【土木計画学研究・講演集 №.12　1989年12月】

AN EXPERT SYSTEM FOR INTERSECTION DESIGN AND CONTROL

by Ricardo G. Sigua,* Masao Kuwahara,** and Masaki Koshi***

A knowledge-based expert system for traffic signal setting of isolated intersections was developed. The system can deal with intersections of irregular geometry, from 'T' or 'Y' intersection to complicated multi-leg configurations. Incorporated in the expert system are the expert's heuristics, rules of thumb, and empirical knowledge. Identification of flow conflicts and formation of phases patterns are performed using geometric and traffic rules while algorithmic processes for calculation of cycle length, green splits, and measures of performance are called by the system. Instead of using a commercially available expert system shell, all parts of the system are developed, including an inference engine that can handle both forward and backward chaining processes. The program is written in Pascal in the PC 9801 environment.

INTRODUCTION

The development of artificial intelligence some 30 years ago brought about new frontiers in the usage of computers. The resulting development of expert systems or knowledge-based systems provides a useful tool to resolve some of the pressing needs for productivity in many applications. MYCIN, a popular expert system applied in the medical field, is a computer system which diagnoses bacterial infections of the blood and prescribes suitable drug therapy based on its findings. Expert systems have been widely used in military applications, too. In civil engineering, most of the works done in the past are in the field of structural engineering.[1] Today, however, the potential of expert systems has been recognized in many engineering applications.

*Ricardo Sigua: Student Member, Univ. of Tokyo
**Masao Kuwahara: Regular Member, Univ. of Tokyo
***Masaki Koshi: Regular Member, Univ. of Tokyo

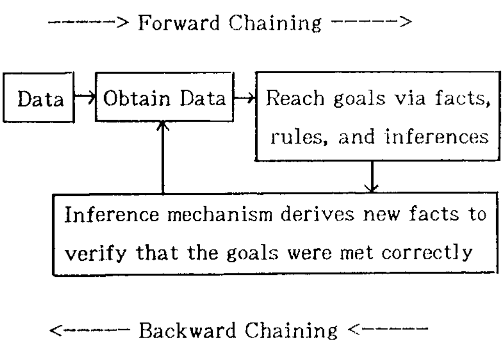
This paper presents an application of expert system in transportation. A classic problem in intersection design and control is considered. Recognizing that there is no exact algorithm that can handle such problem, it is thought that an expert system's potential could be fully utilized. The determination of conflicts and creation of phase patterns requires some expertise. Some heuristics and rules of thumb, which are normally followed by an experienced traffic engineer, are applied. Instead of describing in length what an expert system is, the paper discusses in detail the expert system developed for intersection design and control, referred hereto as ESIDC(Expert System for Intersection Design and Control).

KNOWLEDGE REPRESENTATION IN ESIDC

Inference Engine and Knowledge-base

In a very great deal of expert systems work, there has been a conscious effort to divide the problem into 2 parts, namely, the inference engine and the knowledge-base. The idea is that the inference engine is the general purpose thinking machine and the knowledge-base is that about which the engine shall think.

o **Inference Engine** – The inference engine of ESIDC was developed to perform either forward or backward chaining. Forward chaining simply makes use of initial data to arrive at a conclusion while backward chaining moves in the opposite direction. The diagram below shows how forward and backward chaining operate.[2] ESIDC, however, performs mostly forward chaining process since initial data are normally given.



o **Knowledge-base** – The knowledge-base of ESIDC consists of set of rules in the form of IF-THEN statements. One form of this is:

premise ---> conclusion.

An example of which is:

'If traffic flow volume is greater than 100
then traffic flow is major'

to categorize major and minor traffic flows.

Another common form is:

situation ---> action.

An example of this rule is:

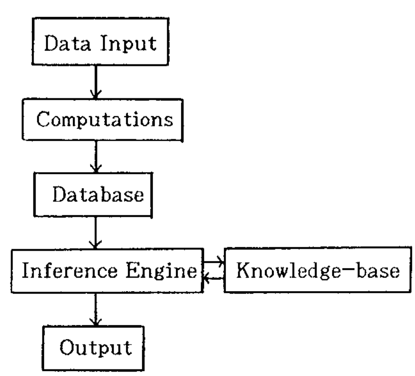
'If there is no conflict then include in phase'

in order to couple a traffic movement to a set of movements in a given signal phase.

Instantiation

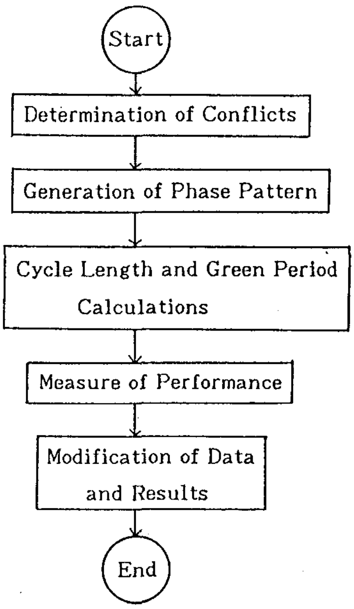
A goal or target has to be set for the inference engine to search through the numerous rules in the knowledge-base. Once a goal has been defined, initial data are necessary to instantiate a rule in the knowledge-base. The key to the system's intelligence is that each instantiation of a rule causes the future instantiation of other rules. Any rule instantiation is a chain reaction, until the prescribed goal is reached.

In ESIDC, database are generated based on data input and computations. This database serves as initial facts/data to instantiate or fire a rule in the knowledge-base after which subsequent rules are instantiated. The knowledge representation of ESIDC is shown in the flow chart below:



OVERVIEW OF ESIDC

The flow of main tasks performed by ESIDC is shown below:

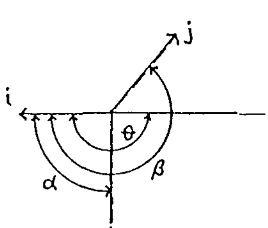


Determination of Conflicts – After inputting necessary traffic data, all types of conflict – crossing, diverging, and converging, for different pairs of flows are identified. In ESIDC the following rules are utilized to determine the type of conflict:

'if origins of 2 flows are the same
then conflict is diverging'

'if exits of 2 flows are the same
then conflict is converging'

To determine crossing conflicts, the following 3 angles β, θ, and α, in relation to movements i and j, have to be evaluated:



Crossing conflicts are found if the rule:

'if $\beta > \theta$ and $\theta > \alpha$ then conflict is crossing'

is satisfied.

Generation of Phase Pattern – The number of phases and the flows allowed to move in each phase are determined by considering the major flows or flows with high volumes first. Then the minor flows are coupled to them afterwards. In ESIDC, the phase patterns are not enumerated beforehand but they are generated depending on the geometric and traffic rules contained in the knowledge–base.

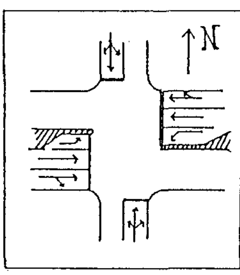
Cycle Length and Green Period Calculations – The algorithm for computation of cycle length and green periods follows the procedure proposed by Akcelik in Ref 3. This is preferred to the conventional method by Webster(Ref 4) because treatment of overlapping movements using Webster's method proved very complicated.

Measures of Performance – After determining the phase distribution and computing the cycle length and green periods, 2 different measures of performance are evaluated: delay and queue length. Individual delay for each movement, total intersection delay, and average delay per vehicle are computed. Queue lengths at the start of green period are also evaluated.

Modification of Data and Results – After the evaluation of measures of performance, the user can make changes to the original data or can test other possible phase patterns.

SAMPLE RUN

Although ESIDC can analyze any irregular intersection, a normal 4-leg is presented here as an example. The system is designed to be menu-driven to facilitate interaction with the user. ESIDC starts with a main menu. Data input can be performed by choosing entry #1 in the menu. Necessary inputs are traffic volume, saturation flow, entry and exit directions, movement's starting loss, end gain, and intergreen period. The figure below shows the configuration of the intersection and the movements to be considered. The right turners are included in the through movements although they can be treated separately.(ESIDC deals with the practice of driving on the right side of the road.)



1 750(1800)	6 360(1800)
2 690(1800)	7 410(1800)
3 250(1800)	8 90(1600)
4 120(1800)	9 395(1800)
5 320(1800)	10 60(1600)

Intersection Layout

Movements with volume and sat. flow(enclosed)

A sample of entering data for movement 1 is shown below. Other movements are entered in the same way.

```
M A I N   M E N U

1. Input Data
2. Run Test Data
3. Modify Flow Data
4. Modify Phasing
5. Run Program
6. Exit Program

   Enter Choice [1-6]: 1
```

```
No. of Movements? 10
Press CAP lock!!!
Movement No. 1
  Approach Leg:[N,E,S,W,NE,NW,SE,SW] E
  Exit Leg:[N,E,S,W,NE,NW,SE,SW] W
  Volume: 750
  Saturation Flow: 1800
  Intergreen: 5
  Start loss: 2
  End Gain: 3
  Practical Degree of Saturation: 0.90
Movement No. 2
  Approach Leg:[N,E,S,W,NE,NW,SE,SW] _
```

After data input, the system can now be run by selecting entry #5. The first set of outputs of the sample run are summaries of the flow information and a table of conflicts for each pair of flows. In the flow table, the directions, whether through, left, or right, are determined correctly by ESIDC. The conflict table gives a summary of the type of conflict for each pair of flows – diverging, converging, crossing, and no conflict. ESIDC generated the initial 3-phase pattern 'appropriate' for the sample problem. Movements 6 and 8 are included in phase 3 because of low volumes. It is assumed that filtering is allowed in this case.

After determining this phase distribution, a summary of flow data is given. The starting and terminating phases can be best understood using a critical search diagram as shown. The links represent the movements while the nodes represent the phases. The procedure for the determination of critical movements, cycle time, and green splits is based on Akcelik's method. Based on the required time the critical movements are found to be movements 1, 3, and 7. This can be checked with the critical search diagram and in the Calculation Table showing estimated required time for each movement. The optimum cycle time is 90 sec. giving an average delay of 39 sec. per veh. and average queue length of 6.4 veh. (Other details of the output are not shown.)

Session for the Initial Scheme

Flow Table:

Movement	Entry	Exit	Direction
1	E	W	through
2	E	W	through
3	E	S	left
4	W	N	left
5	W	E	through
6	W	E	through
7	N	S	through
8	N	E	left
9	S	N	through
10	S	W	left

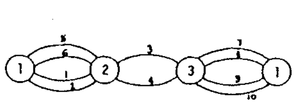
Conflict Table:

	1	2	3	4	5	6	7	8	9
1									
2	dl								
3	dl	dl							
4	cr	cr	no						
5	no	no	cr	dl					
6	no	no	cr	dl	dl				
7	cr	cr	co	cr	cr	cr			
8	cr	cr	cr	cr	co	co	dl		
9	cr	cr	cr	co	cr	cr	no	cr	
10	co	co	cr	cr	cr	cr	cr	no	dl

Note: dl - diverging co - converging

cr - crossing no - no conflict

Final phases:
Phase 1: 1 2 5 6
Phase 2: 3 4
Phase 3: 7 9



Filtering added:
Phase 1: 1 2 5 6
Phase 2: 3 4
Phase 3: 7 8 9 10

Critical Search Diagram

Flow Data:

Move	SP	TP	IG	Volume	Sat_Flow	L	Xp
1	1	2	5	750	1800	4	0.90
2	1	2	5	690	1800	4	0.90
3	2	3	5	250	1800	4	0.90
4	2	3	5	120	1800	4	0.90
5	1	2	5	320	1800	4	0.90
6	1	2	5	360	1800	4	0.90
7	3	1	5	410	1800	4	0.90
8	3	1	5	90	1600	4	0.90
9	3	1	5	395	1800	4	0.90
10	3	1	5	60	1600	4	0.90

Calculations:

Move	y_value	u_value	req_time
1	0.42	0.46	50.30
2	0.38	0.43	46.59
3	0.14	0.15	19.43
4	0.07	0.07	11.41
5	0.18	0.20	23.75
6	0.20	0.22	26.22
7	0.23	0.25	29.31
8	0.06	0.06	10.25
9	0.22	0.24	28.38
10	0.04	0.04	8.17

The critical movements are: 1 3 7

Cycle: 90.0 Ave. delay: 39.18sec. Ave. queue: 6.4

At the time of writing this paper, the present system still serves as a tool to the traffic engineer. It may not give the best solution to the problem at the initial run. It may not be obvious at the beginning but the traffic flow data for east-west directions are somewhat imbalance, i.e., movements at the east approach are much larger than that of the west approach. It might be wise to extend the east movements in a new phase. This task can be performed by modifying the existing phase pattern by selecting option #4 in the Main Menu. Another menu appears on the screen for this purpose. Option #2 in the menu can allow inserting a new phase with the desired movement in it. In the continuation of the session run, a new phase is inserted after Phase 1, with movement 1 to be extended in it. As shown in the new phase pattern, movements not in conflict with movement 1 are coupled to it. In this case, movements 2 and 3 are added. This is to maximize the utilization of the new phase. The flow data are now as shown with the critical movements changed to movements 1, 4, and 7. This shift of critical movements is due to the overlapping movements 1 and 3. The scheme is found to give a cycle length of 60 sec. and a better performance with an average delay of about 28 sec. and average queue length of 4.6 vehicles.

Session for the Second Scheme:
Overlapping Movements

M A I N   M E N U

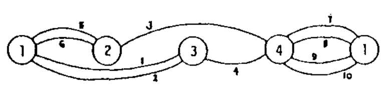
1. Input Data
2. Run Test Data
3. Modify Flow Data
4. Modify Phasing
5. Run Program
6. Exit Program

Enter Choice [1-6]: 4

```
Existing Phase Pattern:

    1:    1   2   5   6
    2:    3   4
    3:    7   8   9   10

Menu for Changing Phase Patterns

    1. Put flow/s in a new phase
       and deleting it from the previous phase

    2. Put flow/s in a new phase without
       deleting it from other phases

    3. Retrieve original phase pattern

    4. Go back to main menu

         Enter Choice [1-4]: 2
Insert new phase after Phase # ?1
Movement to be included in new phase ? 1


         Phase 1:  1  2  5  6
         Phase 2:  1  2  3
         Phase 3:  3  4
         Phase 4:  7  8  9  10
```

Flow Data:

Move	SP	TP	IG	Volume	Sat_Flow	L	Xp
1	1	3	5	750	1800	4	0.90
2	1	3	5	690	1800	4	0.90
3	2	4	5	250	1800	4	0.90
4	3	4	5	120	1800	4	0.90
5	1	2	5	320	1800	4	0.90
6	1	2	5	360	1800	4	0.90
7	4	1	5	410	1800	4	0.90
8	4	1	5	90	1600	4	0.90
9	4	1	5	395	1800	4	0.90
10	4	1	5	60	1600	4	0.90



Critical Search Diagram

```
The critical movements are:    1    4    7

Cycle:  60.0 Ave. delay: 27.76sec.    Ave. queue:   4.6
```

In the 2 schemes considered, movements 8 and 10 are not given exclusive phase but are only allowed to filter through the green period during Phase 3 in the first scheme and during Phase 4 in the second scheme. What will happen if these movements are given an exclusive phase? The session continues by selecting option #1 in the menu for changing phasing pattern. A fifth phase is added with movements 8 and 10 in it. It can be seen that this gives a worse scenario with a cycle time of 110 sec., a longer average delay of about 46 sec. and average queue of 8 vehicles.

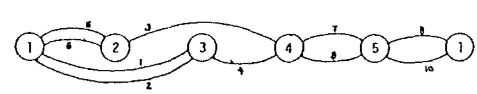
Session for the Third Scheme:

Exclusive Phase for Movements 8 and 10

```
Existing Phase Pattern:

    1:    1   2   5   6
    2:    1   2   3
    3:    3   4
    4:    7   8   9   10

Menu for Changing Phase Patterns

    1. Put flow/s in a new phase
       and deleting it from the previous phase

    2. Put flow/s in a new phase without
       deleting it from other phases

    3. Retrieve original phase pattern

    4. Go back to main menu

         Enter Choice [1-4]: 1
Insert new phase after Phase # ?4
Movements to be included in new phase ?
Movement #?: 8
Movement #?: 10


    Phase 1:  1  2  5  6
    Phase 2:  1  2  3
    Phase 3:  3  4
    Phase 4:  7  9
    Phase 5:  8  10
```

Flow Data:

Move	SP	TP	IG	Volume	Sat_Flow	L	Xp
1	1	3	5	750	1800	4	0.90
2	1	3	5	690	1800	4	0.90
3	2	4	5	250	1800	4	0.90
4	3	4	5	120	1800	4	0.90
5	1	2	5	320	1800	4	0.90
6	1	2	5	360	1800	4	0.90
7	4	5	5	410	1800	4	0.90
8	5	1	5	90	1600	4	0.90
9	4	5	5	395	1800	4	0.90
10	5	1	5	60	1600	4	0.90



Critical Search Diagram

```
The critical movements are:    1    4    7    8

Cycle:  110.0 Ave. delay: 46.19sec.    Ave. queue:   8.0
```

The session ends by going back to the main menu (by selecting option #4 in the current menu) and finally selecting option #6 in the main menu.

```
Menu for Changing Phase Patterns

    1. Put flow/s in a new phase
       and deleting it from the previous phase

    2. Put flow/s in a new phase without
       deleting it from other phases

    3. Retrieve original phase pattern

    4. Go back to main menu

       Enter Choice [1-4]: 4

  M A I N   M E N U

    1. Input Data
    2. Run Test Data
    3. Modify Flow Data
    4. Modify Phasing
    5. Run Program
    6. Exit Program

       Enter Choice [1-6]: 6

Session Ends.
```

CONCLUSION

ESIDC is capable of designing the traffic signal setting of isolated intersections consistent with the design process followed by traffic engineers. The system's execution is very fast although it is implemented in a personal computer. ESIDC is not complete in itself as there are still important aspects of design and control which have to be considered. One aspect of primary importance that has to be added in the future implementation of ESIDC is the problem of maximizing lane allocation in relation to the given design volume and its interaction with phase pattern. As already mentioned in the sample run, ESIDC proposes an initial scheme which does not necessarily give the best solution to the problem. This scheme, however, is one normally followed by a traffic engineer. The author is left with a choice whether to fully automate the system with the sacrifice of less interaction with the user.

REFERENCES

1 Takallou, H.; Knowledge-based Expert Systems: What's Happening in Transportation Engineering, ITE Journal, October 1985, pp. 33-38.

2 Levine, R., Drang, D. and Edelson, B., A Comprehensive Guide to AI and Expert Systems, McGraw-Hill, 1987, p.19.

3 Akcelik, R. Traffic Signals: Capacity and Timing Analysis, ARRB, ARR No. 123, 1981.

4 Webster, F. and Cobbe, B., Traffic Signals, Road Research Technical Paper No. 56, Road Research Laboratory, London, 1966.