# Decision Support System for Controlling Traffic Signals

AYELET GAL-TZUR, DAVID MAHALEL, AND JOSEPH N. PRASHKER

In many metropolitan areas traffic control is monitored from control centers. The operator of a control center is asked to make a quick decision and to modify the signal programs of the urban signal network in whole or in part. In order to make a proper decision, the operator must consider a wide range of alternatives and evaluate their expected effects on the whole traffic system. The complicated structure of the problem and the routine occurrence of random events demonstrate the complexity of the decision process in traffic control. A procedure will be described that is aimed at supporting this decision making. The procedure is characterized by the systematic scanning of a wide range of alternatives and includes a special algorithm for reducing the size of the problem and concentrating on the most promising strategies. A statistical decision tree is used for spanning all alternatives and expressing the subjective priorities among them and the projection regarding their consequences. An important option given to the controller is the ability to acquire more information to support his decision by using on-line simulations. This option is time consuming and therefore has a cost. The operator is given the tools to decide whether the additional information is worth the price. In addition, the system contains a systematic procedure to "learn" from past experience and to improve its ability to make decisions under uncertainty conditions.

The growth of congestion in urban networks and the consequent constraints imposed on mobility have made it vital to manage and utilize the existing infrastructure more efficiently. One of the most prominent procedures available for managing traffic control is that of monitoring traffic-signal programs. Research has attempted to find the optimal signal-timing program for a group of intersections during peak hours (*1–4*). Attempts (*5*) were also directed at finding a global optimum for a group of coordinated traffic signals. These programs were prepared off-line so that an operator in a control center could choose the most appropriate program off the shelf, as it were. More recently, efforts have aimed at developing responsive methods (*6*), which are primarily designed to respond to fluctuations in traffic volumes without external intervention. The main doubt concerning the responsive methods is over their ability to converge to a "good" system optimum; the question is whether they merely provide a local solution at the cost of finding a strategy that might better improve the whole system.

The complexity of the control problem stems from the following properties of the system:

1. **Objectives**—The problem has several objectives that should be met simultaneously, for example, minimization of delay, queues, number of stops, energy consumption, and environmental impacts; some of these objectives contradict one another.

Civil Engineering Department, Technion-Israel Institute of Technology, Haifa, Israel.

2. **Dependency**—In an urban network the output of one intersection is the input of another, and the queue at one intersection can block another intersection. As a result, an intersection cannot be treated individually and must be coordinated with its environment. Consequently, the number variables that have to be calculated simultaneously (e.g., cycle time, green splits, and offsets) becomes very large.
3. **Parameter values**—The large number of parameters describing the network (e.g., saturation flows, acceleration times, platoon dispersion, arrival distribution) and the uncertainty regarding their values make it necessary to consider a range of values.
4. **Mathematical model**—The relationships among the various parameters, variables, and objectives are of a complicated nature; attempts to formulate them into one mathematical model end in inadequate results.

This complicated structure of the problem and the routine occurrence of random events demonstrate the complexity of the decision process of traffic control. Logically, however, one may believe that in the future this process will still involve some degree of human judgment and that the operator in a control center will still play an important role.

Described in this paper is a procedure aimed at supporting the decision making of an operator in a traffic-control center. The procedure is characterized by the systematic scanning of a wide range of alternatives. It includes a special algorithm for reducing the size of the problem and concentrating on the most promising strategies. An important option given to the controller is the ability to acquire more information to support his decision. This option has a cost, though, and the operator is given the tools to decide whether the additional information is worth the price. In addition, the system contains a systematic procedure to "learn" from past experience and to improve its ability to make decisions under uncertainty conditions.

In the next section, a description of the structure, the nature, and the dimensionality of the control problem will be given. Next, a tool will be presented for examining the various strategies and options available to the operator of the control center. Then a procedure to reduce the size of the decision process is described, and, finally, the machine-learning ability of the procedure, that is, an automatic process for collecting and then transferring data into useful knowledge, will be discussed.

## CHARACTERISTICS OF TRAFFIC-CONTROL PROBLEM

A traffic-control program is a set of parameters (e.g., cycle length, green split, offsets) that control the right-of-way and that assign priorities among the links of the network. These programs determine the level of service of each link and as a result can affect the routes that drivers choose. During congestion, the amount of green light assigned to a link can actually determine the traffic volume on that link. This ability can be used for controlling the number of vehicles allowed to enter a congested zone.

The traditional method of coping with the massive size of the problem is to divide it into several stages and subproblems:

1. **Division into zones:** The network is divided into several subnetworks, and each can be considered separately. Some degree of dependency between zones is allowed.
2. **Type of signal strategy:** On the basis of various arguments, a general strategy is selected for each zone, for instance, green wave (*7*), critical intersection control (*3*), network design (like TRANSYT). Each strategy can have several variants, such as green wave with one band or a multiband design (*8*) or a network design with various weights assigned to different links.
3. **Design parameters:** At this point, the various parameters are computed for the signals.

Usually, the decisions of the controller do not explicitly take into account the uncertainty regarding changes in demand, changes in routes, incidents, and other random events that might affect the value of many parameters (e.g., saturation flows, start-up delay, acceleration, and speeds). All these random events reject the assumption of stationary conditions and promote the need to combine probabilities and stochastic considerations into the decision process.

The wide range of parameter values together with several control strategies and tactic decisions, and the possible evaluation of each combination through simulation, increase dramatically the dimension of decision space and the number of alternatives that should be considered. The resultant huge dimension complicates the decision process and necessitates basing it more on a systematic process and less on an intuitive one. The tool proposed for handling this problem is the statistical decision tree.

## STATISTICAL DECISION TREE AS A DECISION-MAKING TOOL

A decision tree is based on Bayes' decision theory, which formulates decision-making processes under uncertainty. The tool is suitable for situations in which one course of action must be selected from several possible acts; their respective outcomes are known with a certain degree of confidence, but not absolutely; and there are ways to increase the level of confidence by gaining information. The question that this theory wishes to answer is, Which course of action should be taken in order to maximize the expected benefit (or to minimize the expected loss)? Is it worthwhile to "pay" for extra information?

In order to use the Bayesian approach, the following data should be known in advance:

1. Possible courses of action, i.e., the signal programs;
2. Possible outcomes of each signal program and the probability of occurrence of each outcome;
3. The utility gained, given the occurrence of a specific outcome;
4. Possible experiments (simulation) that can be conducted in order to gain information about the probability of occurrence of each outcome and the cost associated with each simulation;
5. Possible results of each simulation; and
6. The probability of occurrence of each outcome, given a specific result of a simulation.

Traffic engineers face several objectives that they wish to achieve simultaneously: minimum delays, maximum throughput, minimum queue lengths, prevention of spillbacks, minimum number of stops, minimum fuel consumption, and so on. A possible outcome can be expressed as a function of these variables, for example, a success can be defined as a condition in which all values are in some critical region.

Often, the decision maker is faced with a situation in which it is believed desirable to obtain more knowledge in regard to the likelihood of a possible outcome after implementation of a certain signal program. Gaining such information is possible by running a short-term simulation. The results of such a simulation can then be used to update prior probabilities and to obtain posterior probabilities. The time needed to run the simulation is considered a cost.

All the above components—that is, the possible programs and their results and utilities, the possible simulations and their results and cost, prior probabilities reflecting the level of confidence before the simulation, and posterior probabilities reflecting the level of confidence after the simulation—make up the decision tree. The expected utility of each branch is calculated by a backward search, and the branch with the maximum expected utility is chosen. Thus, the statistical decision tree answers not only the question of the course of action that seems the most beneficial, but also that of the possible simulations that are worthwhile to conduct.

Figure 1 shows how the components of the decision-making process integrate into a complete decision support system (DSS). The DSS should have access to updated data regarding traffic conditions in the network. Expected volumes during the planning horizon can be estimated by an external algorithm that can exchange information with the DSS, or they can be extracted from existing data bases. The operator provides the system with the planning strategies most appropriate for the conditions in the network. On the basis of this information, alternative timing programs can be calculated in one of two ways: (*a*) by attaching external software packages (such as TRANSYT or PASSER) to the DSS through the appropriate interface (such connections are feasible in most software packages) or (*b*) by writing design procedures for signals as part of the DSS. The next two functions of the system, the tree-building stage and the tree-search stage, are the two modules at the heart of the DSS. They should be designed and programmed especially for these purposes. When these two tasks are completed and the simulations to be executed are chosen, they can be executed by using a software (e.g., NETSIM) that can interact with the computer language in which the DSS is programmed. Finally, through the user interface, the DSS should instruct the operator which program to implement.

Figure 2 shows the structure of a decision tree for a case in which two programs are considered (X and Y). The first

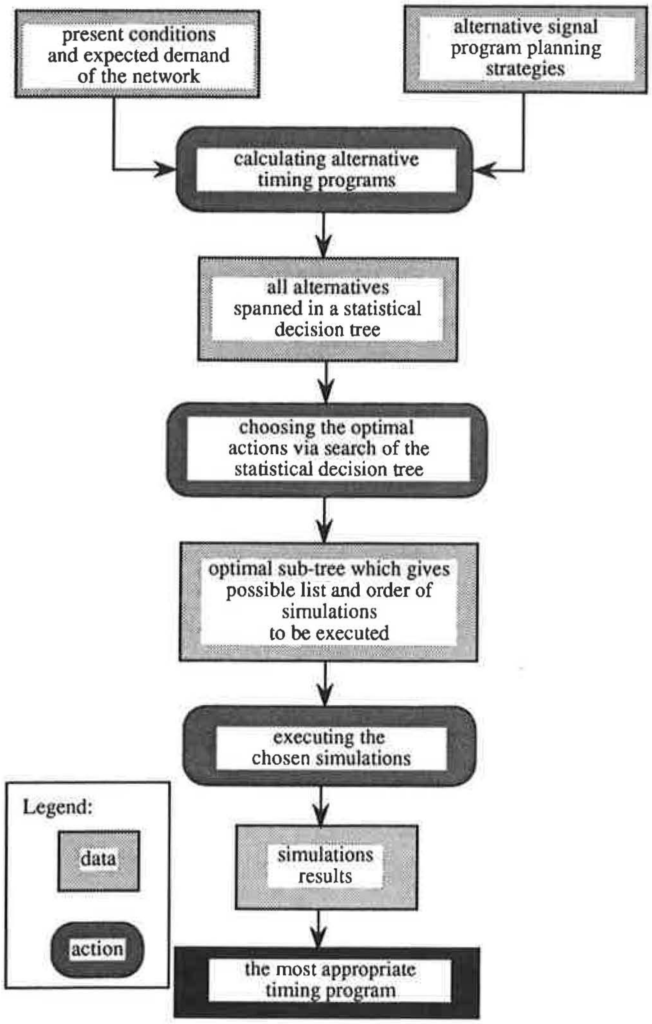


**FIGURE 1 Main components of DSS.**

decision node has four courses of action (alternatives): implement X, implement Y, simulate X, simulate Y. Each course of action has two possible outcomes (success or failure) with its probabilities. Some outcomes result in a termination node and some in new decision nodes, that is, to implement a signal program or to run another simulation.

The size of the tree is a function of the number of possible alternatives, that is, the number of strategies ($R$), the number of demand levels (traffic volumes $M$), and the number of programs taken from the shelf ($B$). Since all parameters are fully defined for a shelf program and, for strategies, should be estimated together with volumes, the total number of possible signal programs ($N$) may be denoted

$$N = R \cdot M + B \tag{1}$$

Testing the performance of each timing program under each possible pattern of traffic volume creates $M \cdot N$ possible simulations. Each branch in the decision tree represents the execution of a certain number of simulations (varying from 0 to $M \cdot N$), in a certain order. The number of branches having $i$ simulations is computed as follows:

$$i! \binom{MN}{i} = \frac{(MN)!}{(MN) - i)!} \tag{2}$$

Each simulation, similarly to each timing program, has two possible outcomes (failure or success), so the number of leaves of a branch containing $i$ simulations is

$$2N \cdot 2^i \tag{3}$$

On the basis of Equations 1–3, the total number of termination nodes on a tree becomes

$$2N \cdot (MN)! \sum_{i=0}^{MN} \frac{2^i}{(MN - i)!} \tag{4}$$

To demonstrate, assume a tree with three strategies, two levels of volumes, and one shelf program. The number of possible final outcomes would be $3 \cdot 10^{16}$.

## TRUNCATION PROCESS

The dimension of the tree and the time it takes to develop all its branches necessitate that the decision maker decide what parts of the tree to span and what to neglect. Truncating a branch implies that some possible actions will not be considered in detail in later steps. To accomplish a justified truncation, the operator should be provided with a quantitative figure of what might be lost if a certain branch of the tree were ignored. The approach to the problem here was based on developing upper bounds for the expected utilities of branches whose examination involves a cost. With this upper bound, the decision maker can then answer questions such as, Is it worth running a simulation for program $j$?

The calculation of upper bounds exploits the nature of the control problem and the dependency relationships among the signal programs. The main assumptions are discussed in the following sections.

### Assumption 1

The success of a certain program increases the posterior probabilities for the success of all other programs; that is,

$$p(i|\{G,S_k\},\{F\}) - p(i|\{G\},\{F\}) \geq 0 \tag{5}$$

where

$G$ = group of simulations that were executed and succeeded,
$F$ = group of simulations that were executed and failed,
$p(i|\{G\},\{F\})$ = conditional probability of program $i$ to obtain successful results, given groups $G$ and $F$,
$p(i|\{G,S_k\},\{F\})$ = conditional probability of program $i$ to obtain successful results, given that group $G$ and simulation $k$ succeeded and group $F$ failed.

This assumption is motivated by the knowledge that the network is not oversaturated and that a signal program can improve conditions.

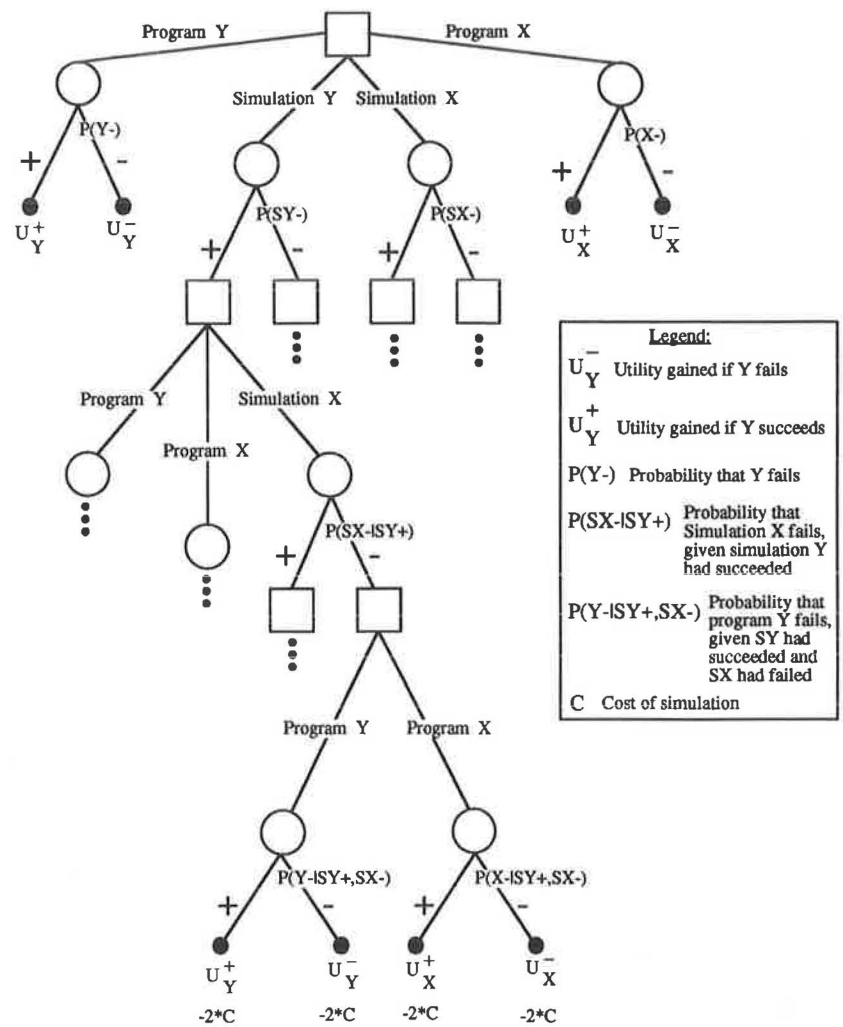


**FIGURE 2 Decision tree with two signal programs.**

### Assumption 2

An additional increase in probabilities following the success of a certain simulation is greatest for the program that was examined, that is,

$$p(i|\{G,S_i\},\{F\}) - p(i|\{G\},\{F\}) \geq p(k|\{G,S_i\},\{F\}) - p(k|\{G\},\{F\}) \quad (6)$$

where $i$ and $k$ are different signal programs.

This assumption simply expresses the obvious fact that the largest contribution of running a simulation accrues to the program examined by the simulation.

### Assumption 3

An additional increase in probability following the success of a certain simulation decreases as the number of simulations preceding it increases.

$$p(i|\{G,S_k\},\{F\}) - p(i|\{G\},\{F\}) \geq p(i|\{G,S_m,S_k\},\{F\}) - p(i|\{G,S_m\},\{F\}) \quad (7)$$

This is simply a manifestation of the law of diminishing returns.

After some laborious operations (9), it can be shown that the maximum loss of expected utility following the relaxation of a certain simulation is calculated according to

$$p(S_i|\{\emptyset\},\{\emptyset\}) \cdot [EU(i|\{S_i\},\{\emptyset\}) - EU(i|\{\emptyset\},\{\emptyset\})] - C \quad (8)$$

where

$p(S_i|\{G\},\{F\})$ = the probability of success of simulation $i$ given groups $G$ and $F$,
$EU(i|\{G\},\{F\})$ = the expected utility of program $i$ given groups $G$ and $F$,
$C$ = the cost of simulation, and
$\emptyset$ = the null set.

This means that the maximum loss of not running simulation is the difference between the expected prior utility and the expected posterior utility of success, provided that the simulation was the first simulation to be run and that it succeeded, multiplied by the probability of success minus the cost of simulation $i$.

The practical question answered by the mathematical expression above is, Which simulation should be relaxed in the truncation process in order to minimize the expected maximum loss of expected utility? Two attributes characterize the simulation to be relaxed:

1. Its contribution to the information about the probability of success of the program tested by it is low.
2. The probability of obtaining successful results when executing it is low.

By multiplying the values of both attributes by each other, the integrated criterion is achieved.

## NUMERICAL EXAMPLE

This example demonstrates a decision problem in which the controller has to choose between one of three possible strategies $\{A_1, A_2, A_3\}$. In the case of a successful implementation, the utilities of the three strategies are 370, 340, and 310, respectively. In the case of a failure, they all have a cost of 50 (utility of $-50$). Figure 3 demonstrates the six possible decisions of the first decision node: to select one of the three strategies or to run a simulation of each one of them. At this stage, the expected utility of the three strategies can be computed and it can be seen that Strategy 1 has the largest expected utility. Instead of selecting a strategy, the controller can run a simulation and can make a decision after obtaining the results. Figure 4 shows part of the tree following the success of simulating $A_1$. The best process is denoted by a bold line. According to these rules, the best path is as follows: simulate $A_1$; if it succeeds, simulate $A_2$; if it succeeds, select $A_2$; if simulation of $A_2$ fails, select $A_1$; if simulation of $A_1$ fails, select $A_3$.

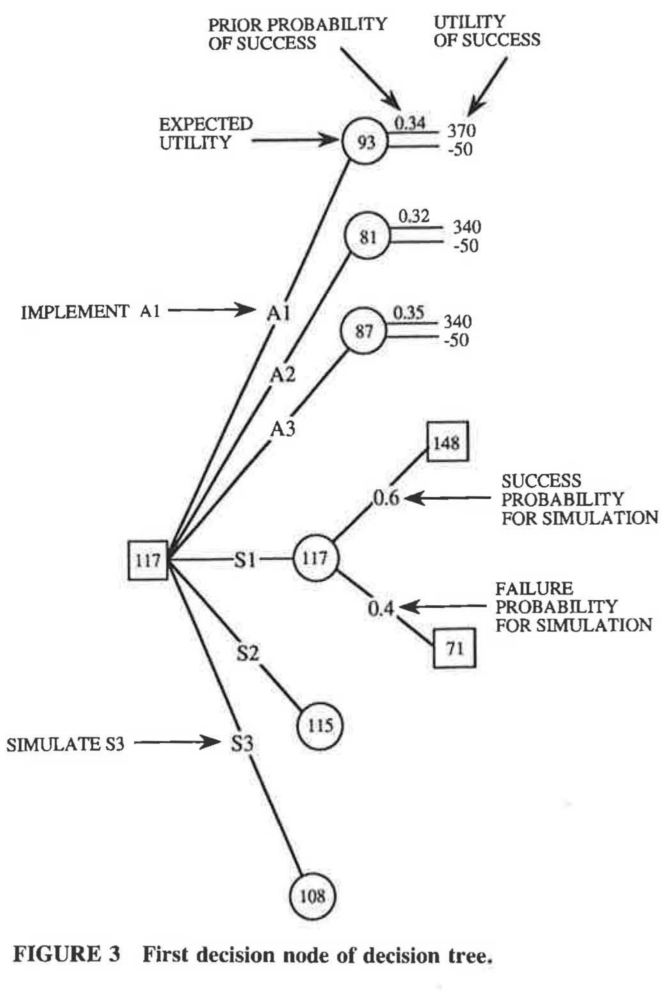


**FIGURE 3 First decision node of decision tree.**

The value of the simulation is illustrated through several facts:

1. **The change in probabilities:** The success of a simulation increases the posterior probability; for example, the prior probability of success for $A_1$ is 0.34 and the posterior probability if the simulation succeeds is 0.48. The difference is a result of the extra information.
2. **The change in expected utility:** In this example, without simulation the maximum expected utility is 95, but after running a simulation, this value increases to 117, that is, an additional utility of 24.

If one considers the relaxation of simulation $A_1$, the upper bound for the utility of this part of the tree should be computed. According to Equation 8 and Figure 4, the calculation is as follows:

$$p(S_i|\{\emptyset\},\{\emptyset\}) = 0.6$$

$$EU(i|\{S_i\},\{\emptyset\}) = 152$$

$$EU(i|\{\emptyset\},\{\emptyset\}) = 93$$

$$C = 10$$

Thus,

$$p(S_i|\{\emptyset\},\{\emptyset\}) \cdot [EU(i|\{S_i\},\{\emptyset\}) - EU(i|\{\emptyset\},\{\emptyset\})] - C = 25$$

The upper bounds of relaxing all other simulations can be computed similarly, and the one with the lowest upper bound is chosen not to be spanned. The upper bound of Simulation 2 is 20 and the upper bound of Simulation 3 is $-7$. It can be observed that Simulation 3 has the lowest upper bound. Moreover, the negative sign indicates that no loss in expected utility would be obtained as a result of relaxing Simulation 3. This phenomenon is due to the low contribution of Simulation 3 to the information in hand compared with its cost.

## DISCUSSION OF RESULTS

Using the decision tree as part of a DSS actually divides the decision-making process into two stages. In the first stage, the controller answers the question of what simulations to run or where one needs to improve one's knowledge at a certain cost. In the second stage, the controller searches for the branch that maximizes expected utility. Afterwards, the necessary simulations are performed and a decision is reached. In this way, the controller can adopt the process that is most suitable for existing conditions.

Between these two stages, the truncation option allows the DSS to limit the time dedicated to the search stage by trimming some branches of the tree. This truncation process might result in a loss of expected utility, but the maximum value of this loss is known in advance to the operator. Appropriate criteria are used to decide whether the truncation is worthwhile.

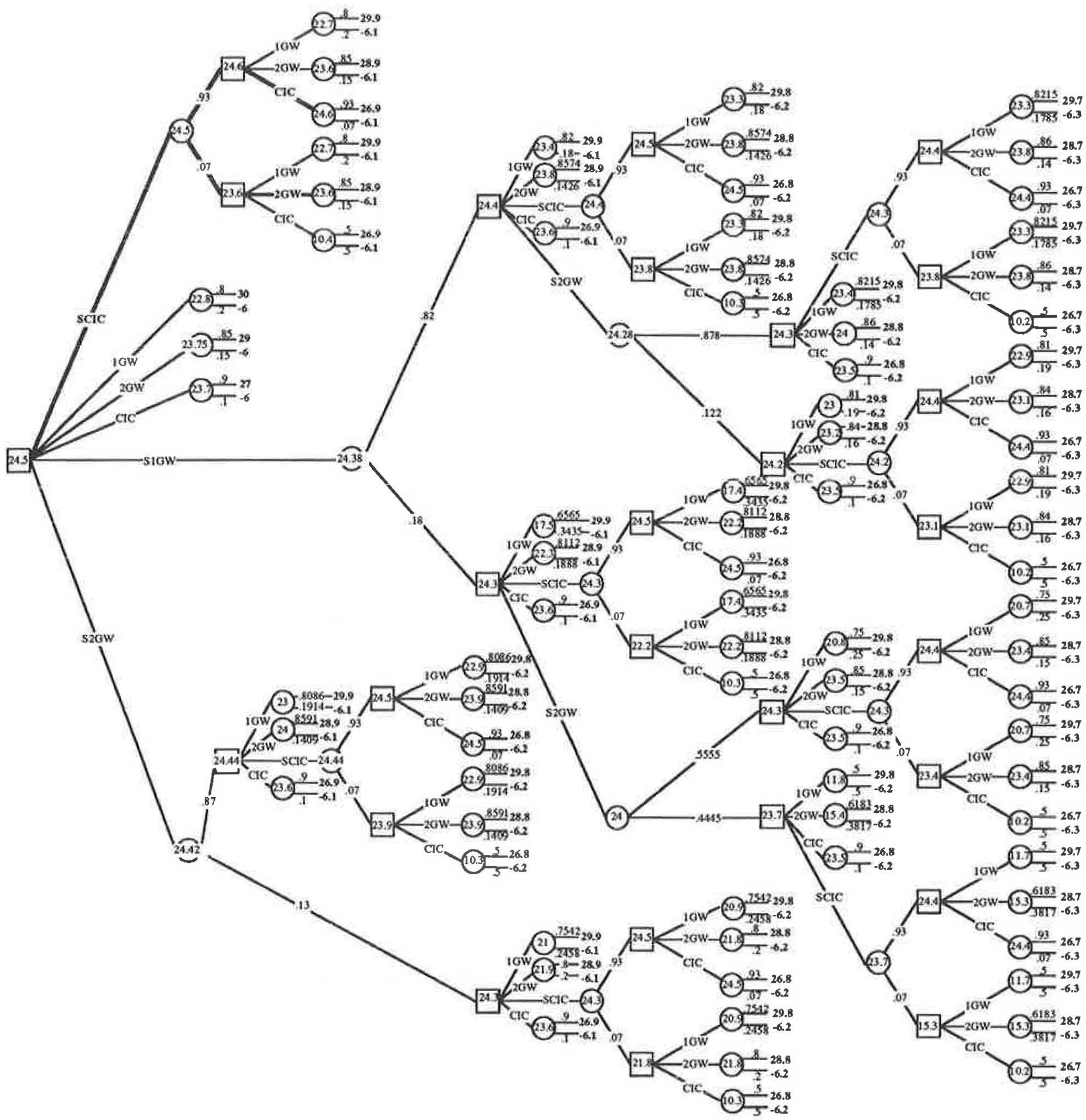


**FIGURE 4 Part of a decision tree for case of successful simulation of $A_1$.**

The DSS described in this paper is still under development and has not yet been fully integrated with all necessary components. The DSS should be designed to run on a work station, and the complexity of the tasks it is meant to perform should fall within the range that this type of equipment can handle.

The quality of the decision-making process described in this paper depends heavily on the amount and quality of information and knowledge available to the controller. This need can be satisfied gradually over time, especially if experience is transferred through machine learning into practical expressions like prior probabilities, success probabilities following simulations, and so forth.

Machine learning is an automatic process of collecting data and transferring it into useful knowledge. This process is done efficiently if, from the limited knowledge that is collected every day, a large number of parameters can be updated. This broad inference should be based on a deep understanding of all dependencies and relationships existing in the system.

## ACKNOWLEDGMENT

This work was supported by the Office of the Traffic Commissioner, Israeli Ministry of Transportation.

## REFERENCES

1. Singh, M. G., and H. Tamura. Modelling and Hierarchical Optimization for Oversaturated Urban Road Traffic Network. *International Journal of Control*, Vol. 20, No. 6, 1974, pp. 913–934.
2. Michalopoulos, P. G., and G. Stephanopoulos. Optimal Control of Oversaturated Intersections: Theoretical and Practical Considerations. *Traffic Engineering and Control*, Vol. 19, No. 5, 1978, pp. 216–222.
3. Michalopoulos, P.G., and G. Stephanopoulos. An Algorithm for Real-Time Control of Critical Intersections. *Traffic Engineering and Control*, Vol. 20, No. 1, 1979, pp. 9–15.
4. Lieberman, E. B. Concepts of Control for Oversaturated Networks. Presented at meeting of TRB Committee on Traffic Signal Systems, Minneapolis, Minn., 1990.
5. Robertson, D. I. *TRANSYT : A Traffic Network Study Tool.* TRRL LR 253. U.K. Transport and Road Research Laboratory, Crowthorne, Berkshire, England, 1969.
6. Robertson, D. I., and P. B. Hunt. A Method of Estimating the Benefit of Co-ordinating Signals by TRANSYT and SCOOT. *Traffic Engineering and Control*, Vol. 23, No. 11, 1982, pp. 527–531.
7. Little, J. D. C., M. D. Kelson, and N. H. Gartner. MAXBAND: A Program for Setting Signals on Arteries and Triangular Networks. In *Transportation Research Record 795*, TRB, National Research Council, Washington, D.C., 1981, pp. 40–46.
8. Gartner, N. H., S. F. Assmann, F. Lasaga, and D. L. Hou. MULTIBAND—A Variable-Bandwidth Arterial Progression Scheme. In *Transportation Research Record 1287*, TRB, National Research Council, Washington, D.C., 1990, pp. 212–222.
9. Gal-Tzur, A., and D. Mahalel. Evaluating the Option to Truncate Decision Trees. *TRI Research Report* (in Hebrew). Transportation Research Institute, Technion, Haifa, Israel, 1992.

*Publication of this paper sponsored by Committee on Traffic Signal Systems.*