EXPERT SYSTEM FOR TRAFFIC SIGNAL SETTING ASSISTANCE

By Carlos Zozaya-Gorostiza¹ and Chris Hendrickson,² A. M. ASCE
(Reviewed by the Urban Transportation Division)

ABSTRACT: An experimental knowledge-based expert system to assist in traffic signal setting for isolated intersections is presented. In contrast to existing computer aids, the system can be applied to intersections of highly irregular geometries. Algorithmic processes to evaluate signal settings and decision tables to identify traffic flow conflicts are invoked by the expert system; phase distribution of flows is performed by applying heuristic rules. The system was written in the OPS5 expert system environment. Advantages and disadvantages of the expert system programming approach relative to conventional algorithmic processes in the traffic engineering domain are described.

INTRODUCTION

Traffic signal setting for isolated intersections is a classic and common problem in transportation engineering. The problem has been the subject of intense study (10). Numerous computer aids have been developed to aid engineers in selecting appropriate phase distributions and control strategies; Radwan and Sadegh (8) provide a recent review of microcomputer based aids. Despite the existence of such analysis aids, a number of problems exist that prohibit their useful application and require expert attention. Moreover, the importance of traffic signal setting to the effective performance of roadway systems suggests that improvements to existing aids would be desirable (9).

As Ozanne (7) notes, a common shortcoming of existing computer aids is the inability to deal with uncommon geometries or special design considerations. For example, existing software generally cannot deal with five leg intersections or with dual-system intersections in which intersections are so close together that a single control strategy must be used. Queue length restrictions, unusual turns, transit stops, or pedestrian requirements make standard analysis recommendations inapplicable in many cases. Ozanne cites the example of a peculiar intersection in which queue lengths are important at certain times of the day and not at others (7). It is extremely difficult to handle situations of peculiar geometries or heuristic search and evaluation among numerous alternatives in the context of standard programming techniques. In essence, most analysis programs existing today must preprogram all possible contingencies and specify the sequence of actions undertaken by the program. As a result, the range of possible applications is limited.


Note.—Discussion open until August 1, 1987. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on August 28, 1985. This paper is part of the Journal of Transportation Engineering, Vol. 113, No. 2, March, 1987. ©ASCE, ISSN 0733-947X/87/0002-0108/$01.00. Paper No. 20590.
In this paper, we will consider a prototype traffic signal setting assistant, called TRALI, which is based on radically different programming techniques. The use of a knowledge-based expert system programming environment permits the incorporation of expert's heuristics, rules of thumb and empirical knowledge in dealing with peculiar situations or special design goals. Moreover, algorithmic processes commonly used in other aids can be called by TRALI and applied whenever appropriate. For example, TRALI evaluates Webster's formula for estimating user delay as one step in design (11).

Knowledge-based expert systems are computer programs based on Artificial Intelligence (AI) techniques and designed to reach the level of performance of a human expert in a limited problem solving domain. A general review of expert systems appears in Ref. 5; a review of existing and potential applications in transportation systems engineering appears in Ref. 6. These programs offer significant advantages in problem domains where explicit algorithms do not exist or where traditional computer programs provide only restricted problem solving capabilities.

The bulk of this paper will describe the structure and operation of the traffic signal setting expert system assistant. In the process of describing TRALI, various aspects of expert system design will be illustrated. TRALI was developed as an experimental prototype to explore the potential of expert systems; it is not a production level system suitable for immediate field application. Nevertheless, it serves to illustrate the characteristics of expert system implementations, and it could serve as a prototype for the next generation of software for traffic signal setting. An example application of TRALI also appears. A concluding section summarizes the advantages and disadvantages of expert systems for transportation problems.

**Overview of TRALI**

Designing the operation of a traffic signal is not an algorithmic process. The experience and knowledge that the traffic engineer has with the problem are important factors that affect the final solution. The process begins with a description of the geometry of the intersection and of the set of valid movements that takes place. Then, data is collected about the volume and composition of the flows and about other variables that are considered to be important for the design process. Once enough information is available, the traffic engineer proposes some alternatives that are appropriate for the problem. In contrast, a computer program might enumerate all possibilities for simple intersections. These alternatives are evaluated using simulation packages or appropriate models (11), and the results are analyzed. Some of the best alternatives are evaluated more carefully and the process is repeated until a sufficiently good alternative is found. In practice, interactions with other intersections are very important, but our prototype considers the case of an isolated intersection.

TRALI does not enumerate all possible solutions for a given intersection. It is intended to mimic a search by a knowledgeable traffic engineer and to follow a more direct procedure towards a good design alternative. Traffic system knowledge is expressed as rules in the knowledge-base
of the program, whereas procedures for search are developed from Artificial Intelligence techniques.

An overview of the main tasks used by TRALI in designing the operation of a traffic signal for an isolated intersection is shown in Fig. 1. These five steps involve:

1. Conflict Determination.—
   TRALI receives information of the flows at the intersection (volumes, angles, speeds, composition of traffic) and a description of the intersection layout (approaches, lanes). Then, any crossing conflicts occurring between pairs of flows are identified. Merging conflicts are not considered in the current version of TRALI.

2. Proposal of a Phase Distribution.—
   The second step consists of grouping traffic flows into signal phases. The purpose is to determine the number of phases in the cycle and the flows assigned or allowed to move in each phase. The formation of a phase distribution consists of two tasks: Selection of a parent flow for each phase, and coupling the other flows to parents. The first task involves the analysis of the flow magnitudes and conflicts. A new phase is justified only when there are flows in conflict whose volume is high and cannot be joined to an existing phase. Once the final set of parents has been determined, each one of the remaining flows must be coupled to one or more parents. Each child is joined to the parents with which it has stronger similarities and weaker conflicts. The number of possible phase distributions is large; however most of them may be eliminated by using simple rules.

3. Determination of the Optimum Cycle Length and Periods.—
   The candidate phase distribution is evaluated using Webster’s formulas (11). The current version of TRALI calculates the optimum cycle length following the procedure described in Ref. 12.

FIG. 1.—Overview of TRALI
4. Calculation of figures of merit.—
Given the proposed phase distribution and the optimum cycle length, three different figures of merit are evaluated: Average delay per lane, average queue length, and total delay per cycle. Their calculation is an algorithmic procedure which is based on formulas derived from classical queueing theory. For transient queueing behavior in peaking periods other methods could be used. The expert system could calculate these measures for either the optimum cycle or for a range of cycles selected by the user.

5. Modifications to data and results.—
After the proposed phase distribution has been evaluated, the traffic engineer has the opportunity of making changes to the data and the results. Modifications might include changing the magnitudes of flows, creating a new phase, splitting an old phase, or moving a flow from one phase to another. Such changes permit sensitivity analysis of the design for different hours of the day or for different turn restrictions. Also, the results may be modified because some specific characteristics of the intersection were not considered by the rules in the knowledge-base.

The first four steps are realized sequentially for the first proposed design. However, new solutions may require repetition of these steps. Selection of TRALI’s tasks that must be repeated depends on the traffic engineer’s requests. For example, changing flow levels requires reanalysis of conflicts, the phase distribution and the figures of merit. In Fig. 2, a network representation of the different levels of interaction is shown.

Changes in the flows can affect the magnitudes or the angles of entry and exit. Variations in the magnitudes represent different traffic volumes, while modifications in the angles are caused by turn restrictions. If flow angles are not changed, conflicts remain unchanged, and this task is skipped. In contrast, if the traffic engineer imposes or relaxes turn restrictions, new conflicts must be determined.
If the traffic engineer decides to change the phase distribution proposed by TRALI, he has two options: Either moving a flow from one phase to another, or inserting a flow in a new phase without removing it from the phase where it is located. With these options, splitting may be accomplished, and special phase distributions may be created. Once the traffic engineer has done all the modifications to the phase distribution, TRALI evaluates the optimum cycle for this alternative and computes the figures of merit associated with it.

Interactive analysis is repeated until the traffic engineer finds that a good solution has been identified. Then, he or she may want to modify or insert new rules in the knowledge-base that reflect the experiences derived from the design process. These rules will be useful in the future, because they will determine the way TRALI will generate alternatives for other intersections.

In the current state of TRALI, no mechanisms are available either to evaluate the uncertainty of rules and the effect of input data uncertainty or to apply multiple design criteria. However, extending TRALI in these ways is possible as noted in the following.

**STRUCTURE OF TRALI AND OPS5**

TRALI is written in OPS5, a language for development of expert systems (1,3). This programming environment was chosen because of its flexibility with respect to problem solving strategies. This characteristic is particularly useful for traffic systems problems where multiple competitive tasks are present. Fig. 3 shows the four main components of the OPS5 system:

- The user interface, which is used by the traffic engineer to interact with TRALI
- The context, which is like a "blackboard" or temporary store where information about the current problem is written,

![Diagram](image-url)
• The knowledge-base, which has the rules that tell the system how to solve the problem and update the context.

• The inference machine, which is in charge of matching the rules of the knowledge-base with the elements of the context, deciding which rule dominates over the others, and "firing" or carrying out the action prescribed by a rule.

Each of these components are described more extensively in the following and in Ref. 1.

User Interface.—When the program is running, interaction with the user is by means of menus that allow the selection of activities that are to be performed. The user interface also permits the traffic engineer to change the program control and to insert or remove information from the context. The initial data describing a problem can be read from a file created previously or received from the keyboard. In the latter case, a LISP routine is called to prompt for the data required.

Context.—The context is the part of the system that keeps the information about a particular intersection and the current state of problem solving. It may be compared with a blackboard where information is readily written, read or erased. This information is stored in the form of objects that are referenced by their names. An object resembles the record data type in PASCAL. It has an arbitrary number of fields called attributes that store data; each attribute name is preceded by the sign "^". For example, the information about the flows at the intersection is kept in flow objects like this one:

```
(flow
  ^name        west_right
  ^value       200
  ^angle_in    180.0
  ^angle_out   90.0
  ^lane_in     2
    :          :
)
```

This object describes a flow named west_right with volume of 200 vehicles/hr that enters the intersection in lane 2 at an angle of 180 degrees and leaves it at an angle of 90 degrees. Similarly, an object describing a goal to be achieved is represented as:

```
(goal
  ^name             get_phase_distribution
  ^status           active
)
```

In this case, the object indicates that the goal for generating the phase distribution is active and the inference engine should give it attention.

Objects are created, removed or modified by the program. Objects are also used by OPS5 to see what should be done next. In contrast with algorithmic programs, the control of process is not defined in advance. Each time an action is executed, the program examines the context and
determines which rules are now candidates to be fired. Therefore, the context stores not only data and results, but also process knowledge.

**Knowledge-Base.**—TRALI is a rule-based expert system. Its knowledge-base is formed of rules or *productions* that indicate what to do and when. A production resembles an IF-THEN rule of FORTRAN or PASCAL:

```
IF (condition 1) (condition 2) ... (condition n)
THEN (action 1) (action 2) ... (action n)
```

The Left-Hand-Side (LHS) of the rule is a set of conditions that needs to be satisfied by the objects of the context before the rule is executed, while the Right-Hand-Side (RHS) indicates the actions to undertake when the rule is fired (i.e., the LHS is evaluated to be true and the specified actions are taken). Over 200 rules are included in TRALI’s knowledge base.

All the rules in the knowledge-base are at the same level in TRALI, regardless of the order in which they were inserted to the system. This characteristic of expert systems permits the continuous enrichment of the knowledge-base from the experiences of the experts. In the next section we will describe with more detail an example of how the knowledge is stored in the knowledge-base.

**Inference Machine.**—This part of the system is in charge of matching the objects in the context and the conditions in the rules. During execution all rules are analyzed and a “conflict set” of those rules whose LHS are matched with objects in the context is determined. These are the candidate rules to be fired. Then a dominant rule is chosen depending on the strategy used by the inference machine. After this rule is fired, a new set of matching rules is formed. The process is repeated until no rule can be satisfied or until a production indicates the end of the session. OPS5 is responsible for doing the matching and execution; the traffic engineer does not need to know the computational details of how this is done. However, the user may trace the whole session and go backwards if this is desired; also, the next rule to be fired may be changed by means of altering the objects in the context. Indeed, the traffic engineer should do this to insure that he or she understands the reasons particular designs are to be recommended and the associated computations.

TRALI has a hybrid or combined control strategy. Some sub-tasks are done sequentially resembling the execution of an algorithmic program. For example, sequential control is utilized during evaluation of the figures of merit for a phase distribution. A second type of control strategy called *forward-chaining* (6) is used for general design and for representing dependencies among different activities. This strategy attempts to reach goals by starting from known information.
For example, conflicts can be determined only after turn movements have been defined and differences in the flow angles have been computed. The only requirement for an activity to be executed is that all its preceding activities have been accomplished previously. However, the order in which goals are activated is not predetermined.

**Knowledge Representation in TRALI**

Of the four components described in the previous section, the knowledge-base is the most interesting. We may say that it is the "body" of the expert system because it has the knowledge of what should be done and when. Knowledge about both the process and domain of signal setting in included. The first type of knowledge, called process knowledge is expressed in the form of IF-THEN rules, and represents control strategies. TRALI performs different actions depending on what goals have been accomplished, when they were satisfied, and what goals remain active in the context. For example, the following production indicates that the tasks of generating the child flows for each phase should be done (i.e., becomes "active") only after the parents of the phases have already been determined (i.e., the end_get_parents goal has been "processed"):

```
(p control_to_children
 ;
 ; This production transfers the control to the block where the
 ; children for each phase are generated.
 ;
 (goal 'name end_get_parents 'status processed)
 --)
 (make goal 'name end_generate_children 'status active)
 (make goal 'name get_children 'status active 'type initialize)
 (make goal 'name deactivate_goals 'status active)
)
```

The other type of knowledge, the domain knowledge is also written in the form of IF-THEN rules. For example, the next production specifies that a flow is a left_turn when the difference between the angle at which it leaves the intersection and the angle at which it enters the intersection is greater than 0 and less than 180 degrees:

```
(p get_left_turns_1
 ;
 ; This production defines the left turns when the angle
 ; angle of exit is greater than the angle of entry
 ;
 (goal 'name give_flow_turns 'status active)
 (flow 'difference_angles { > 0 < 180 }
 {  'turn_movement nil) (flow})
 --)
 (modify (flow) 'turn_movement left)
)
```
A similar production considers the case when the angle of exit is less than the angle of entry. Angles are measured with respect to a fixed system of cartesian coordinates. Unfortunately, conflicts cannot be identified with this system of coordinates because they depend also on the direction of the movements and on the relative position of the flows in the lanes when approaching the intersection.

The following production illustrates some of the heuristic knowledge used to couple children and parent traffic movements. The first condition indicates that the rule may be executed only when the task of joining children and parents is active. The third condition eliminates the possibility of putting a flow in more than one phase. The fourth and the fifth conditions consider the heuristic knowledge that a child would be better coupled with flows that are in the same approach (difference in the incidence angle is 0, 180 or -180°), or in similar approaches (absolute difference in their angles of incidence less than or equal to 20°). If there is a better parent, the flow is not coupled with the candidate that is being analyzed. Finally, the last condition indicates that a child will be coupled with the parent flow having the most similar incidence angle. When the rule is executed a new phase object is created.

```
(p get_children_3
 ; This production gets the children for any parent using the
 ; criteria that a child flow will be coupled with a parent
 ; only when it cannot be coupled with a parent that
 ; has a smaller difference in their angles of incidence
 ;
 (goal 'name join_children 'status active)
 { (flow 'name (flow_name) 'status active 'value (value) 'angle_in
   (angle_flow)) (flow) }
 -(phase 'flow_name (flow_name))
 (phase 'phase_name (phase_name))
 (child 'child_name (flow_name) 'phase_name (phase_name)
   'difference_in (difference) )
 -(child 'child_name (flow_name) 'phase_name () (phase_name)
   'difference_in { (= 20 )= -20 })
 -(child 'child_name (flow_name) 'phase_name () (phase_name)
   'difference_in (( 180 -180 0 )) )
 -(child 'child_name (flow_name) 'difference_in ( (difference))
 ->)
 (make phase 'phase_name (phase_name) 'flow_name (flow_name)
   'value (value) 'angle_in (angle_flow) 'status active)
 (modify (flow) 'status processed)
)
```

To determine the existence of conflicts between flows at the intersection would require many rules to be specified. To simplify this determination, another mechanism called a decision table is used to identify conflicts. A decision table evaluates a conclusion or indicates an action based upon specific combination of the satisfaction or violation of a set of conditions (2,4). An example of a very simple decision table is illus-
trated in Appendix I. This decision table indicates that only when the three conditions relating the angles of A and B are satisfied simultaneously is a new conflict identified and a conflict record or “object” should be created. If one of the conditions is not satisfied, nothing is done. In this decision table, only one combination determined the creation of the object representing the conflict. However, TRALI has a network of eight decision tables, some of them with up to eight different options.

**Example Application**

TRALI is designed to be capable of working with intersections of irregular geometry. However, the example application of a four-leg intersection is used here because it is the most common type of intersection in urban networks. The example appears in Wohl and Martin (Ref. 12, p. 459), and is shown in Fig. 4. All volumes are in equivalent passenger-car units.

The following data were given:

- **Flows**
  Name, magnitude (in vehicles/hr, VPH), angles when entering and when leaving the intersection (degrees), and lane when approaching the intersection.

- **Lanes**
  Name and approach in which the lane is located.

- **Approaches**
  Name, speed of vehicles (miles/hour) and width to be traversed (ft).

![FIG. 4.—Example of an Isolated Intersection](117)
Additional data required to solve the problem was assumed in the application of TRALI:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver’s Reaction Time</td>
<td>1 sec</td>
</tr>
<tr>
<td>Average Length of Vehicle</td>
<td>17 ft</td>
</tr>
<tr>
<td>Braking Deceleration</td>
<td>12 ft/s²</td>
</tr>
<tr>
<td>Time Lost by First Vehicle in Queue</td>
<td>3.7 sec</td>
</tr>
<tr>
<td>Saturation Flow</td>
<td>1,714 VPH/lane</td>
</tr>
</tbody>
</table>

After doing a classification of the turn movements, TRALI displays the flows shown in Appendix II in which turning movements are named. These turning movement designations are used subsequently in the phase distribution.

Names of flows were given with the intention of being self-descriptive of the type of turn. We can see how the type of turn was correctly determined by the program for each flow. Conflicts at the intersection are shown in Fig. 4. Only “crossing conflicts” are represented because they were considered more important than “merging conflicts.” TRALI determined the 24 conflicts at the intersection and displayed Appendix III to the traffic engineer; up to 91 conflicts might have been possible.

The next task was to propose a phase distribution. As left turn flows are not big, TRALI concluded that some conflicts should be allowed in the same phase and generated the phase distribution shown in Appendix IV. Note that the program generated this phase distribution without enumerating the alternatives. The choice of having two phases seems to be good for this intersection, and therefore, this is the first alternative that the program considers. This phase distribution is also recommended by Wohl and Martin (12).

Calculation of the “optimum cycle” using Webster’s formulas gave a cycle length of 66.3 secs. In this example, the traffic engineer wanted to evaluate a range of cycles. The interaction with TRALI is shown in the following; user responses are underlined:

```
********************************************************************
The value of the optimum cycle is 66.3 seconds
********************************************************************

Do you want to evaluate a range of cycles (yes)?
yes

********************************************************************
Give me the minimum value for ranging ) 40
Give me the maximum value for ranging ) 80
Give me the increment for ranging ) 10
********************************************************************
```

Figures of merit for the optimum cycle are shown in Appendix V, and total intersection delays for the different cycle lengths evaluated are listed in Appendix VI. After analyzing this phase distribution, the traffic en-
engineer wanted to change some data and to try a special phase distribution including a third phase for only one flow:

****************************************************************************
** MAIN MENU FOR THE TRAFFIC ENGINEER **
*****************************************************************************

1. Change the flows at the intersection
2. Change the phase distribution (try splits)
3. Run TRALI with new data
4. Exit the program

Which? 1

MENU FOR CHANGING THE FLOWS

1. Change the magnitude of a flow
2. Change the lane and the angle of incidence
3. Change the angle at which the flow leaves the intersection
4. Exit this menu and go to the main menu

Which? 1

** Give me the name of the flow to correct \textit{southbound} **
** This flow does not exist, try again **

** Give me the name of the flow to correct \textit{southbound\_left} **
\textit{Give me the new magnitude} \textit{50}

MENU FOR CHANGING THE FLOWS

1. Change the magnitude of a flow
2. Change the lane and the angle of incidence
3. Change the angle at which the flow leaves the intersection
4. Exit this menu and go to the main menu

Which? 1

** Give me the name of the flow to correct \textit{westbound\_straight\_3} **
\textit{Give me the new magnitude} \textit{100}
1. Change the magnitude of a flow
2. Change the lane and the angle of incidence
3. Change the angle at which the flow leaves the intersection
4. Exit this menu and go to the main menu

Which? > 4

***************************************************************
** MAIN MENU FOR THE TRAFFIC ENGINEER **
***************************************************************

1. Change the flows at the intersection
2. Change the phase distribution (try splits)
3. Run TRALI with new data
4. Exit the program

Which? > 2

MENU FOR CHANGING THE PHASES

1. Put a flow in a phase deleting it from the previous one
2. Put a flow in a phase without deleting it from other phases (split)
3. Exit this menu and go to the main menu

Which? > 1

** Give me the name of the flow to correct > southbound_left
**Give me the number of the new phase > 3

-> For the flow that you chose for a phase,
   I would recommend to include also these:

   Flow> SOUTHBOUND_RIGHT
   Flow> EASTBOUND_STRAIGHT_2
   Flow> NORTHBOUND_LEFT
   Flow> NORTHBOUND_RIGHT
   Flow> EASTBOUND_STRAIGHT_1
   Flow> WESTBOUND_RIGHT
   Flow> EASTBOUND_RIGHT
   Flow> SOUTHBOUND_STRAIGHT

In this exchange, note that the system suggested including other flows that do not have conflicts with the flow southbound_left. This recommendation is intended to use the new phase with more volume in order
to reduce the queues that would be caused by having an extra phase with small volume. The alternative of having three phases showed to be worse than the original solution. Figures of merit for the new “optimum” cycle of 93.0 secs are shown in Appendix VII.

Finally, the traffic engineer decided to exit the program:

********************************************************* *************
** MAIN MENU FOR THE TRAFFIC ENGINEER **
*********************************************************

1. Change the flows at the intersection
2. Change the phase distribution (try splits)
3. Run TRALI with new data
4. Exit the program

Which? 4

** This is the END **

end -- explicit half

237 PRODUCTIONS (1648 / 4048 NODES)
603 FIRINGS (1320 RHS ACTIONS)
110 MEAN WORKING MEMORY SIZE (164 MAXIMUM)
14 MEAN CONFLICT SET SIZE (84 MAXIMUM)
380 MEAN TOKEN MEMORY SIZE (680 MAXIMUM)
NIL

The final information displayed for the session indicates that TRALI has a total of 237 rules, and that during the session 603 rules were executed with 1,320 actions (“RHS actions”) prescribed. Each time the inference machine selected a dominant rule there was an average of 14 possible rules to choose from. This indicates that many different options might have occurred depending on the information contained in the context.

CONCLUSIONS

TRALI is an experimental system that is not yet ready for field application. The system is not complete with respect to evaluation of competing figures of merit and the speed of execution is relatively slow. However, it illustrates a radically different approach to the problem of developing computer aids for the design of the operation of a traffic signal in an isolated intersection. TRALI is intended to generate solutions resembling the design process followed by a traffic engineer. Only a few alternatives that might be the best solutions are analyzed. This process is controlled by defining multiple goals, as supported by OPS5, the expert system development framework chosen for TRALI.

The use of a knowledge-based expert system approach has a number of advantages over conventional programs:
• The system is not restricted to work with intersections of regular geometry. The program was developed for analyzing intersections with any number of approaches and with any flow movements.

• As search of possible phase distributions is done through the application of heuristic rules in the knowledge-base, the system may be applied to intersections of many flows where enumeration of alternatives would be costly.

• Expert systems are applicable when selection of the best alternative contemplates multiple competing figures of merit. For example, a good design might not be that which minimizes total delay, but one having an acceptable delay and short average queues in the lanes. Diagnosis of how good a design is can be implemented in the form of IF-THEN rules in the knowledge-base.

• An important characteristic of expert systems is that the knowledge contained in the knowledge-base can be adapted and enriched continuously. This flexibility permits knowledge from several traffic engineers to be used by the system, although it is important to resolve conflicts of opinion between experts.

• Avoiding the need for programming the sequence of events and a complete enumeration of all possible cases greatly simplified the development of the program.

However, there are some difficulties when approaching the problem by means of an expert system:

• Knowledge needs to be formalized. This task implies that traffic engineers need to identify the rules they use for designing alternatives.

• Portability of the system is restricted to hardware that supports the chosen expert system framework. Numerous development environments are now available, so this problem will be less in the future.

• Easy connections between new expert systems and existing simulation or optimization programs require additional work. In particular, the development framework OPS5 is not appropriate for doing many computations.

The addition of greater computational power is particularly important in traffic engineering, with the consideration of network level effects. However, we believe that these obstacles will be overcome and that systems like TRALI should be developed for field applications.

ACKNOWLEDGMENTS

We would like to thank Steven Fenves, Mary Lou Maher and Daniel Rehak of Carnegie-Mellon University for helpful comments on this research.
APPENDIX I.—DECISION TABLE FOR FINDING A TYPE OF CONFLICT BETWEEN A PAIR OF FLOWS A AND B

<table>
<thead>
<tr>
<th>CONDITIONS</th>
<th>OPTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Incidence angle of A is less than (360 − leaving angle of A)</td>
<td>True</td>
</tr>
<tr>
<td>2) Incidence angle of B is greater than the incidence angle of A</td>
<td>True</td>
</tr>
<tr>
<td>3) Incidence angle of B is less than (360 − leaving angle of A)</td>
<td>True</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Create a conflict of A and B</td>
<td>X</td>
</tr>
<tr>
<td>2) Do nothing</td>
<td>X</td>
</tr>
</tbody>
</table>

APPENDIX II.—TABLE OF FLOW DATA

TABLE OF FLOWS

<table>
<thead>
<tr>
<th>FLOW VALUE</th>
<th>ANGLE-IN</th>
<th>ANGLE-OUT</th>
<th>TYPE TURN</th>
</tr>
</thead>
<tbody>
<tr>
<td>WESTBOUND_STRAIGHT_3</td>
<td>583</td>
<td>180.0</td>
<td>180.0</td>
</tr>
<tr>
<td>SOUTHBOUND_STRAIGHT</td>
<td>315</td>
<td>270.0</td>
<td>270.0</td>
</tr>
<tr>
<td>WESTBOUND_RIGHT</td>
<td>52</td>
<td>180.0</td>
<td>90.0</td>
</tr>
<tr>
<td>WESTBOUND_STRAIGHT_4</td>
<td>519</td>
<td>180.0</td>
<td>180.0</td>
</tr>
<tr>
<td>WESTBOUND_LEFT</td>
<td>11</td>
<td>180.0</td>
<td>270.0</td>
</tr>
<tr>
<td>EASTBOUND_RIGHT</td>
<td>42</td>
<td>0.0</td>
<td>270.0</td>
</tr>
<tr>
<td>EASTBOUND_STRAIGHT_1</td>
<td>408</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>EASTBOUND_STRAIGHT_2</td>
<td>398</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>EASTBOUND_LEFT</td>
<td>39</td>
<td>0.0</td>
<td>90.0</td>
</tr>
<tr>
<td>SOUTHBOUND_RIGHT</td>
<td>15</td>
<td>270.0</td>
<td>180.0</td>
</tr>
<tr>
<td>SOUTHBOUND_LEFT</td>
<td>39</td>
<td>270.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NORTHBOUND_RIGHT</td>
<td>22</td>
<td>90.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NORTHBOUND_LEFT</td>
<td>53</td>
<td>90.0</td>
<td>180.0</td>
</tr>
<tr>
<td>NORTHBOUND_STRAIGHT</td>
<td>420</td>
<td>90.0</td>
<td>90.0</td>
</tr>
</tbody>
</table>

APPENDIX III.—CONFLICTS DETERMINED BY TRALI

TABLE OF CONFLICTS

<table>
<thead>
<tr>
<th>FLOW WITH FLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>EASTBOUND_STRAIGHT_1</td>
</tr>
<tr>
<td>EASTBOUND_STRAIGHT_2</td>
</tr>
<tr>
<td>EASTBOUND_LEFT</td>
</tr>
<tr>
<td>EASTBOUND_LEFT</td>
</tr>
<tr>
<td>SOUTHBOUND_STRAIGHT</td>
</tr>
<tr>
<td>SOUTHBOUND_STRAIGHT</td>
</tr>
<tr>
<td>EASTBOUND_STRAIGHT_1</td>
</tr>
<tr>
<td>EASTBOUND_STRAIGHT_2</td>
</tr>
</tbody>
</table>
APPENDIX IV.—PHASE DISTRIBUTION PROPOSED BY TRALI

*********************************************************************************************
TABLE OF PHASES
*********************************************************************************************

<table>
<thead>
<tr>
<th>PHASE</th>
<th>FLOW</th>
<th>VALUE</th>
<th>ANGLE-IN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 EASTBOUND_LEFT</td>
<td></td>
<td></td>
<td>0.0</td>
</tr>
<tr>
<td>1 EASTBOUND_STRAIGHT_2</td>
<td>39</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>1 EASTBOUND_STRAIGHT_1</td>
<td>408</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>1 EASTBOUND_RIGHT</td>
<td>42</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>1 WESTBOUND_LEFT</td>
<td>11</td>
<td>180.0</td>
<td></td>
</tr>
<tr>
<td>1 WESTBOUND_STRAIGHT_4</td>
<td>519</td>
<td>180.0</td>
<td></td>
</tr>
<tr>
<td>1 WESTBOUND_RIGHT</td>
<td>52</td>
<td>180.0</td>
<td></td>
</tr>
<tr>
<td>1 WESTBOUND_STRAIGHT_3</td>
<td>583</td>
<td>180.0</td>
<td></td>
</tr>
<tr>
<td>2 NORTHBOUND_STRAIGHT</td>
<td>420</td>
<td>90.0</td>
<td></td>
</tr>
<tr>
<td>2 NORTHBOUND_LEFT</td>
<td>53</td>
<td>90.0</td>
<td></td>
</tr>
<tr>
<td>2 NORTHBOUND_RIGHT</td>
<td>22</td>
<td>90.0</td>
<td></td>
</tr>
<tr>
<td>2 SOUTHBOUND_LEFT</td>
<td>39</td>
<td>270.0</td>
<td></td>
</tr>
<tr>
<td>2 SOUTHBOUND_RIGHT</td>
<td>15</td>
<td>270.0</td>
<td></td>
</tr>
<tr>
<td>2 SOUTHBOUND_STRAIGHT</td>
<td>315</td>
<td>270.0</td>
<td></td>
</tr>
</tbody>
</table>

APPENDIX V.—FIGURES OF MERIT FOR “OPTIMUM” CYCLE OF 66.3 SECONDS

*********************************************************************************************
Results for optimum cycle of 66.3 seconds
*********************************************************************************************

<table>
<thead>
<tr>
<th>LANE</th>
<th>APPROACH</th>
<th>VOLUME</th>
<th>DELAY</th>
<th>AVERAGE_QUEUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EAST_BOUND</td>
<td>450</td>
<td>16.1</td>
<td>4.0</td>
</tr>
<tr>
<td>2</td>
<td>EAST_BOUND</td>
<td>437</td>
<td>15.7</td>
<td>3.9</td>
</tr>
<tr>
<td>3</td>
<td>WEST_BOUND</td>
<td>594</td>
<td>22.8</td>
<td>6.4</td>
</tr>
<tr>
<td>4</td>
<td>WEST_BOUND</td>
<td>571</td>
<td>21.2</td>
<td>5.9</td>
</tr>
<tr>
<td>5</td>
<td>NORTH_BOUND</td>
<td>495</td>
<td>22.7</td>
<td>5.5</td>
</tr>
<tr>
<td>6</td>
<td>SOUTH_BOUND</td>
<td>369</td>
<td>17.2</td>
<td>3.5</td>
</tr>
</tbody>
</table>

** Total Delay Optimum = 15.9
APPENDIX VI.—COMPARISON OF TOTAL DELAY FOR DIFFERENT CYCLE LENGTHS

<table>
<thead>
<tr>
<th>CYCLE LENGTH (seconds)</th>
<th>TOTAL DELAY (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.0</td>
<td>20.7</td>
</tr>
<tr>
<td>50.0</td>
<td>16.2</td>
</tr>
<tr>
<td>60.0</td>
<td>15.7</td>
</tr>
<tr>
<td>70.0</td>
<td>16.1</td>
</tr>
<tr>
<td>80.0</td>
<td>16.9</td>
</tr>
</tbody>
</table>

APPENDIX VII.—FIGURES OF MERIT FOR “OPTIMUM” CYCLE OF 93.0 SECONDS

Results for optimum cycle of 93.0 seconds

<table>
<thead>
<tr>
<th>LANE</th>
<th>APPROACH</th>
<th>VOLUME</th>
<th>DELAY</th>
<th>AVERAGE_QUEUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EAST_BOUND</td>
<td>450</td>
<td>24.3</td>
<td>6.3</td>
</tr>
<tr>
<td>2</td>
<td>EAST_BOUND</td>
<td>437</td>
<td>23.7</td>
<td>6.0</td>
</tr>
<tr>
<td>3</td>
<td>WEST_BOUND</td>
<td>111</td>
<td>16.2</td>
<td>1.5</td>
</tr>
<tr>
<td>4</td>
<td>WEST_BOUND</td>
<td>571</td>
<td>32.8</td>
<td>9.2</td>
</tr>
<tr>
<td>5</td>
<td>NORTH_BOUND</td>
<td>495</td>
<td>33.0</td>
<td>8.1</td>
</tr>
<tr>
<td>6</td>
<td>SOUTH_BOUND</td>
<td>380</td>
<td>23.6</td>
<td>4.8</td>
</tr>
</tbody>
</table>

** Total Delay Optimum = 21.1

APPENDIX VIII.—REFERENCES