topology, node characteristics, and a training or learning algorithm. The training algorithm establishes an initial set of weights and determines how and when the connecting weights change to improve performance. The goal of the learning algorithm is to train the network to generate desired output vectors for corresponding input vectors. The training is done over sets of input and desired output (target) vectors. Backpropagation [41] is one of the most widely

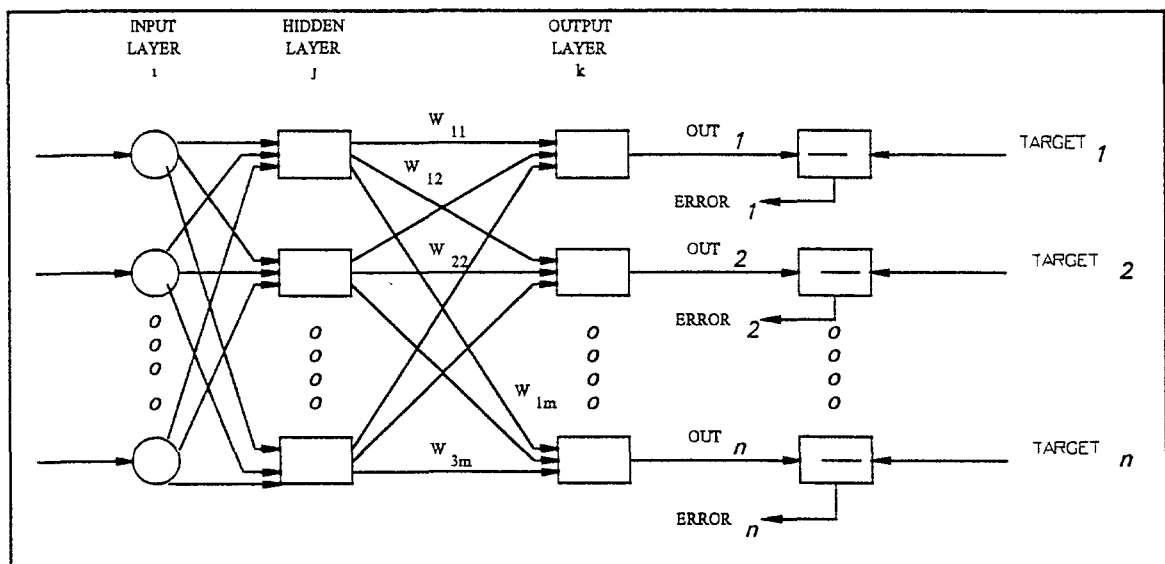


Figure 2 A multi-layer backpropagation network (From: Weiss and Kulikowski, 1991)

used approaches for training a neural network and will be used for training the traffic neural network in this research.

The training in Backpropagation consists of two steps:

1. Forward pass - In this step, each neuron in a layer computes a function of its inputs and passes the resultant activation value as its output to the next layer. Thus, the network output is generated by propagating the initial inputs through the layers of the network.

2. Backward pass: In this step, the network's outputs from the forward pass are compared with a desired or target output vector obtained from a "trainer" or "teacher". The error between the actual network output and the desired output is "backpropagated" through the network, while updating the weights in the direction that would most reduce this error.

Here, the approach is to train a neural network to approximate the recommendations of TRANSYT-7F. Thus, over a large number of input vectors, which represent differing traffic situations, TRANSYT-7F's recommendations will be used as the target output vectors. As mentioned earlier, traffic flow variables are the only dynamic variables. All other inputs for TRANSYT-7F remain static over time. This implies that for a given road network, the neural network only needs to learn the mapping between the traffic flow demand and the signal timing parameters for the various intersections of a road network. For example, Figure 3 illustrates a neural network appropriate for control of the one-way two-phase signalized intersection shown in Figure 4. The input layer represents the traffic demand on the network, i.e., the flow on each link, and the output layer represents the signal plan timing parameters for the network, i.e., cycle length, phase

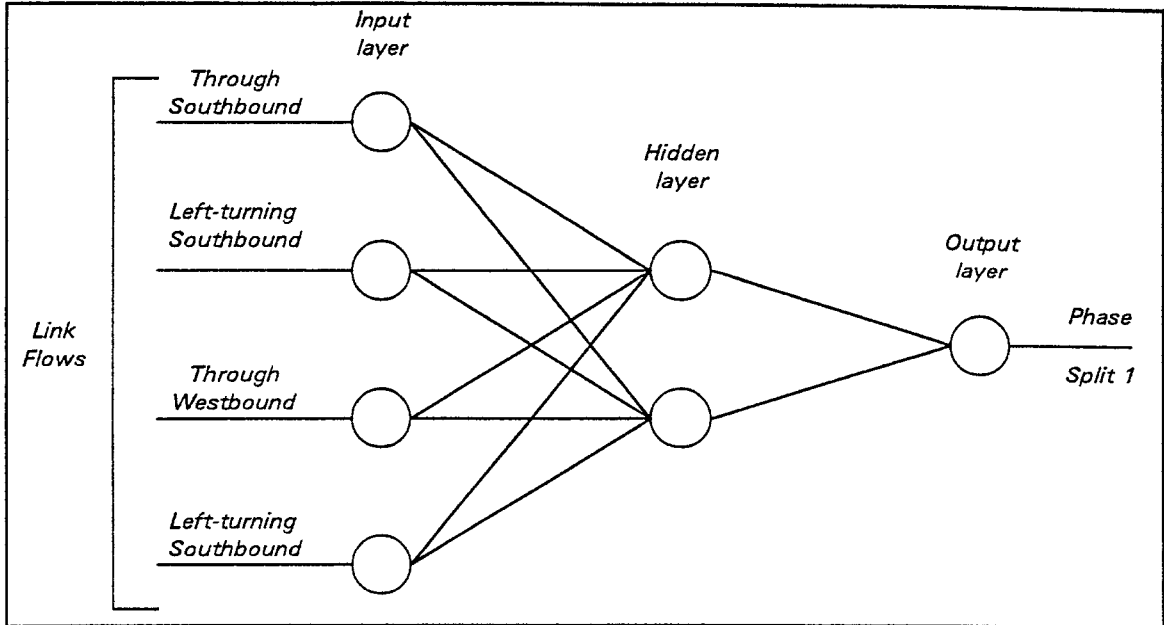


Figure 3 Neural network architecture for phase splits estimation for the isolated intersection

lengths and offsets. For the one-way single intersection network, there are four link flows - through southbound, right-turning southbound, through westbound and left-turning westbound. The intersection has two phases, i.e., one each for the southbound and westbound movements. Assuming that the cycle length is fixed, only one of the phase lengths need to be calculated.

Traffic demands can be determined from historical data or from detector measurements. This demand is provided as an input to signal-plan generation software, such as TRANSYT-7F, to produce a signal timing plan. Each pair of traffic demand and the corresponding signal plan constitutes a training pair, or an exemplar. A set of such exemplars constitutes a training set.

The traffic neural network was trained to learn the mapping between the input and output of TRANSYT-7F using the training set. Backpropagation was used as the training algorithm. Ideally, the trained neural network will generalize the relationship between the

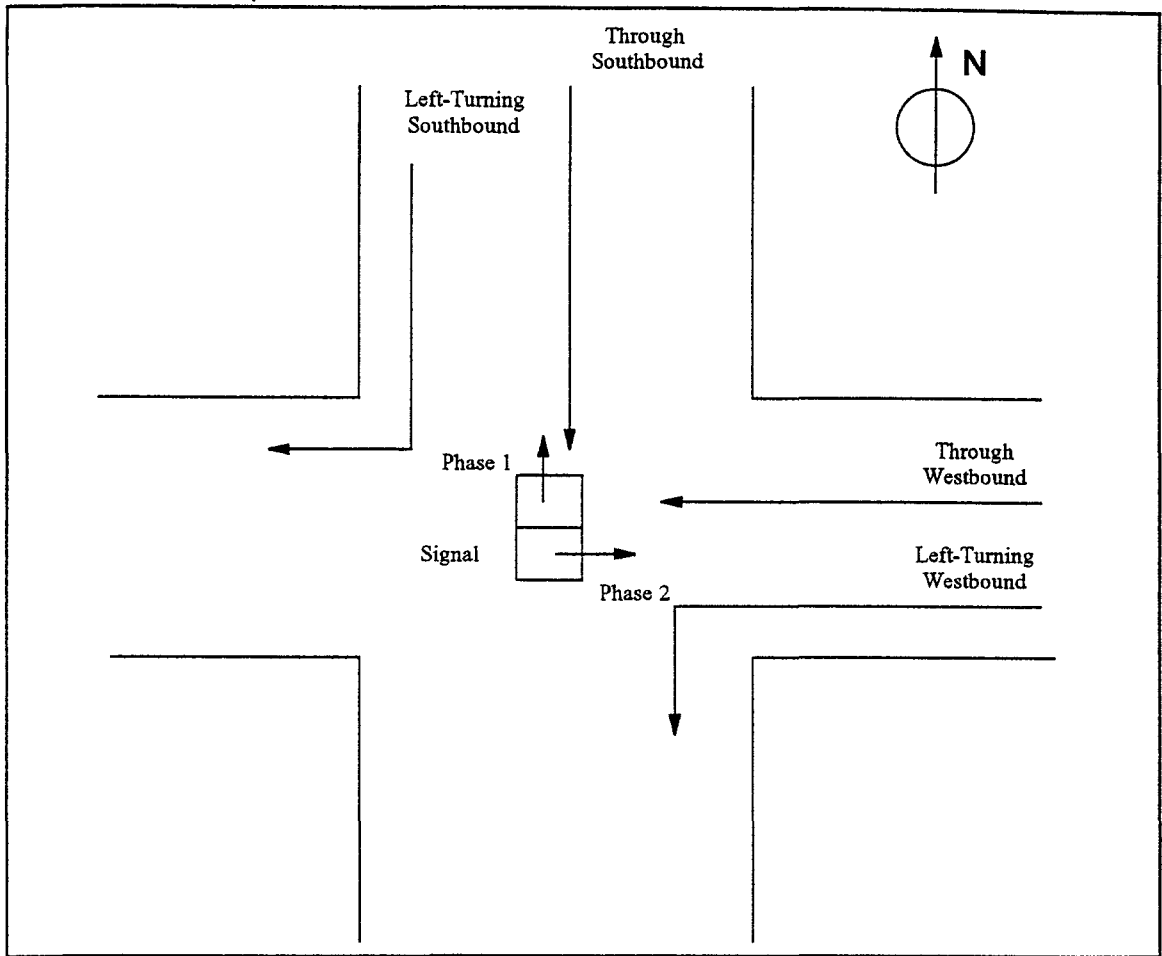


Figure 4 An isolated one-way intersection with fixed cycle length

traffic flow demands and the signal timing parameters, and produce a response for any kind of input pattern that is "consistent" with its training data.

Traffic control problems in the real world are not as idealized or simple as the example discussed in this section. Traffic networks consist of many intersections and the number of input flow variables and the output signal plan variables expand with the number of intersections. Before neural networks can be accepted as a viable tool for real-time traffic control, they must be tested on more complex, real-world road networks. In the next section the learning ability of a neural network for a real, arterial road network is explored.

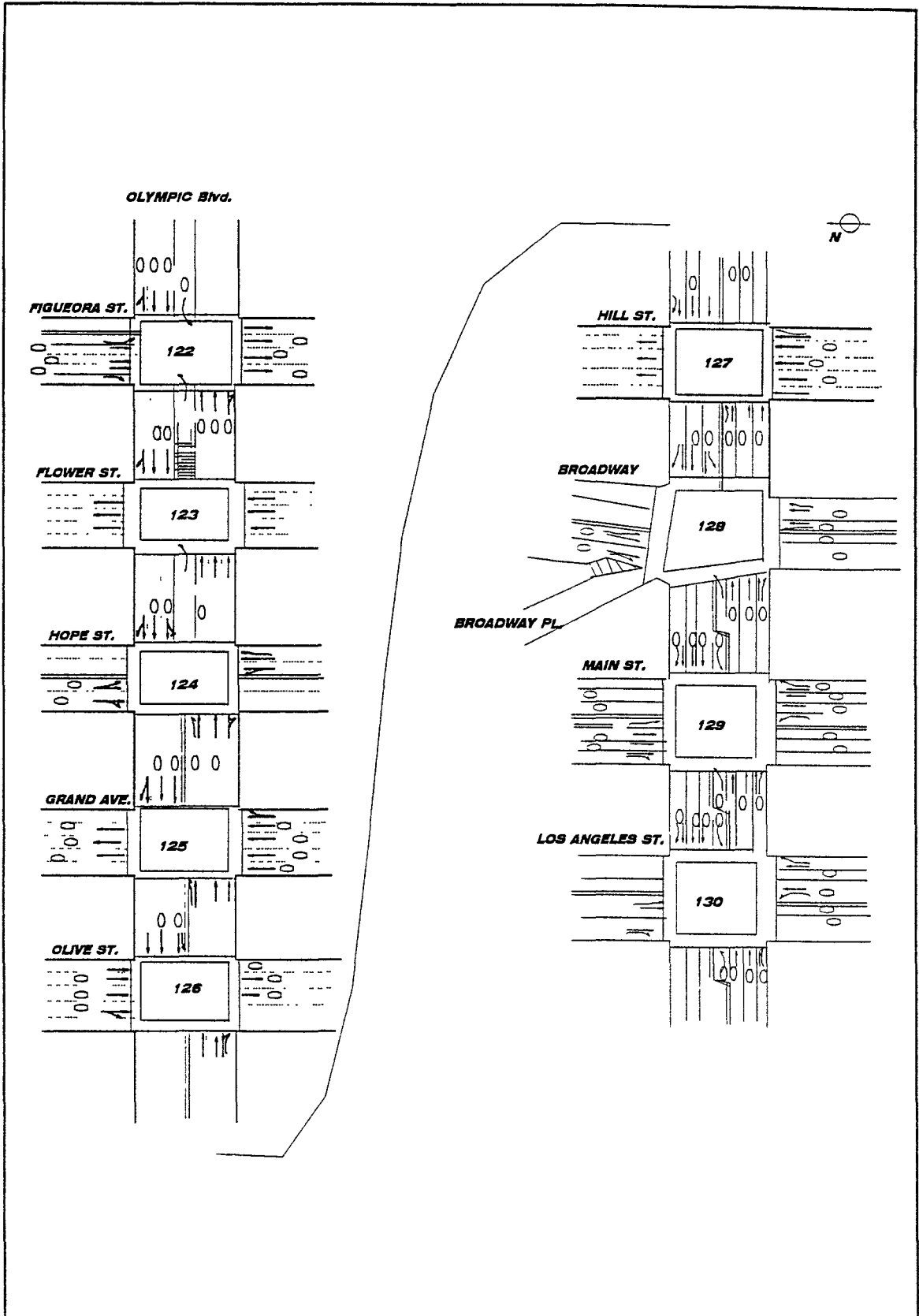


Figure 5 Topological layout of Olympic Blvd. arterial

3 Centralized Learning with a Real Arterial Road Network

In this section, the learning approach is applied to a realistic traffic network in urban areas - an arterial road network. Here, the main objective is to use a neural network model for optimization of system-wide cycle length, phase splits and offsets relative to the master controller.

3.1 An Arterial Road Network

The arterial shown in Figure 5 is selected as the case study. The arterial chosen is the section of Olympic Blvd. in Los Angeles having 9 signalized intersections crossing Los Angeles St. at the west end and Figueroa St. at the east end. The traffic data was collected by the Los Angeles Department of Transportation using surveillance detectors. Not all the approaches to each of the intersections were detectorized. Therefore, an estimate of the missing data was made using the information collected from other detectors. Whenever not available, the turning movements were assumed to be 10% of the total flow on the approach to the intersection. This was decided based on discussions with the traffic operations personnel at Los Angeles Department of Transportation. The traffic counts data were accumulated for every detector at 15-min intervals for a typical workday. Since traffic data was available for only one day, there were 96 (i.e., 24 x 4 quarters) different time period data points.

3.2 Plan Generation Using TRANSYT-7F

TRANSYT-7F was used to calculate the signal plans for the varying traffic

conditions at each of the 96 quarter hours. The cycle length was assumed to vary between 30 seconds and 120 seconds with an increment of five seconds, to reflect the practice of traffic engineers. The MOEs used to compute the value of PI were average delay and number of stops. The PI was assumed to be a linear function of delay and stops. Normal optimization was used with default optimization steps. The yellow period and all-red period were provided a default of three and one seconds, respectively. TRANSYT-7F's default step sizes were used in the hill climbing routine. Saturation flows suggested by TRANSYT-7F for normal conditions (i.e., average driver behavior pattern) were used. No separate pedestrian phases were provided and no explicit mid-block flow sources or sinks were coded. TRANSYT-7F automatically adjusted the input flow rates to sum to the total flow rates.

3.3 Initial Neural Network Architecture

A signal timing plan consisted of system-wide cycle length, phase lengths for all the phases at each of the signalized intersections, and offset of each intersection with respect to the master controller. For the given arterial, there were nine intersections, with intersection #122 (i.e., intersection of Olympic Blvd. and Figueroa St.) assumed to be the master intersection. Each intersection had two phases. So for the arterial there was one cycle length, 18 phase lengths and 9 offsets. All together, there were *28 output variables*.

For any road network, link flows are the only variables that we assume continuously vary with time. Other variables such as network topology are assumed to be constant. A "link" does not refer to each lane or each approach to the intersection, but refers to the traffic flow movements at each of the intersections. The link definition here

is synonymous with TRANSYT-7F's definition. In other words, the number of traffic flow variables for a neural network is the same as the number of links that would be defined in TRANSYT-7F for a given arterial. There are 55 links in the Olympic Blvd. arterial segment, i.e., there are *55 flow-based input variables*.

The problem of the optimal hidden layer configuration is still an open question and in general, a large number of simulation runs may be required to arrive at the best configuration. However, some guidelines are available for selecting the number of hidden nodes. Mirchandini and Cao [42] argue that the number of hidden units J depends on the dimension n of the input vector and on the number of separable regions M in the n -dimensional Euclidean input space. Suffice it to say that their analysis concludes that a reasonable number of hidden units J can be given by $J = \log_2 M$. For the current case study, we assume that each of the 96 15-min traffic volume data counts on any day belong to a different separable region and that traffic patterns on weekdays versus weekends belong to different regions. Thus, the suggested number of hidden units would be $\log_2 (192) = 7.58 \approx 8$. Therefore, *8 hidden units* were selected for the initial experimentation.

This network configuration with 55 input units, 8 hidden units and 28 output units is used in initial experiments. However, experiments reported in [39] show that our general results are robust with respect to differing number of hidden units.

3.4 Experimental Design

The input traffic flow values for the neural network were linearly scaled between zero and one by dividing the traffic flows by the maximum possible flow. The target

cycle length generated by TRANSYT-7F was also scaled down by a factor of 120.0, i.e., maximum value of target cycle length, to limit the neural network's output to 1. The target phase lengths and the target offsets values from TRANSYT-7F were calculated as the percentage of the cycle length. These values were then scaled down by dividing by 100.0.

The neural network model does not explicitly enforce the minimum phase length constraint and the cycle length constraint, i.e., the sum of the predicted phase lengths must be equal to the predicted cycle length. Therefore, the output generated by the neural network needs to be processed before being used as an input to a simulator for evaluating performance. A post-processor was designed that converted the signal plan values predicted by the neural network into a format that could be used by the simulator to evaluate the performance of the predicted signal plan. The predicted cycle length was kept unchanged, and the phase error, defined as the difference between the predicted cycle length and the sum of the predicted phase lengths, was proportionally distributed among the predicted phase lengths. The error distribution was weighted by the individual phase length values, i.e., the phase having a larger value of phase length got a larger share of the error.

To effectively use the limited data, a ten-fold cross validation error estimation technique was employed [43]. The advantage of cross-validation is that all the cases in the available database are alternately used for testing and for training the neural network. The data is randomly partitioned into 10 mutually exclusive subsets of approximately equal size. The training data for the Olympic Blvd. case study was randomly divided into 6 subsets of 10 samples and 4 subsets of 9 samples each. Ten-fold cross validation

requires ten experimental trials; in each, a different subset is used for testing, and the remaining 9 subsets are used for training. Thus, in each experiment the network is tested on data not used for training. Collectively, over the 10 experiments, each exemplar is tested on a network that did not use the exemplar for training. The test set results for the ten experiments are combined to arrive at the network's performance statistics. The output from the neural network for the test data was then compared with the target output generated by TRANSYT-7F. The average test error rate over all 10 experiments is the cross-validation error rate. In each experiment the neural network was trained for 10,000 iterations through the training data, since multiple iterations through the data are required, as in almost all applications, to suitably train the network using Backpropagation.

3.5 Experimental Results

In this section, the performance of a trained neural network as a predictor of the traffic signal plans for varying traffic conditions is evaluated. The performance of the neural network is evaluated in terms of the PI of the predicted signal plans and the ability of the neural network to predict the value of one of the more important output variables, cycle length.

As noted in Section 3.3, the traffic flow on each link in the roadway arterial is used as the only input to the neural network and the neural network is trained to predict signal timing settings (i.e., cycle length, offset and phase intervals) for each intersection of the arterial segment.

3.5.1 Performance Index

As we noted earlier, the performance of any signal plan for a road network can

be evaluated by assessing the PI. Two signal plans can be compared by comparing the respective PIs. A traffic simulator such as TRANSYT-7F, can be used to evaluate the PI for a given plan, as well as to generate plans intended to optimize PI. In particular, in addition to using TRANSYT-7F to "train" the neural network by generating plans for training data, TRANSYT-7F was used to evaluate the plans generated by the neural network on test data after training. As we noted, the PI was calculated as a linear function of the number of stops and delays. Henceforth, the PI value for the plan predicted by the neural network will be referred as the *Predicted PI* and the PI of the best plan generated by TRANSYT-7F as the *Target PI*.

3.5.1.1 Aggregate PI

The performance of a control strategy - either TRANSYT-7F or a neural network - for a typical day can be evaluated by summing the PIs of the signal plans implemented for each of the time periods over the day and then comparing these aggregate PI values with each other. The PI values of the individual signal plans generated by TRANSYT-7F and predicted by the neural network were calculated by simulating the operation of these plans under TRANSYT-7F. For the case study here, the aggregate performance index value for TRANSYT-7F's generated signal plans was 8611.54. The aggregate performance index for the signal plans predicted by the neural network was 8544.43. This comparison predicts a 0.779 % reduction in PI when the road network is being controlled by a neural network-based centralized control system compared to TRANSYT-7F based control. The aggregate PI results indicate that even though TRANSYT-7F is used to train the neural network, a signal plan generated by the neural network can perform better than the signal plan generated by TRANSYT-7F. This is

possible for a number of reasons:

a. TRANSYT-7F does not carry out an exhaustive search in the solution space. The optimization engine uses a hill-climbing approach to arrive at the "optimal" signal plan. This approach does not guarantee a global optimal solution.

b. While generating signal plans for training neural networks, the cycle length values were constrained to lie between 30 seconds and 120 seconds with five-second increments. These constraints were imposed to represent common practice of traffic engineers. Use of these constraints forced TRANSYT-7F to evaluate only certain cycle lengths within a permissible range of cycle lengths. Thus, the cycle length selected by TRANSYT-7F may not be the best cycle length in the specified range. The cycle length output value of the neural net may 'naturally' interpolate between these 5 sec. increments on which it was trained. In any case, it is not limited to cycle outputs at 5 sec. intervals.

In summary, a comparison of the aggregate PIs of plans generated by the neural network and TRANSYT-7F reveals comparable behavior. Further insight can be gained by comparing plans for individual test cases.

3.5.1.2 Correlation Analysis

The graph in Figure 6 shows the values for Target PIs and the Predicted PIs on test data as a function of time-of-day. With the exception at approximately 1400 hrs the Predicted PIs are comparable to the Target PIs over all the 96 quarter hours. The same data is viewed as a scatter graph in Figure 7, which shows a strong positive correlation between the Target PIs and the Predicted PIs. A linear regression between the Target PI and Predicted PI values reveals a correlation coefficient value of 0.98. Such a high correlation coefficient indicates that the neural network's performance 'mirrors' that of

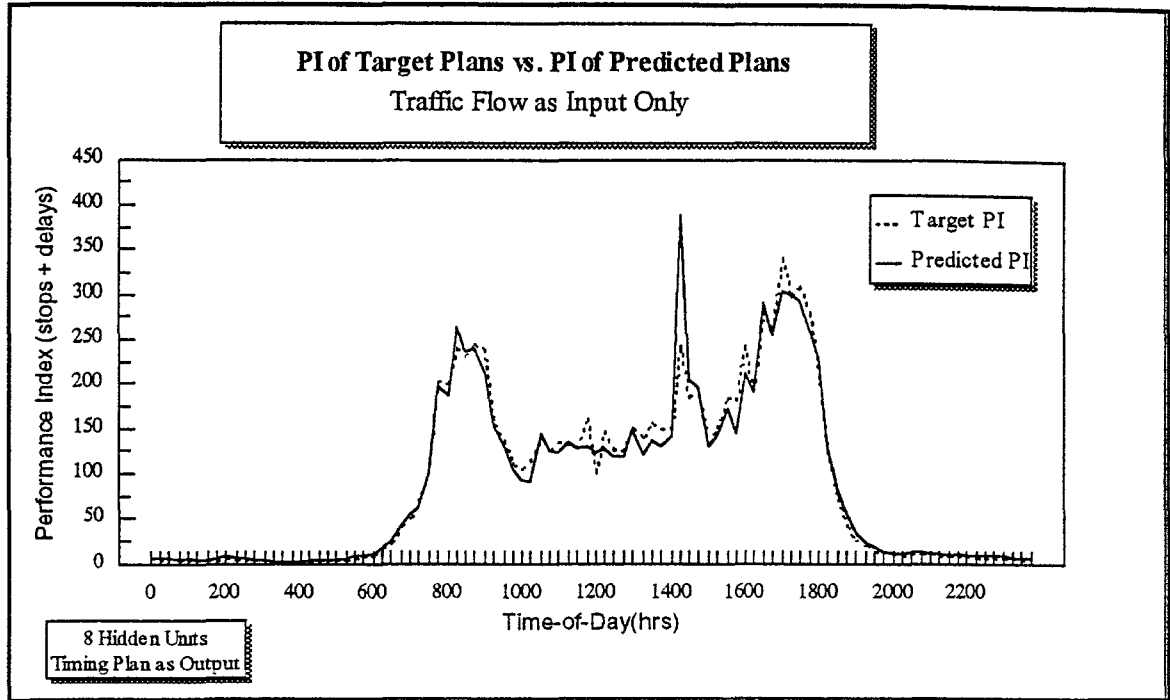


Figure 6 PI of target and predicted signal plans as a function of time-of-day

TRANSYT-7F's optimizer across the data.

The results thus far indicate that the quality (PI) of signal plans generated by the neural network accurately reflect the quality of TRANSYT-7F plans, but no evaluation has been made to estimate the neural network's performance for individual outputs.

3.5.2 Cycle Length

There is little value in comparing the target and predicted values for each of the 28 output variables. However, one of these outputs - cycle length - is known to have a significant effect on the performance of a signal plan. The other signal plan variables - offsets and phases - are also functions of cycle length. A cycle length smaller than the "optimal" cycle length tends to increase the number of stops and a cycle length larger than the "optimal" cycle length tends to increase the delay. Intuitively, this suggests that the predicted cycle length of the neural network should not differ significantly from

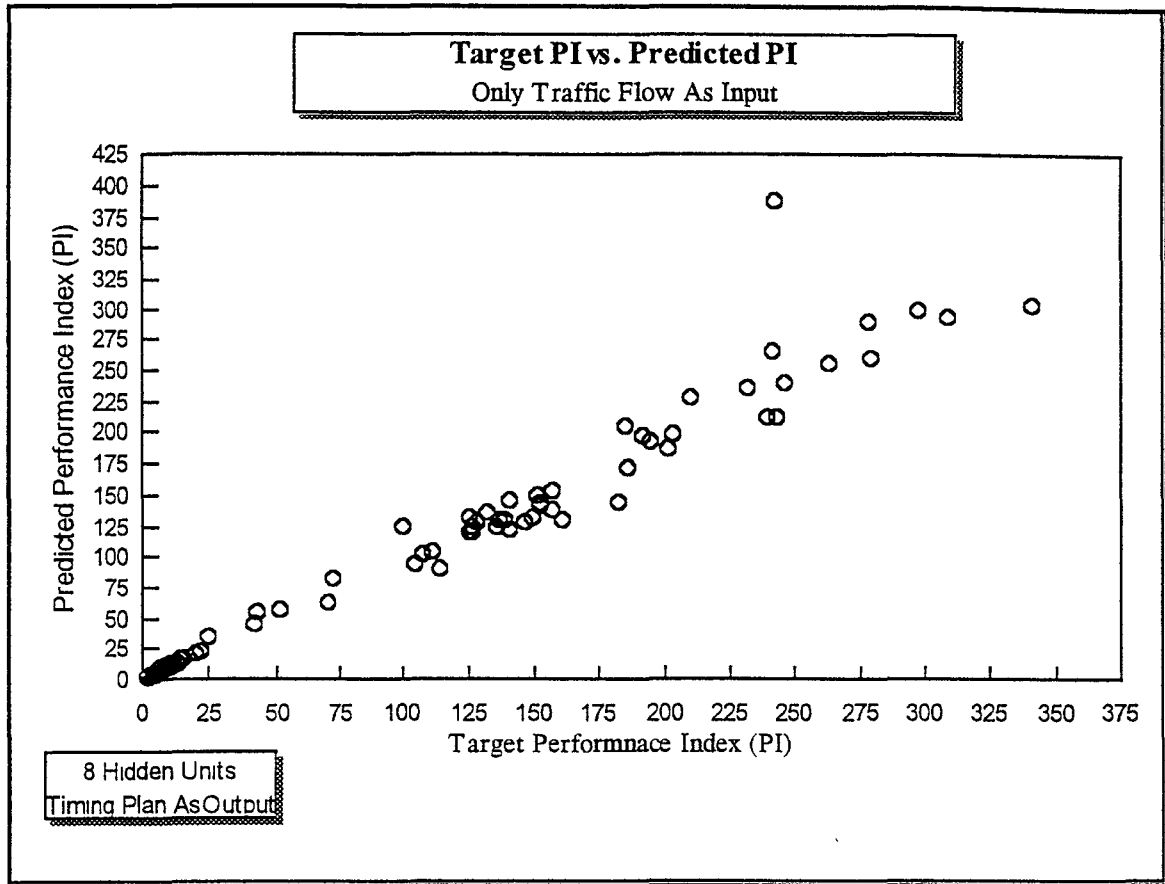


Figure 7 Scatter plot of PI of target and predicted signal plans

TRANSYT-7F's suggested cycle length, particularly given that their PIs are so well correlated. Time will be taken to compare the target cycle lengths generated by TRANSYT-7F and the neural network predicted cycle lengths for the same input traffic conditions.

3.5.2.1 Correlation Analysis

The graph in Figure 8 shows the variation in the value of predicted and target cycle length as a function of the Time-of-Day (TOD). The graph in Figure 9 scatter plots the same data: target cycle length versus predicted cycle length. A correlation analysis between the target and predicted cycle length gives a correlation coefficient value of 0.62. The moderate positive correlation coefficient suggests a positive relationship between the

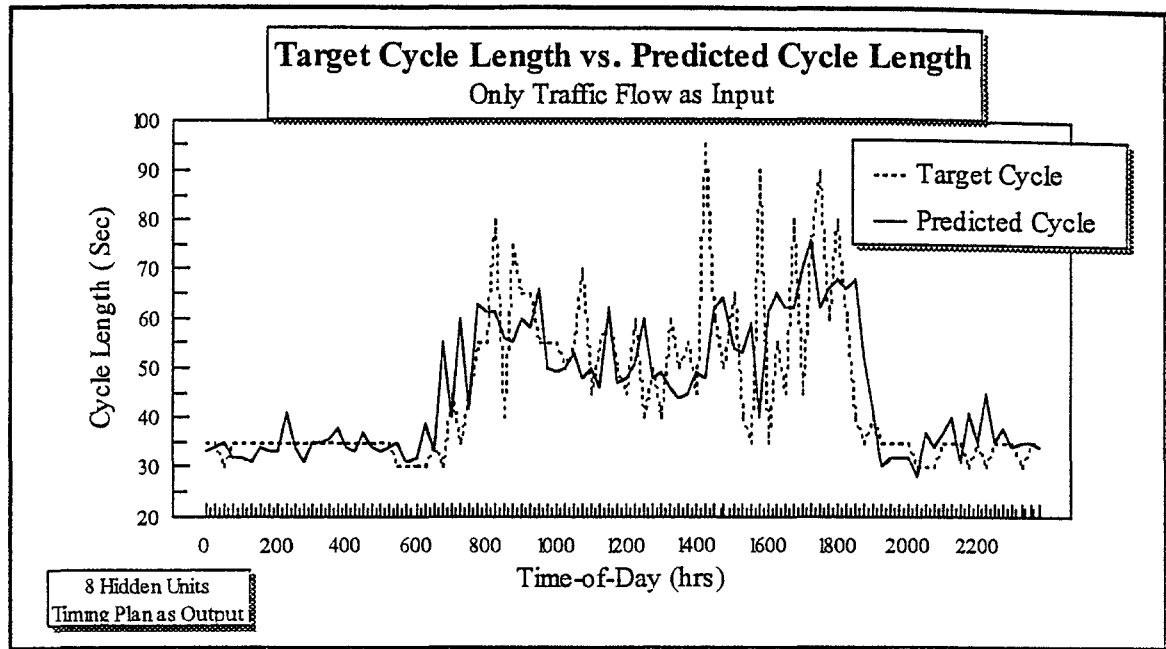


Figure 8 Target and predicted cycle length as function of time-of-day

predicted and the target cycle length. Nonetheless, as Figure 8 reveals most clearly, at certain points there are large gaps between predicted and target cycle lengths. The significance of this difference between the target and the predicted cycle length and its effect on the performance of the signal plan is analyzed next.

3.5.2.2 Cycle Sensitivity Analysis

Even though there is a moderate correlation between the target and the predicted cycle length, the results in Section 3.5.1 showed that there is a very small difference between the Target PIs and Predicted PIs. It is also known that the value of the performance index is highly dependent on the cycle length. This inconsistency motivates a deeper analysis between predicted cycle lengths and the corresponding PI values.

The dependency of the performance index value on cycle length is measured by a cycle length sensitivity parameter. Cycle length sensitivity is one of the outputs generated by TRANSYT-7F when a cycle evaluation is requested. Cycle sensitivity is

a measure of how sensitive traffic performance of the network is to changes in cycle length [40]. This measure is computed as follows:

$$CS = \frac{SD(PI)}{AVG(PI)} * 100 \quad (2)$$

where CS = cycle length sensitivity expressed in percent, which in statistical terms is the coefficient of variation;
 SD(PI) = the standard deviation of the PIs estimated by TRANSYT-7F for each cycle length evaluated; and
 AVG(PI) = the average value of the PIs estimated by

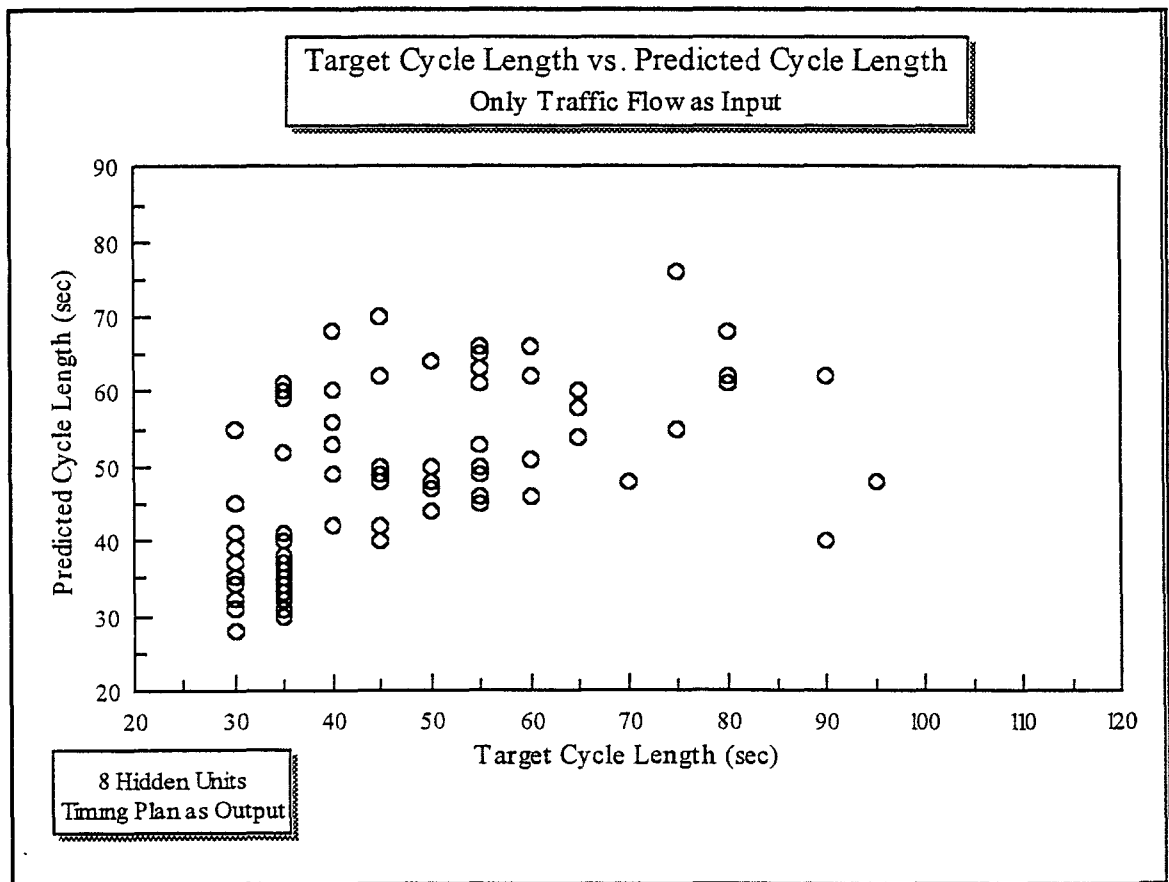


Figure 9 Scatter plot of target and predicted cycle length

TRANSYT-7F for each cycle length evaluated.

Relatively low values of the cycle length sensitivity parameter indicate that traffic performance is not particularly sensitive to cycle length in the range of cycle lengths evaluated; any of many cycle lengths evaluated by TRANSYT-7F would lead to roughly the same performance. A high value of the parameter would imply that the traffic performance will vary greatly with the choice of cycle length.

Cycle length "error" is defined as the absolute difference between the target cycle length produced by TRANSYT-7F and the cycle length predicted by the neural network after training. Figure 10 graphs the cycle length error as a function of cycle length sensitivity. A correlation analysis between the cycle sensitivity parameter and the cycle length error produced a correlation coefficient value of -0.40891 . The negative value implies an inverse correlation between the cycle sensitivity parameter value and the cycle

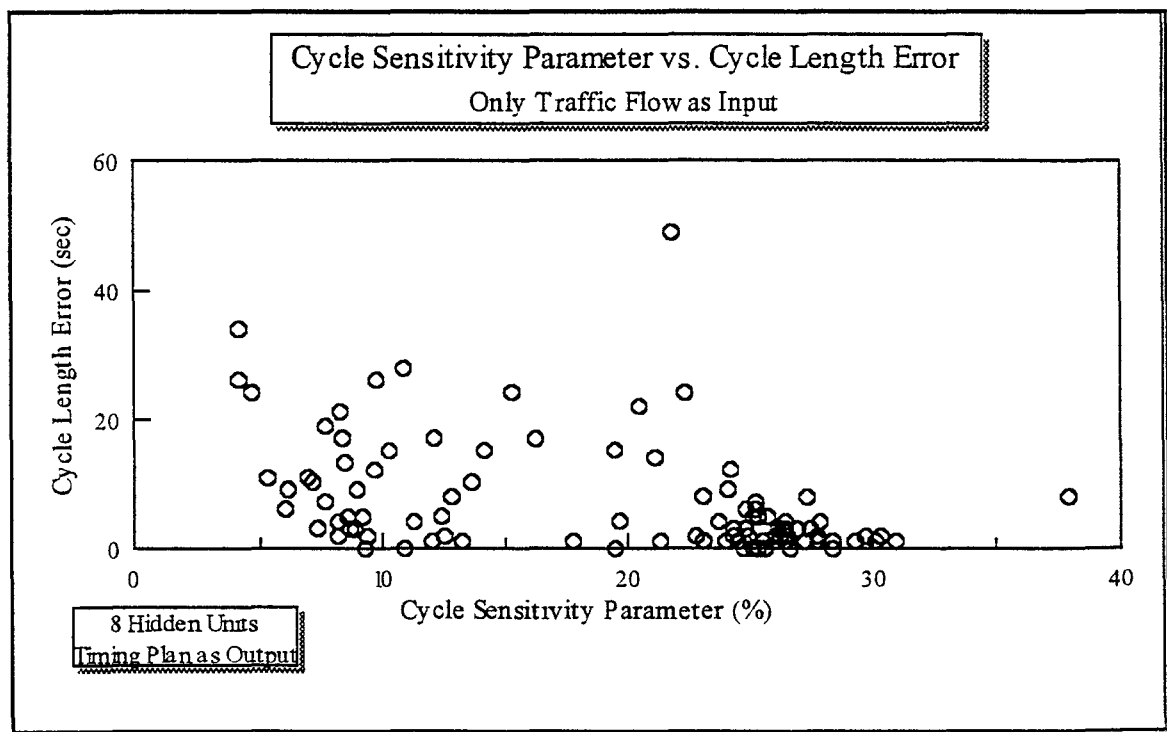


Figure 10 Scatter plot of cycle sensitivity parameter and absolute cycle length 'error'

length error. This means that cycle length values predicted by the neural network tend to have higher deviations from the target (optimal) cycle lengths when the cycle sensitivity values are lower. In other words, cycle length deviation tends to be greater when it matters less. This implies, as our results indicate, that there will not be substantial degradation in the performance of the signal plan predicted by the neural network compared to the signal plan generated by TRANSYT-7F, even though there may be relatively large deviations between predicted and target cycle lengths.

3.5.3 Transition Cost

Another important point to consider is the transition cost associated with varying cycle length. There is a transition cost due to increased vehicle delays over the next two to three cycles when the system wide cycle length is changed. Therefore, in practice, a system where the cycle length changes are minimal and the variations in the cycle length are smooth is desired. Figure 8 suggests that changes in predicted cycle length were not as drastic as the changes in the cycle lengths generated by TRANSYT-7F. Therefore, a hypothesis that the neural network tends to smooth out the values of the cycle length variable in a beneficial way, is postulated. In order to test this hypothesis, the predicted cycle length values were compared with smoothed target cycle length values obtained by taking the average of nine cycle length values corresponding to 9 time-of-day points, four on either side of the current time-of-day point. For example, if the current time-of-day is t then the smoothed cycle length at t is computed as the average of the target cycle lengths at $t-4$, $t-3$, $t-2$, $t-1$, t , $t+1$, $t+2$, $t+3$, $t+4$ time-of-day points.

The smoothed target cycle length values were compared with the cycle length

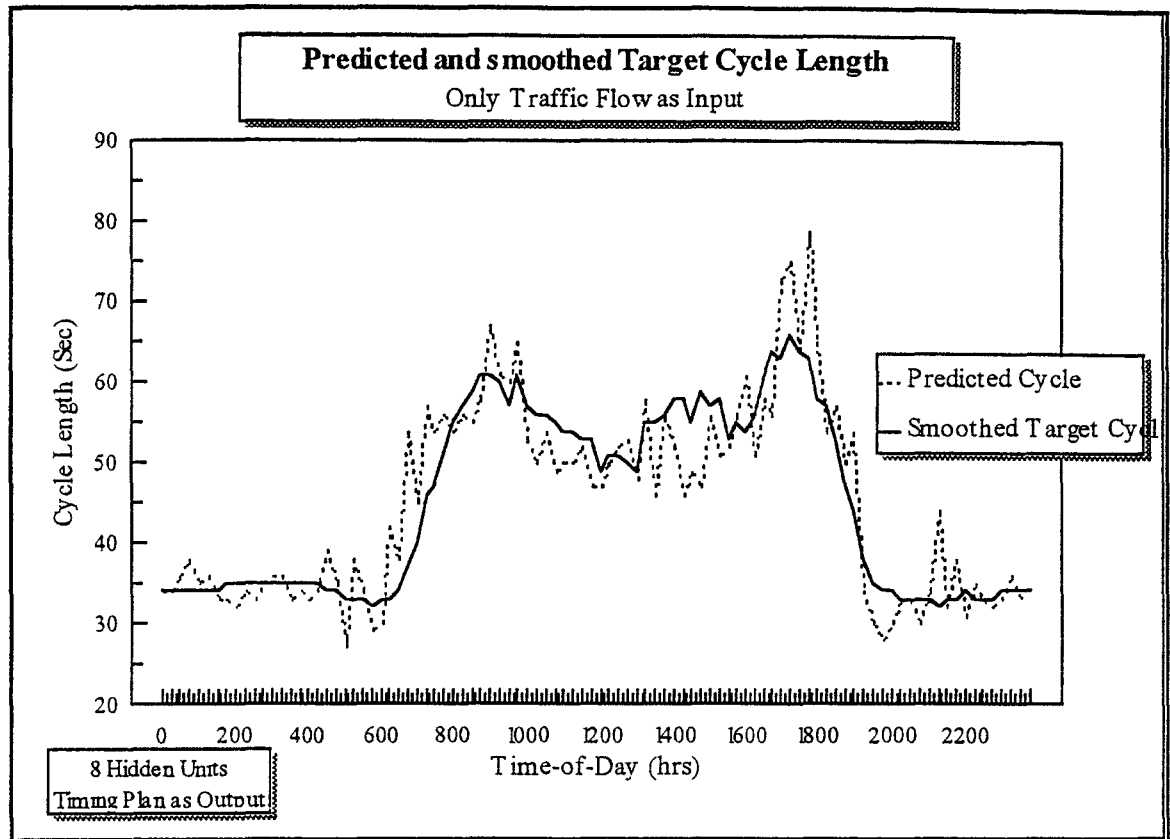


Figure 11 Smoothed target and predicted cycle length as a function of time-of-day

values predicted by the neural network. The graph in Figure 11 shows the variation of the smoothed target and the predicted cycle length as a function of time-of-day. The graph in Figure 12 scatter plots the same data. A correlation analysis between the smoothed target and the predicted cycle length gives a correlation coefficient value of 0.9029. The high correlation coefficient shows a strong positive relationship between the smoothed target and the predicted cycle length. This suggests that the neural network "naturally" smooths the cycle length in a manner similar to that of a conventional smoothing algorithm. In doing so, the neural network mitigates large variations in cycle length values between adjacent time points which, in turn, would reduce transition costs.

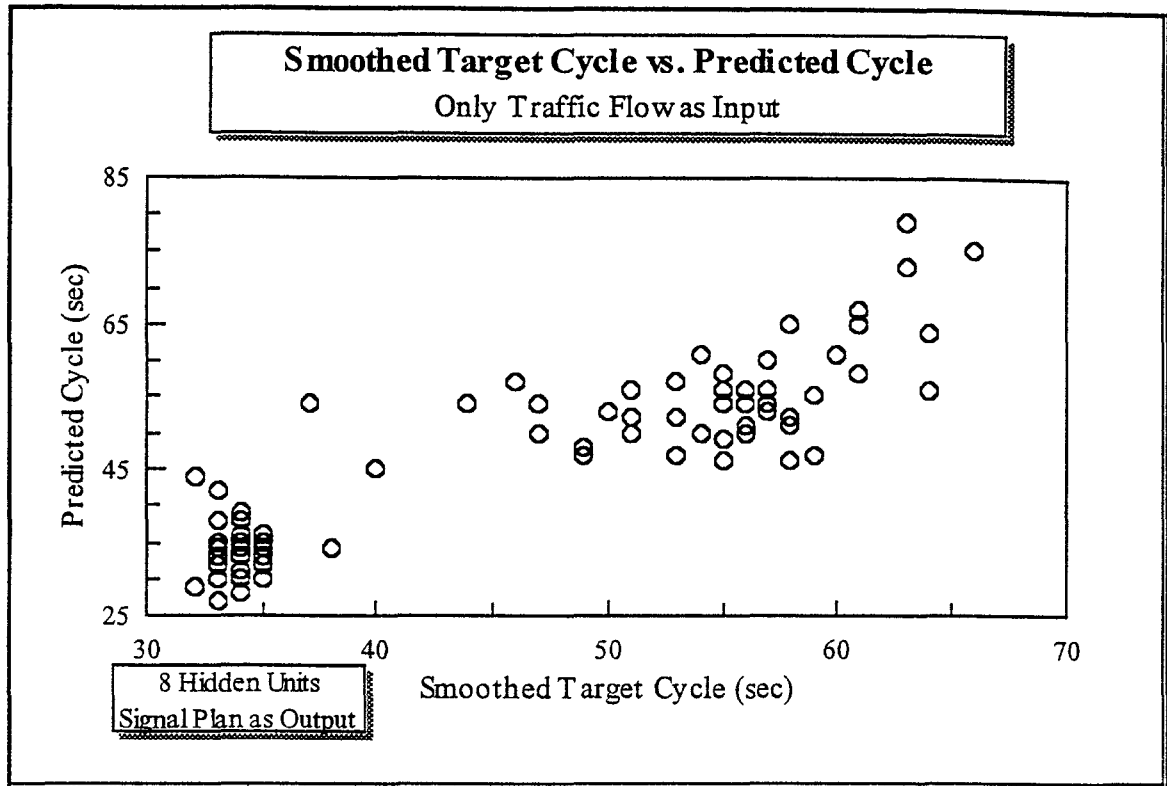


Figure 12 Scatter plot of smoothed target and predicted cycle lengths

3.6 Summary

Our experiments indicate that the neural network and TRANSYT-7F have comparable PIs under varying traffic conditions, and the neural network also reduced large fluctuations in cycle length, thus promising to mitigate transition costs associated with varying cycle length. Since the neural network "generates" (outputs) plans based on an efficient pattern matching procedure, instead of using optimization algorithms directly, the approach is promising for real-time traffic control - plans can be generated more frequently to meet continuous changes in traffic flow. Note that the cross-validation methodology insures that neural network performance has been measured on test exemplars - patterns that the neural network had not seen during training in each of the 10 experiments.

4 Concluding Remarks

TRANSYT-7F generates signal plans for off-line traffic control systems, such as TOD-based control. A real-time version of TRANSYT-7F, SCOOT, is also used in many cities. These systems appear to perform well, but they are also limited. TRANSYT-7F takes considerable time to generate signal plans and therefore, is not suitable for on-line plan generation. In contrast, SCOOT is constrained by the solution space it searches to determine the optimal traffic control parameters. A neural network-based approach does not suffer from these limitations. The network can be trained off-line using an unconstrained TRANSYT-7F simulator, and can subsequently generate plans using an efficient pattern matching strategy that is conditioned on current road conditions.

This paper takes an important step in determining the feasibility of using a neural network-based learning system to control a traffic network. The PI analysis in Section 3.5.1 showed that the neural network performed as well as TRANSYT-7F under simulated conditions. Other experiments reported in [39] varied neural network architecture and indicate that the learning ability is robust relative to the number of hidden units. We have also experimented with neural networks in the context of distributed control using OPAC as a trainer [39]. The same general advantages of neural network control noted here, are also found in the distributed context.

Collectively, efficiency and accuracy suggest the promise of neural networks for real-time plan generation in a centralized system, but more experiments are needed to determine the robustness of the approach to differing road network architectures. Neural networks seem to meet at least some of the requirements of a good traffic control system

and thus, show potential to be a viable on-line tool for generating signal plans for a centralized control system.

5 Acknowledgements

We would like to sincerely thank Mr. Kenny Quach of Los Angeles Department of Transportation (LADOT) for providing actual traffic data for our case study.

6 References

1. Strobel H., Computer Controlled Urban Transportation, John Wiley & Sons, 1982.
2. Robertson D.I., "TRANSYT: a Traffic Network Study Tool," Road Research Laboratory Report No. RL-253, Crowthorne, Berkshire, England: 1969.
3. Cohen S., "Traffic control strategies during peak hours. The Case of Aix-En-Provence," Second International Conference on Road Traffic Control, IEE, London:1986, pp. 133-136.
4. Lieberman E.B., and Woo. J.L., SIGOP III User's Manual, U.S. Dept. of Transportation, Federal Highway Administration, Report No. FHWA-IP-82-A, Washington, D.C.: 1982.
5. Gartner N.H., Little J.D.C., and Gabbay H., "Steady state traffic optimization by MITROP", pp. 315-327. (source not available)
6. Yagoda H.N., et al., "FORCAST on-line: Making UTCS easier to use," International Conference on Road Traffic Signalling, IEE, London: 1982, pp. 97-102.
7. Hillier J.A., and Rothery R., "The synchronization of traffic signals for minimum delay," Transportation Science, 1967: pp. 81-94.
8. Federal Highway Administration, Traffic Control Systems Handbook, Report No. FHWA-IP-85-12, Washington D.C.: 1985.
9. Schlabbach K., "SIGMA- A new German way for optimizing signals," First International Conference on Applications of Advanced Technologies in Transportation Engineering, Feb 1989: West Lafayette, Indiana, pp. 355-360.

10. Road Research Laboratory, Research on Road Traffic, London, UK.: 1965.
11. Holroyd J., and Hillier J.A., " Area traffic control in Glasgow: A summary of results from four control schemes", Traffic Engineering and Control, 11(5), 1969: pp. 220-223.
12. Ambrosino G., Bielli M., Boero M., Fleischmann S., Hock R., and Irgens M., "A blackboard model for traffic control operations", in Advanced Telematics in Road Transport vol 1., part of DRIVE Project V1055, Elsevier, Amsterdam: 1991, pp. 615-634.
13. Scemama G., "Creation of a knowledge base system for urban traffic control:Methodology", Recherche Transporte Securite (English issue) No. 5, Jun 1990, pp. 11-20.
14. Valdes A., and Rica S. de la, "Area traffic control by computer in Madrid", Traffic Engineering and Control, No. 12, 1970: pp. 132-134.
15. Holroyd J., and Hillier J.A., The Glasgow Experiment: PLIDENT and After, Transport and Road Research Laboratory ,Report No. LR 354, Crowthorne, UK: 1971.
16. Ross D.W., et al., Improved Control Logic for Use with Computer Controlled Traffic, NCHRP Report 3-18 by Stanford Research Institute, Transportation Research Board, Washington, D.C.: 1973.
17. Rach L., "The Development and evaluation of Metropolitan Toronto's real time program for computerized traffic control devices," Control in Transportation Systems: Proc. of the IFAC/IFIP/IFORS Third International Symposium, 1976: pp. 349-362.
18. Nakahara T., Yumoto N., and Tanaka A., "Multi-criterion area traffic control system with feedback features", Paper presented at the First International IFAC/IFIP Symposium on Traffic Control, 1970.
19. Hunt P.B., et al., SCOOT - A Traffic Responsive Method of Coordinating Signals, Transport and Road Research Laboratory ,Report No. LR 1014, Crowthorne, UK: 1971
20. Chien E.S., and Shen D.W.C., "Optimal control of urban traffic signals by means of band transformations," Control in Transportation Systems: Proc. of the IFAC/IFIP/IFORS Third International Symposium, 1976. pp. 153-161.
21. Rakha H. and Aerde Van M., "Real-time incremental coordination of traffic signals using TRANSYT-7F," Traffic Management - Proc. of the Engineering Foundation Conference, 1991: pp. 167-191.

22. Gabard J.F., Henry J.J., Abours S., and Lesort J.B., "CALIFE: On-line calculations of fixed time plans," IFAC Control in Transportation Systems, 1986: pp. 233-237.
23. Nakatsuji T., and Kaku T., "Development of a self-organizing traffic control system using neural network models," Transportation Research Board 1324, 1991: pp. 137-145.
24. Henry J.J., Farges J.L., and Tuffal J., "The PRODYN real time algorithm," IFAC Control in Transportation Systems, 1986: pp. 305-310.
25. Mauro V., and Taranto C. Di, "UTOPIA," IFAC Control, Computers, Communications in Transportation, 1989: pp. 245-252.
26. Drouin M., Abou-Kandil H., Dib G., and Bertrand P., "A new approach for real-time control of urban traffic networks," IFAC Control in Transportation Systems, 1983: pp. 237-241.
27. Drenick R.F., Ahmed S.A., and McShane W.R., "Decentralized control of congested street networks," paper presented at 1981 annual meeting of the Transportation Research Board (TRB), Washington, D.C.
28. Lin D., Ulrich F., Kinney L.L., and Kumar K.S.P., "Hierarchical techniques in Traffic Control", Control in Transportation Systems: Proc. of the IFAC/IFIP/IFORS Third International Symposium, 1976: pp. 163-171.
29. Lee C.S.G., and Saridis G.N., "Hierarchically intelligent control and management of traffic systems," IFAC Control Science and Technology, IFAC 8th Triennial World Congress, Japan:1981, pp. 2395-2400.
30. Greenberg P., Trabelsi A., and Tabak D., "Distributed microcomputer based control of multiple signalized traffic intersections," IFAC Control in Transportation Systems, 1983, pp. 135-140.
31. Lowrie, P.R., "The Sydney Coordinated Adaptive Traffic System (SCATS) - Principles, methodology, algorithms," International Conference on Road Traffic Signaling, IEE, London, pp. 67-70.
32. Heydecker, B.G., "A continuous time formulation for traffic-responsive signal control," Transportation and Traffic Theory: Proc. of the Eleventh International Symposium on Transportation and Traffic Theory, M.Koshi ed., 1990: pp. 599-618.
33. Bell, M.G.H., "A probabilistic approach to the optimization of traffic signal settings in discrete time," Transportation and Traffic Theory: Proc. of the Eleventh International Symposium on Transportation and Traffic Theory, M. Koshi Ed., 1990: pp. 619-632.

34. Gartner N.H., "OPAC: Strategy for demand-responsive decentralized traffic signal control," Control, Computers, Communications in Transportation, IFAC/IFIP/IFORS Symposium, Sep 1989: pp.241-244
35. Barriere J.F., Farges J.L. and Henry J.J., "Decentralized vs hierarchy in optimal traffic control," IFAC Control in Transportation, 1986: pp. 209-214.
36. List G., and Pond J., "A strategy for real-time control of demand-responsive signal networks", First International Conference on Applications of Advanced Technologies in Transportation Engineering, Feb 1989: West Lafayette, Indiana, Feb 1989: pp. 367-372.
37. Rodriguez-Mosocoso J.J., and Fisher D.H., "A connectionist model to intelligent real-time traffic control", Appeared in 2nd International Workshop in Neural Networks and Their Applications, Nov 13-16 1989.
38. Iba W., "Improving the cost-performance tradeoff in traffic control using autonomous intelligent agents", in Workshop Notes (distribution limited to workshop attendees) at AAAI-93 AI in Intelligent Vehicle Highway Systems Workshop, American Association for Artificial Intelligence, Washington, D.C.: July 11-15 1993, pp. 66-72.
39. Saraf, Rajeev K., "Adaptive traffic control using neural networks," Ph.D. thesis, Vanderbilt University, 1994.
40. TRANSYT-7F User's Manual (Release 6), Transportation Research Center, University of Florida, Gainesville. Prepared for Federal Highway Administration, 1988: pp 2.21-2.22.
41. Rumelhart, D.E., Hinton G.E., and Williams R.J., "Learning internal representations by error propagation," in Parallel Distributed Processing, vol. 1., 1986, pp. 318-62.
42. Mirchandini G., and Cao W., "On hidden units in neural nets," IEEE Transactions on Circuits and Systems 36 (5), pp. 661-664.
43. Weiss M.S., and Kulikowski C.A., Computer systems that learn, Kaufman Publishers, San Mateo, CA: 1991.