Knowledge based approach to urban traffic control

N. Findler

Department of Computer Science and Engineering, and Artificial Intelligence Laboratory, Arizona State University, Tempe, AZ 85287-5406, USA

ABSTRACT

We present the general design of a system to control urban street traffic signals. It is based on cooperating, learning, real-time, distributed expert systems. We also describe the operation of a running prototype program which, while using several simplifying assumptions, has proven the technical feasibility of the approach. It has also attained a 36% improvement in the traffic flow under non-saturated conditions. Current developments include the design of a general-purpose system that can be customized to most street configurations. Finally, we draw conclusions concerning the distributed planning and problem solving methodology.

INTRODUCTION

Traffic engineers have been using different tools of mathematics, statistics and computer science to devise systems that can improve the traffic conditions of our congested cities [4-6]. Some techniques of Artificial Intelligence have also been employed to generate, for example, better *constant* control of street traffic lights and real-time expert systems controlling street traffic lights *centrally* in a dynamic fashion [7]. Distributed and dynamic control, however, has not been used although it can offer several advantages as follows [1-3]:

- Spatially and chronologically *local conditions* are usually more relevant to the decisions to be made. Traffic accidents, the ending of a major sport event, road repair or an "unscheduled" holiday are examples of local changes that cannot be considered by a centrally controlled regime.

- The rate of change in local conditions is usually very high. Even if high-performance sensors are available for data input, communication and computational bottlenecks would not allow the existence of a timely and responsive control environment.
- A centrally organized, real-time planning technique of satisfactory quality is not feasible because of the overwhelming amount of data to be processed and the large number of decisions to be made and communicated to the traffic lights.
- Changes to a distributed control system are easy and inexpensive to make, when the "permanent" traffic environment changes.

THE APPROACH

The following working conditions and assumptions have been established for our long-term efforts:

- There is *one processor at each intersection,* which communicates *directly* with the four processors at the adjacent intersections.
- The *communicated information* is three-fold:
  - *raw data* (essentially, the number and the speed of cars going in each of the four directions at an intersection),
  - *processed information* (the type and the rate of change of certain traffic flow features),
  - *expert advice* (e.g., "lengthen the period of green light in the East-West direction").

  It should be noted that the latter two categories of information can propagate over an indefinite number of intersections but with gradually changing contents. Such "combined" information coming from many intersections along a given direction is the weighted average of the contributing information — the farther away the source, the less important its contribution is.
- The operation of the whole system is based on
  - a set of cooperating real-time expert systems which work in conjunction with a simulation-based planning system,

- a limited amount of noisefree communication which triggers both gradual and sudden changes in the traffic light control regime.

- There are several possible *criteria of operation* or *measures of effectiveness* which represent the *objective function* to be minimized. These can be the average travel time, the maximum waiting time at intersections, the average number of stops during travel, etc.
- The system would attain local and global optima within moving "time windows" through learning programs. The learning programs work in two phases and along different dimensions, as discussed later.

Such system for traffic control is to produce several benefits, such as faster traffic flow, more efficient usage of available roads, fewer accidents, lower driver frustration, reduced air pollution and fuel consumption.

ON THE CONTROL STRATEGIES

There are different strategies possible for the control — each with its own set of rules to follow. In the explanation, we will employ the average travelling velocity of cars going though an intersection as the objective function to be minimized. (The terms *intersection* and *traffic light* are used interchangeably.) The control variables for the traffic light are:

- the length of the cycle;
- the length of "active time" (sum of the periods of amber and green lights in one or the other direction); a related entity is the "cycle split" — the ratio of the green periods in the two directions;
- the point of time when the cycle starts.

We parenthetically note that traffic experts do not vary the time period for the amber phase, t_a. Its duration (3 to 5 seconds) depends on several factors, such as the usual car speed and average deceleration rate in that direction, the width of the intersection, and so on. A good rule of thumb is to keep it at

$$t_a[\text{sec}] = 0.1 * v_{max}[\text{m/h}]$$

where v_{max} is the speed limit.

There are three basic strategies, depending on the prevailing local traffic pattern. The third strategy can be further divided into three substrategies, with reference to the type of control variable referenced. We present the strategies in the order of usage priority.

Strategy I: *The Semiactuated Regime*

This strategy is to be used when the traffic flow in one of the intersecting streets is extremely small. The light stays green in the busier direction until one or more cars approach the intersection in the less busy street. Then the light turns green in the latter direction as needed, up to a predetermined maximum time.

Strategy II: *The Platooning Regime*

In moderate traffic, it is a good idea to encourage cars to travel in "platoons" — in small groups separated by gaps. Ideally, the light should be green when the cars are coming to cross the intersection, and red when the gaps appear. (Note that the staggering of the traffic lights to attain this effect can be easily controlled also by a static and global control system.) Since it is more effective to stagger the traffic lights to handle platoons than to follow one of the lower priority schemes to be described below, the platooning scheme should be followed when possible.

Strategy III: *The Regime To Control Individual Characteristics Separately*

This mode of operation can be based on three sets of rules. Each set controls a different variable. We rank them again in the order of usage priority.

Substrategy IIIa: *Modify Cycle Length*

As a general heuristic, it has been found that when the traffic flow is heavy (say, above 1300 cars/lane/hour), longer cycle lengths speed traffic. In turn, when traffic is lighter, shorter cycle lengths are advisable. However, cycles of longer than 180 seconds or shorter than 40 seconds are inefficient and should not be used.

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Substrategy IIIb: *Change Cycle Splits*

The cycle should, in general, be split so that the direction with heavier traffic flow receives the longer green light.

Substrategy IIIc: *Change Cycle Start Time*

If it is found that too high a proportion of cars arrive on the red light or they have to wait longer than seems appropriate with the given the flow and cycle split, the cycle start time should be adjusted to reduce waiting time. Depending on the prevailing traffic pattern, such a measure may be of long-term help or may improve the situation only temporarily.

THE INFORMATION COMMUNICATED BETWEEN CONTROLLERS

The following symbolic and numerical information is the result of some calculation on locally sensed data, which must then be transmitted to the appropriate *adjacent* processor:

- a car crossed the intersection in the direction in question when the light turned green;
- the number of cars having crossed the intersection in the direction in question;
- data on cycle length, cycle start time, and cycle split time;
- a congestion is being experienced at the controlled intersection, which is moving toward the adjacent one;
- a congestion is moving from the adjacent intersection toward the controlled one;
- a severe congestion is being experienced at the controlled intersection, which is moving toward the adjacent one;
- a severe congestion is moving from the adjacent intersection toward the controlled one;

In the above, *flow* is the number of cars clearing the intersection per minute and per lane, *congestion* occurs when a car must wait a whole cycle before clearing the intersection, and *severe congestion* occurs when a car must wait at least two cycles before clearing the intersection.

RULES AND META-RULES TO CONTROL THE TRAFFIC LIGHTS

There is a possible natural segmentation of the rules. Each of the strategies and substrategies listed above corresponds to a particular mode of operation whose controlling rules belong to a distinct *rule segment*. (Different rule segments may have a small but necessary overlap of membership). We have also identified a set of *meta-rules* that, in response to the current traffic pattern and the characteristic period, point to the rule segment to be applied until the environment changes.

ON SCENARIO GENERATION

A large number of experiments needs to be performed in trying to optimize the rule base of the *distributed, cooperative expert systems*. Each series of experiments has to be provided with an overall traffic pattern that applies to a *characteristic period* of the day (e.g., early morning rush hour, mid-day traffic, late afternoon rush hour, evening traffic and night traffic), of the day of the week (e.g., workday, Saturday, Sunday, other holiday) and, possibly, of the season of the year (e.g., vacation time).

In our explanation, we will refer to the *Manhattan grid* (see Figure 1) as the generic basis of city maps. (This was also used in our first simplified prototype program.) The following idea enables us to study only a relatively small segment of the whole network, without losing information and risking unrealistic traffic situations. Let us call a rectangle cut out from an indefinitely large *Manhattan grid* the *area of concern*. If the number of intersections in the E-W direction is *w* (width) and in the N-S direction is *d* (depth), we can name each intersection within the area of concern by a number as shown in Figure 2.

Further, let the area of concern be surrounded by four *peripheries*, each of which contains a sequence of street intersections. Therefore, the names of the intersections along the four peripheries are as follows:

Top, E-W direction:	1	2	...	w
Left, N-S direction:	1	w+1	...	(d-1)w+1
Bottom, E-W direction:	(d-1)w+1	(d-1)w+2	...	d.w
Right, N-S direction:	w	2w	...	d.w

We represent the characteristic traffic pattern of a scenario to be generated, as defined before, by two *w*x*d* matrices. The matrix elements stand for the "source" and "sink" specifications, respectively, for each intersection — i.e., the number of cars originating from a given intersection and coming to it as a destination. The actual values of the elements, produced by pseudo-random number generators, have two constraints:

- The sum of the source numbers equals the sum of the sink numbers, and they both equal a constant representing the characteristic period.
- The source and the sink numbers associated with the intersections on the peripheries equal a user-specified constant times the random numbers obtained. We can thus take care of the fact that a lot of traffic goes to and comes from the area of concern across the peripheries.

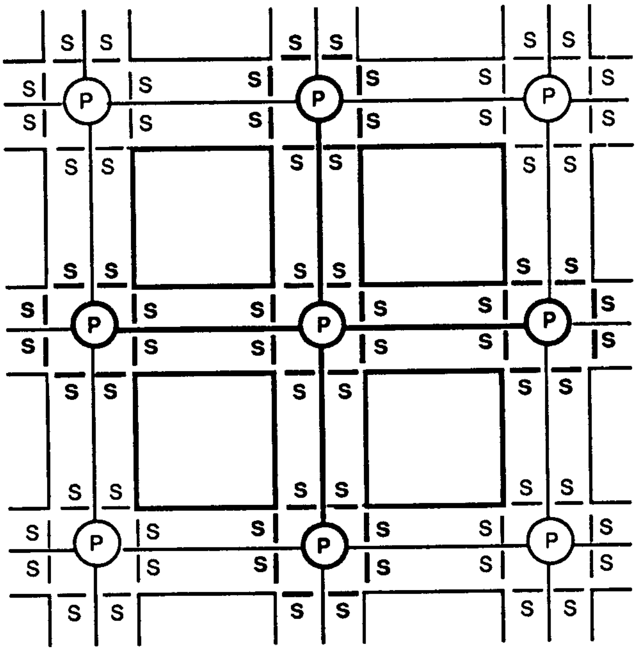

Figure 1 — The "Manhattan grid" street pattern used in the prototype program. There is a processor, *P*, at every intersection receiving input data from its own sensors, *S*, and from the four adjacent processors. There are two lanes in each direction and no left turn is permitted.

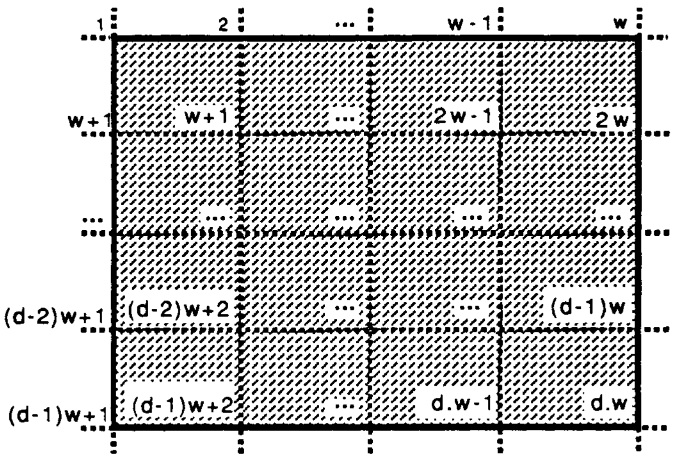

Figure 2 — The *area of concern* cut out of an indefinitely large Manhattan grid; the four *peripheries* bordering it are marked by heavy lines and the intersections are numbered.

THE OPTIMIZATION OF THE RULE BASE

We have designed two distinct *phases of learning*. The first is characterized as *laboratory-based* and the second *field-based* (although some of the computing activity during the two phases takes place both in the laboratory and in the field).

The phase one, laboratory-based, learning prepares the optimum rule bases of the expert system for each defined *scenario*. It means that the system will be operating in an optimum manner, assuming the traffic flow to be steady and equal to the one in the scenario at hand. To arrive at the optimum rule base requires a two-stage development which concerns the selection of *appropriate set of rules* and the best *parametric values* of each rule, respectively. The researcher selects one of the possible *measures of effectiveness* (the objective function to be optimized). (Different ones may be relevant from the viewpoint of traffic flow, air pollution, fuel economy and driver's psychology.) An important criterion is also computational appropriateness; that is, the time and memory requirement of computing the measure of effectiveness being considered.

The optimum parametric values is to be determined by an efficient *hill-climbing method* (such as the steepest ascent approach), usual with numerical optimization problems when the objective function is not available in analytical form.

The following approach is taken for selecting the most appropriate set of rules. First, one identifies a *core set* of rules that are considered indispensable for running the system. The parametric values of these are optimized and kept constant during the later development. Next, one rule at a time is added to the core set from the large total set of possible rules, its parametric values are optimized and the additional benefit due to the new rule is evaluated in quantitative terms. The rule with the best such effect is added to the core set which now becomes the *basic set of rules*. This basic set keeps on growing until the system finds that the addition of the next best rule is no longer cost-effective; that is, the improvement it brings about no longer justifies the cost of computing it.

At this point, we have a rule base that has optimum parametric values and comprises an optimum set of rules with reference to the *average operation* within a scenario. In other words, it is the best control mechanism *statistically speaking*. One of the major advantages of the dynamic, distributed approach is that the control system can respond to spatially and chronologically *local* perturbations; i.e., it can react optimally to a sudden change in the environment (for example, an accident, the ending of a sport event, a snow storm) or to a temporary but longer lasting change (for example, some road repair, strike by public transportation employees). This is the task of *phase two learning*.

The response of the control regime to local perturbations consists of temporary changes in both the set of active rules and their parametric values. The former means that certain rules are invoked that were not active in the "equilibrium operation". Recall the different control strategies for the street signals and the corresponding sets of rules that become active when the appropriate traffic pattern prevails. Similarly, we envisage some special rules that come into action in response to particular sudden changes in the environment. Further, the "statistically optimum" parametric values are subject to temporary modifications to respond to the local perturbations.

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

The system is prepared for it — operating in a *predictive mode* — because each processor gets information from the four adjacent processors about the traffic pattern to come. Another learning mode is about changing rule priorities, when warranted, for the *resolution of conflict* between rules with condition parts matching the current situation.

The above powerful but complex methods must be tried out and fine-tuned first in the laboratory but the routine, field operations are also likely to affect them.

THE IMPLEMENTATION OF A PROTOTYPE SYSTEM

We have implemented a prototype system to prove the feasibility and effectiveness of the approach described above. It consists of five program components: a traffic simulator, a traffic scenario generator, a graphics display module, a rule-base coupled with a rule-driver to control the traffic signal parameters, and a hill-climber optimization module for the rule parameters. There were certain simplifications introduced, as compared to the ideas presented before:

- We have used the Manhattan grid as the street pattern. It means that left turns are prohibited, all streets are two-way, have two lanes in each direction, cross at right angles, and run either in the North-South or East-West direction. (Note that the new system has been designed so that all realistic features, such as one-way streets, changeable lane directions, left-turn lanes, can be specified, and an appropriate computer network can be custom-made for most existing road configurations.)
- Since left turns are not permitted in the Manhattan grid, a significant skew developed initially by the overwhelming number of clock-wise routing patterns. This was then rectified by some ad hoc techniques involving nodes at the peripheries of the area of concern as well as suboptimal routings that compensated for the skew. Also, each intersection had a traffic signal.
- There was only one quality measure for the rule set, the average travelling velocity which is equal to the ratio between the total travel distance and total travel time during a simulation run (with reference to a certain scenario).

- To speed up the simulation runs to an acceptable level, it was necessary to reduce the number of routine calculations and the number of items the system had to keep track of. This meant, for example, a uniform formula for the acceleration and deceleration of cars. (However, the initial and final speeds, of course, depended on the local conditions.)
- Cars would switch lanes when it is safe to do (a simplification...) and either they are in the left lane and wish to make a right turn soon or the occupancy levels of the two lanes are very uneven. Also, start-up delays after the signal turns green were made uniform for all first cars and, to a different degree, for all subsequent cars.
- We have decided to speed up execution and use only compiled (and not interpreted) LISP programs. It meant that new rules could not be generated automatically by the program — missing out on a very high-level learning feature.
- A rather significant (and ill-advised) simplification was to ignore the possibility of several rules satisfying the current conditions and the need to resolve the conflict among them. The system simply fired the first applicable rule and never tried to re-arrange the order of the rules on the basis of their level of expected success or frequency of usage. (We have done some re-ordering manually to respond to certain apparent problems in the results but this can be only an ad hoc remedy in a prototype system.)
- The amount of processed information passed from adjacent processors was very limited and, therefore, did not yield the *predictive value* needed for effective control. Further, no expert advice/request propagated from other processors at all. The net result of this preliminary choice was that, although the the traffic signal at a given intersection did respond to a suboptimum local traffic flow, the overall traffic pattern did not get much help.
- Only a simplified, quick-and-dirty version of the hill-climbing method was implemented that could make simultaneous use of some 30 Apollo workstations connected in a network(!) It is less than certain that a set of *overall*, rather than local, optimum parameter values have been reached. Table 1 contains the tabulated results of the hill-climbing optimization process.

- Limitations in computing resources have enabled us to complete scenario runs that generate only either relatively light or already saturated traffic flows — neither of which can really show the strength and flexibility of the proposed system.

Intersection Number	Before Optimization		After Optimization	
	Average Waiting Time [sec]	No. of Cars Passing through	Average Waiting Time [sec]	No. of Cars Passing through
1	34.37	2.255	26.89	2.251
2	32.12	2.260	19.25	2.250
3	27.02	2.270	17.70	2.250
4	28.18	2.240	16.78	2.202
5	31.84	2.265	20.03	2.254
6	24.10	2.246	18.62	2.230
7	24.54	2.268	14.64	2.241
8	28.92	2.265	17.39	2.246
9	41.92	2.229	15.89	2.196
10	37.48	2.276	20.44	2.248
11	23.71	2.227	19.33	2.198
12	24.59	2.207	20.27	2.187
13	27.55	2.239	16.09	2 .209
14	28.74	2.171	15.06	2.155
15	28.32	2.254	19.01	2.223
16	23.40	2.195	15.99	2.185
17	30.61	2.180	13.90	2.175
18	31.83	2.200	20.12	2.173
19	28.85	2.120	24.12	2.120
20	32.44	2.199	19.24	2.184
21	31.27	2.218	25.93	2.210
22	25.42	2.195	18.52	2.197
23	27.86	2.204	18.21	2.199
24	31.36	2.152	16.72	2.129
25	28.91	2.208	20.35	2.198

Table 1 — Comparison of quality measures in moderate traffic within a 5x5 block area before and after the hill-climbing optimization process. The waiting time before the optimization process, averaged over all intersections being 29.41 seconds, has been reduced to 18.82 seconds — an improvement of 36%.

SUMMARY

We have discussed an economically important domain of computer applications — to improve urban traffic flow by using cooperative, distributed, learning expert systems that control street traffic signals. This domain has very specific reliability concerns, quality measures, computational and communication requirements, timing aspects, geographically distributed input and output operations, inter-node cooperation, and a need for reliable and gracefully degrading performance when some operational and/or computational units become disabled. All these characteristics point to the need for the Distributed Planning and Problem Solving approach, using a network of identical processors.

Some of the knowledge is needed by every node in the network (e.g., the rules of the control operation), some is node-specific (e.g., geometrical information about its close environment). The system works in *real-time* and requires satisfactory solutions *by certain time.* The control task has a medium-level time-criticality.

Our current efforts aim at generalizing the area of applicability of the work on the prototype system, eliminating most of the simplifying assumptions and inefficiencies in it in order to produce a system that can be custom-made for all realistic road configurations. The expected benefits are faster traffic flow, more efficient usage of available roads, lesser cost of building future roads, reduced air pollution, reduced fuel consumption, fewer accidents, and lower driver frustration.

ACKNOWLEDGMENT

I thank John Stapp, Serban Catrava and Sudeep Surender for their conceptual and programming contributions.

REFERENCES

[1] Findler, N. V. *Contributions to a Computer-Based Theory of Strategies*. Chapter 6: Distributed Planning and Problem Solving Systems. Springer-Verlag, New York, Berlin, Heidelberg, 1990.

[2] Findler, N. V. and J. Stapp. A distributed approach to optimized control of street traffic signals. *Journal of Transportation Engineering*, **118**, 99-110, 1992.

[3] Findler, N. V. Dimensions of learning in a real-time knowledge-based control system. *Proc. of IFAC/IFIP/IMACS Internat. Symp.*, 313-317. The Hague, The Netherlands; 1992.

[4] Foraste, B., and G. Scemama. Surveillance and congested traffic control in Paris by expert system. *Institution of Electrical Engineers International Conference on Road Traffic Control*. 91-94, 1986.

[5] McShane, W., and R. Roess.*Traffic Engineering*. Prentice Hall: Englewood Cliffs, NJ, 1990.

[6] Steffi, Y., and W. B. Powell. Optimal signal settings over transportation networks. *Journal of Transportation Engineering*, **109**, 824-839, 1983.

[7] Zozaya-Gorostiza, C., and C. Hendrickson. Expert system for traffic signal setting assistance. *Journal of Transportation Engineering*, **113**, 108-126, 1987.