Optimising fuzzy logic traffic signal control systems

Stuart Clement
Transport Systems Centre
University of South Australia

Abstract:

The role of traffic signals is evolving to include people-oriented as well as vehicle-oriented objectives. Future signal control systems will need to handle a wide range of input types as improved models (e.g., for cyclist and pedestrian behavior) and other objective functions (e.g., for environmental concerns) become available. Fuzzy logic systems can be built to handle existing vehicle-based objectives and are readily extendable using the hierarchical approach. Some fuzzy logic traffic signal control systems based on vehicle-oriented objectives have been built and tested against simulated models since 1977. Two of the more recent efforts produced at the Transport Operations Research Group, University of Newcastle upon Tyne UK, were designed to incorporate policy-sensitive objectives in the future but tuning the systems proved time-consuming and tedious. Automatic optimisation techniques were sought and genetic algorithms were proposed as a possibility. This paper discusses the elements of fuzzy logic systems and pinpoints those that could be optimisable using genetic algorithms. Methods applicable to optimising fuzzy logic traffic signal control systems offline are suggested.

Contact author:

Stuart Clement
Transport Systems Centre
University of South Australia
GPO Box 2471
ADELAIDE 5001

Telephone: (08) 8302 1862
Fax: (08) 8302 1880
Email: Stuart.Clement@unisa.edu.au
Clement

Introduction

In recent years there has been a paradigm shift in the role of traffic signals towards actively managing road traffic networks to include people-oriented as well as vehicle-oriented objectives. The need for policy objectives ranging beyond catering for motor vehicle movement efficiencies has been recognised politically (Routledge, Kemp and Radia 1996) largely as a result of growing public awareness of the environmental impact of road traffic. Many authorities are now pursuing policies to:

- manage demand and congestion;
- influence mode and route choice;
- improve priority for buses, trams and other public service vehicles;
- provide better and safer facilities for pedestrians, cyclists and other vulnerable road users;
- reduce vehicle emissions, noise and visual intrusion; and
- improve safety for all road user groups

In the UK this led to the establishment of the five-year government-funded research and development effort towards Urban Traffic Management and Control (UTMC) systems. An explanation of the differences between the projected UTMC systems and the older Urban Traffic Control (UTC) systems can be found in Clement, Bell, Cassir and Grosso (1997a).

Fuzzy logic control systems for traffic signals have been reported sporadically since the efforts of Pappis and Mamdani (1977). Further efforts have followed including Nakatsuyama, Nagahashi and Nishisuka (1983); Brubaker and Sheerer (1992); Chiu and Chand (1992); Hoyer and Jumar (1994); Lee, Lee and Leekwang (1994); Janecek and Zargham (1995); Jerabek and Lachiver (1995); Izes, McShane and Kim (1995); Ho (1996); and Zhou, Wu, Lee, Fu and Miška (1997). Until recently all of the systems applied to intersections were designed to minimise disruption to the flow of motor vehicles using traditional measures such as reduction in delays.

Before the UTMC strategy was unveiled, the Transport Operations Research Group (TORG) at the University of Newcastle upon Tyne, England, had begun development of a fuzzy logic (FL) system as a means of optimising and controlling traffic signals. This system was designed to incorporate broad policy objectives into the operational mechanism (Sayers, Bell, Mieden and Busch 1996) but these had not been implemented at time of publication. A variation on this work was completed by Senoner (1997) who based his system on the ideas of Lee, Lee and Leekwang (1994). Initial tests were promising but the tuning of such a system was time-consuming and hence a method for optimising such a controller was sought.

Optimisation using genetic algorithm (GA) technology was a possibility. A GA is an optimisation and search tool that uses probabilistic search methods based on ideas from natural genetics and evolutionary principles (Clement 1997b). GAs are regarded as general purpose and robust. Various methods for using GAs to optimise FL systems...
have been applied to fields as diverse as water level control, semi-active vehicle suspension systems and space-based oxygen production systems.

Many FL systems have been employed in realworld transportation applications (Teodorovic 1994) but the Brubaker and Sheerer (1992) system was the only instance found in the literature of FL applied directly to traffic signals. The system of Zhou et al. (1997) used an FL system to classify traffic situations. Each situation was then used to define parameters for the signal system which used conventional controllers. The other FL systems for signals listed above were built and tested under simulated traffic scenarios ranging from the very stylised in the early examples to close approximations to realworld situations in the later instances. Most were optimised for vehicle-oriented objectives such as reduction in delays. The system developed by Sayers et al. (1996) is designed specifically for realworld application and has the potential to include other objectives. The Senoner (1997) system was tested on a realistic four-intersection simulated network and worked well but would be expensive to implement due to its need for many vehicle detectors. None of the FL signal control systems incorporated automatic optimisation of the fuzzy logic parameters though there are many instances in the literature of GA-optimised FL systems for other control applications. See Cordón, Herrera and Lozano (1996) for a classified review and Clement (1997c) for a discussion of some of these systems.

This paper contains ideas, methods and possible starting points for the development of genetic algorithm-optimised fuzzy logic systems applied to adaptive traffic signal control. An explanation of the elements of a fuzzy logic system is given by taking the example of Brubaker and Sheerer (1992). The reader is given an understanding of the workings of FL systems before the opportunities for and ramifications of optimising FL systems offline are discussed. Finally some suggested approaches to applying GA-optimised FL systems to traffic signal control are given.

**Fuzzy logic systems**

**Overview**

In general most control systems function by monitoring the control problem environment and then reacting appropriately by outputting instructions to one or more devices that influence that environment. This measuring inputs and producing outputs process is continual.

Fuzzy logic control systems are no different in their overall aims but are unique in the methods used to effect control. FL set theory is an extension of classical set theory in that it describes set membership in terms closely linked to natural language rather than in the usual binary descriptions of either belonging to or not belonging to a set. Terms such as long, short, high and low are subjective and therefore open to differing interpretations by a range of people. These can be handled by fuzzy logic where a degree of 'belongingness' to a fuzzy set is assigned to a value of an input variable. For example, Australia would be regarded by almost all people as being a big country.
Defuzzification

The Brubaker and Sheerer (1992) system was a practical application used to control signals of an onramp to a freeway. This example, with some simplifying for clarity, is used here to describe the elements of FL systems and thus highlight for possible optimisation. The inputs to the system were the density and speed of freeway traffic.

Input variables, fuzzy sets, membership functions and fuzzification

Input variables are usually evident from the type of environment to be controlled. For example, traffic signals would have inputs to cater for vehicular traffic, pollution, pedestrians, transit priority etc. We refer to the range of possible values for an input variable as the universe of discourse. In Figure 2 the universe of discourse of the Density variable is 0-1.0 and denoted by X.
The fuzzy sets which describe the input variable in linguistic terms over the universe of discourse are chosen. In the example of Figure 2 the fuzzy sets are light, medium and heavy. Note that each fuzzy set spans the universe of discourse. There is no theoretical restriction on the maximum number of fuzzy sets used for each input variable. Intuitively three would be the minimum. Therefore the number of fuzzy sets is a candidate for optimisation. In practical terms a large number of fuzzy sets impacts heavily on the size of the rulebase and increases the complexity of manually tuning the FL system. Therefore setting an upper limit on the possible number of fuzzy sets would be practical.

![fuzzy sets and their membership functions](image)

**Figure 2 Assigning degrees of membership, \( \mu(x) \) to Density**

The process known as fuzzification takes the crisp input value and translates it to a fuzzy value called the degree of membership, \( \mu(x) \), over the range \([0,1]\) in the fuzzy sets. The membership functions of the fuzzy sets of Figure 2 reflect how traffic engineers may differ in their description of traffic density values or alternatively what ranges of values constitute light, medium and heavy density. Some engineers might say a density of 0.35 is medium while others say it is heavy. Note that even though the maximum degree of membership possible is unity, for any given input value the degrees of membership do not have to total unity.

Figure 2 illustrates how fuzzification is performed for a density of 0.35. The result is that we say the density value of 0.35 has a degree of membership of 0.25 in the heavy fuzzy set and a degree of membership of 0.75 in the medium fuzzy set. This is more conveniently written: \( \mu(\text{density}_{\text{heavy}})(0.35) = 0.25 \) and \( \mu(\text{density}_{\text{medium}})(0.35) = 0.75 \).

Though for completeness \( \mu(\text{density}_{\text{light}})(0.35) = 0.0 \) is true, it takes no further part in the fuzzy process.
The input value of a speed of 17mph was used in the fuzzification process on the fuzzy sets and their membership functions of Figure 3 to produce the fuzzy degrees of membership values contained in Table 1.

![Figure 3 Fuzzification of speed value](image)

Table 1 Fuzzified example input values of density and speed

<table>
<thead>
<tr>
<th>Density</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu(\text{density})_{\text{heavy}}(0.35) = 0.25$</td>
<td>$\mu(\text{speed})_{\text{slow}}(17) = 0.3$</td>
</tr>
<tr>
<td>$\mu(\text{density})_{\text{medium}}(0.35) = 0.75$</td>
<td>$\mu(\text{speed})_{\text{medium}}(17) = 0.48$</td>
</tr>
</tbody>
</table>

Clearly the membership functions have a bearing on the fuzzification process. If the lines of the functions are in different positions then different degrees of membership for the same input value would result. Therefore optimising the position and shape of the membership functions is possible. Other membership function geometries can be used and where there is choice there is an opportunity for optimisation. Many FL systems use a membership function based on a curvilinear function. For example a membership function based on a distribution in the normal family could be described by the mean and the variance. Quadratic functions can also be used.

The fuzzification process is carried out for each input variable whenever required by the sampling and control process. The fuzzified values are used in the fuzzy or output inferencing process which is governed by the contents of the rulebase and the inferencing method chosen.

Output variables, the rulebase, fuzzy inferencing and defuzzification

Output variables are, like the input variables, usually constrained by the environment to be controlled. For example the outputs from a traffic signal control system would essentially be redlight and greenlight duration. Output variables are described by fuzzy sets in a similar manner to input variables and therefore the number of fuzzy sets for each output variable and the membership function for each set are optimisable. The fuzzy sets and their membership functions shown in Figure 4 are taken from the...
Optimising fuzzy logic traffic signal control systems

Brubaker and Sheerer system. For example purposes the duration of both the red and green lights has been limited to 20 seconds.

Rules have the form

\[ \text{if antecedent, AND antecedent, \ldots \ antecedent, then consequent, \ldots \ consequent} \]

where the antecedents refer to the fuzzy sets of the input variables and the consequents refer to the fuzzy sets of the output variables.

The rulebase matrix for our example is shown in Table 2.

<table>
<thead>
<tr>
<th>Density x Speed</th>
<th>Slow</th>
<th>Medium</th>
<th>Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>constant_on</td>
<td>constant_on</td>
<td>constant_on</td>
</tr>
<tr>
<td>Medium</td>
<td>medium</td>
<td>long</td>
<td>constant_on</td>
</tr>
<tr>
<td>Heavy</td>
<td>short</td>
<td>medium</td>
<td>long</td>
</tr>
</tbody>
</table>

The resultant rules are given in Table 3 which also shows which rules are fired through the first part of the process known as fuzzy, or output, inferencing. This is done by processing each rule as if it were an ordinary Boolean if statement and by applying the condition of nonzero degree of membership to each of the antecedents (see Table 1 for relevant values).

The next problem is to work out the degree of firing of each of the fired rules. Take Rule 4 which uses the AND operator. This is the fuzzy AND operator that extends the Boolean meaning used in the first part of fuzzy inferencing. By convention the fuzzy AND directs us to take the minimum of all the degree of membership values on the antecedent side of the rule. Hence we apply the fuzzy values for density and speed, \( \mu(\text{density})_{0.35} = 0.75 \) and \( \mu(\text{speed})_{0.17} = 0.3 \) respectively, and therefore take \( \min(0.75, 0.3) \) and say that Rule 4 is fired to degree 0.3. More conveniently it is written as \( \mu_{\text{fired}}(\text{Rule 4}) = 0.3 \). If the OR operator were used in the rules then the convention is to take the maximum of the two values.

Following the same procedure for the other fired rules we have the following:

\[ \mu_{\text{fired}}(\text{Rule 1}) = 0.25, \mu_{\text{fired}}(\text{Rule 2}) = 0.25, \mu_{\text{fired}}(\text{Rule 4}) = 0.3, \mu_{\text{fired}}(\text{Rule 5}) = 0.48 \]
The defuzzification process maps each of these fuzzy values onto the relevant output variable fuzzy set (see Figure 4) and then combines the mappings to produce a crisp value used by the mechanisms in the control system itself.

**Figure 4  Mapping of fuzzy values to output fuzzy sets using the clipping method**

There are several choices to be made in the defuzzification process and these can be grouped into two parts. The first choice determines the shape of the fuzzy set after the mapping process: the choice is between clipping or scaling. Figure 4 shows the clipping method. See Clement (1997c) for a description of other methods, in particular the scaling method. Note that the mapping of the degree of firing of Rule 2 onto the medium fuzzy set is not shown in Figure 4 as its results are subsumed by the mapping of the Rule 4 firing.

The second choice is how to deal with the resultant shapes to arrive at the final crisp output value. Several methods are available such as weighted average, centre of gravity, Yager's method, centre of largest area and the mean of maxima method (Senoner 1997). In addition, if the centre of gravity method is used, the clipped or scaled areas can be unioned or summed.

In this example, the centroids of the clipped regions (2.8, 6.0, 6.0, 13.5 seconds as shown in Figure 4) are applied to the weighted average procedure to produce the resultant green duration:

\[
\text{Duration} = \frac{0.25 \times 2.8 + 0.25 \times 6.0 + 0.3 \times 6.0 + 0.48 \times 13.5}{0.25 + 0.25 + 0.3 + 0.48} = 8.2 \text{s}
\]

The same process is used to find the red duration though a different rulebase is used and hence the red duration time is highly likely to differ from the green duration time.
Considerations for optimising fuzzy logic systems

The rulebase

The number of fuzzy sets associated with each input variable impacts heavily on the structure and size of the rulebase. A simple rulebase consists of rules of the form:

\[ \text{if } (x, \text{ is } A) \text{ AND } (x, \text{ is } B) \text{ then } (y, \text{ is } C) \]

If each of the input variables were described by three fuzzy sets then the rulebase would be complete if populated by nine rules. If each of the inputs were then described by five fuzzy sets the rulebase would comprise 25 rules. Clearly, more input variables would require more rules though it is likely that some of these rules would be redundant in a practical application. See Table 3 where three possible rules (all apply to light traffic density) are covered by Rule 7.

Fuzzy inferencing and defuzzification

Fuzzy logic systems are very forgiving to the designer in that they will generally work reasonably well even though a processing style decision may not be optimal. For example, the weighted average method requires centroids to be taken. If the output fuzzy set membership functions are geometrically isosceles then the degree of firing of the relevant rule is of no significance since the centroid will always lie on the same x-axis value. Under these conditions, the interpretation to be placed on the AND and OR operators of the rulebase is unimportant as the same values will be produced either way. This also applies to whether clipping or scaling is used. In reality, membership functions are commonly determined by a process of trial and error (or sometimes through optimisation) and seldom remain isosceles. But this property is useful for checking the output of an FL system during initial building and testing.

Choice of methods can produce differences in output values and this is illustrated in Table 4 where green and red duration values for the example are calculated for all combinations of two choices for each of the two parts of the defuzzification process.

<table>
<thead>
<tr>
<th>Defuzzification</th>
<th>Duration</th>
<th>Green</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Part 1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>clipping</td>
<td>weighted average</td>
<td>8.2</td>
<td>5.9</td>
</tr>
<tr>
<td></td>
<td>centre of gravity</td>
<td>11.4</td>
<td>8.6</td>
</tr>
<tr>
<td><strong>Part 2</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>clipping</td>
<td>weighted average</td>
<td>8.3</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>centre of gravity</td>
<td>10.8</td>
<td>7.8</td>
</tr>
</tbody>
</table>

Note that the ratio of red time to green time for each of the four combinations shows little variation. /Extend the table to include the ratio values./
Automatic optimisation using genetic algorithms

Initial methods

It is clear that nearly all of the elements of an FL system could be optimised automatically. Assuming that the numbers of input and output variables is fixed then the following elements are possibilities for optimisation:

- the number of fuzzy sets representing the linguistic meaning of each of the input and output variables;
- the rulebase itself;
- the membership function of each of the fuzzy sets of each of the input and output variables;
- the fuzzy inferencing method; and
- the defuzzification technique.

There are many instances in the literature of GAs being used to optimise FL systems (see Cordon, Herrera and Lozano (1996) for a classified review). Most of these refer to optimisations performed on the rulebase, the fuzzy sets and their membership functions simultaneously. Some GAs optimised the membership functions of the input variables only while others optimised the membership functions of both input and output variables. There were no instances found in the literature where optimisation included a choice of fuzzy inferencing method and defuzzification technique: these choices would be easy to code into a GA but would require the FL system to have the methods implemented.

Assuming for the moment that the structure of the rulebase is fixed (ie for each rule the number of antecedents and the number of consequents are constant) and the geometric type of the membership functions is fixed (eg triangular/trapezoid only) then the number of elements of the FL system being optimised with one GA impacts on the length of the chromosome used to represent the FL element parameters. The more elements to optimise the greater the length of the chromosome. This in turn impacts on the effectiveness of the GA operators as well as the processing time usually needed due to the increased GA population requirements (Clement 1997b). Therefore alternatives to single GA optimisation are likely to be necessary.

One possibility is to iteratively tune the FL system for each of its elements in turn. Thus the first optimisation could be for the number of fuzzy sets for each of the input variables, the membership functions (geometry) and the rulebase; the output variables, the fuzzy inferencing method and defuzzification technique having been temporarily selected or designed. After suitable optimisation, the fuzzy sets, membership functions and rulebase could be built from the best chromosome of the GA. The input variables' fuzzy sets and membership functions would then be fixed and GA optimisation would be performed for the output variables' fuzzy sets, the membership functions and the rulebase again. Optimising for the rulebase would be required since a change in the number of fuzzy sets of the output variables impacts on the consequent side of the rules. The input side could then be reoptimised using the optimised output side parameters and...
Optimising fuzzy logic traffic signal control systems

Finally, output side optimisation would be performed. This process could continue until changes to all parameters become minimal or for a set number of optimisation cycles.

Alternative rulebase structures

The assumption was made above that for each rule the number of antecedents and the number of consequents were constant. This need not be the case as some of the input variables of FL systems can be omitted from some rules in some circumstances (Hoffmann and Pfister 1995; Klungboonkrong 1997), hence the number of antecedents can change. The impact of this is again on the chromosome string length which can therefore vary depending on the size of the rulebase and the structure of the rules. This poses problems for the conventional GA reproduction operators since the position of the crossover point must occur between set boundary limits governed by the length of the chromosome.

One way of overcoming this is to use the so-called messy GA technology. With messy GAs, more complex operators are needed to recombine the chromosome string into a sensible structure after the crossover operation. Hoffmann and Pfister (1995) use cut and splice operators which use extra information, contained within the chromosome, about the context of a section of code. Examples of sections of code are rules, fuzzy set names and membership functions.

In addition there are some circumstances when entire rules are redundant and Zizka (1996) reported considerable success by limiting the number of rules in the rulebase.

Objective functions and optimisation techniques

Difficulties of extending objective functions

The initial point made in this paper was that the range of objectives for traffic signals is becoming much broader than in the past. The anticipated range is far wider than the range currently encompassed by the models commonly used in contemporary traffic signal control. Some of the difficulties of including other objectives in the optimisation objective function of an analytical approach are described in Thompson-Clement and Taylor (1996) where an attempt was made to integrate a 'stand-alone' fuel consumption model with the other optimisation objectives. This fuel consumption model was not a derivative of vehicular delays or stops but used the traffic data as direct input to its constituent analytical processes. Similar comments of integration difficulties apply to noise models such as NetNoise (Woolley 1997) which also utilises direct data input.

Choice of optimisation technique

The GA requires a method to assess the fitness of the individuals of its changing (through reproduction) population (the individuals contain the parameter information for
The classification approach

It has become generally accepted that a failing of the signal control systems currently in use (e.g., SCATS, SCOOT) is that they try to impose similar phasing and timing constraints over a wide area due to inflexible optimisation goals. It is recognised that such restraints may be in order for some intersections but not for others in a different area of the network. Therefore, systems of the type suggested by Clement (1997b) and developed by Senoner (1997) may be more suitable for a wide range of sites, as these systems are essentially single intersection controllers. Network optimisation considerations are effected through communication with their neighbouring controllers. These systems have the potential to be much more flexible in two significant ways. First, optimisation objectives can be set for the local environment (not all policy objectives would be required at each intersection) and are easily altered depending on circumstances. Second, these systems are designed to produce very flexible phasing and timing arrangements not easily available to the controllers of today. Though modifications to SCATS and its derivatives have achieved such arrangements as double-phasing. Another approach was taken by Zhou et al. (1997) who used an FL system to classify traffic situations. Operating parameters were then set for each of these classes and control was then left to conventional controllers which recognised particular situations and changed parameter sets accordingly. The method of classifying situations then tuning FL system parameters to each class appears to be the most promising. This...
would depend on finding an efficient method for classification that would include the policy-sensitive objectives discussed earlier ie encompass the traffic and network user approach. It would also depend on developing an effective way of easily tuning the FL systems to each of these classes.

Hierarchical fuzzy logic systems

A further complication arises when one considers the number of inputs needed for traffic signal control. Clearly many inputs are needed and more will be needed when the range of objectives is extended. The major effect of many input variables is that the number of rules increases algebraically with the number of input variables (Hoffmann and Pfister 1995) hence the GA chromosome string required for optimisation would quickly become very long thus increasing the time required for a GA processing run.

Another consideration is that with two input variables it is easy to visualise the fuzzy control surface in three dimensions and a graphical visualisation feature is de rigueur for contemporary FL construction environments. See Clement (1997c) for examples of control surfaces taken from the SiceFuzzy demonstration program.

Figure 5 The Senonor fuzzy logic traffic signal control model showing the hierarchical modular approach taken

One method to counter the large rulebase/big GA chromosome problem and to make manual tuning of the fuzzy system easier is to build a hierarchical FL system (Hoffmann and Pfister 1994; Sayers et al 1996). The modular and hierarchical ideas of the Sayers...
model were extended by the system of Senoner (1997) and the schematic of Figure 5 shows the layering of the complete scheme into several FL systems each of which can be tuned independently. Note the use of intermediate variables used as connectors between the various modules and how one \textit{near} input is fed into RB5, a second layer rulebase.

Essentially each rulebase (eg RB4) of Figure 5 along with its inputs and output is representative of an FL system. The meaning of each input variable is explained fully in Senoner (1997) along with the methods by which the system works. The input variables include the number of cars waiting (\textit{near}), the elapsed time since the last green light (\textit{eltime}), the time remaining before the likely arrival of the upstream platoon of vehicles (\textit{sysitime}), the number of vehicles in the link leading to the downstream intersection (\textit{fncar}), the number of outgoing vehicles for the last five seconds per lane (\textit{outcar}) and the number of remaining vehicles in the lane (\textit{necar}). The Senoner system operates on competition for green time between the traffic streams of an intersection. Examples of the saving in rulebase size are contained in Clement (1997c).

Conclusions

The future direction of signal control is towards including policy-sensitive objectives in the optimisation process and for systems that are flexible to the local environment of each intersection. Fuzzy logic signal control systems appear to be well-suited to the problem when tested under laboratory simulation conditions and using communication between neighbouring controllers as the means for network optimisation. Despite the laboratory successes only one system of those reported has been applied to a practical situation.

Tuning the fuzzy logic systems is usually carried out by trial and error and is often a tedious operation. Genetic algorithm technology is a possibility for automatic optimisation of the setup parameters and methods of fuzzy logic systems. These parameters include the numbers of fuzzy sets for the input and output sides and the membership functions themselves. The choices of methods are for the fuzzy inferencing method and for the defuzzification technique. The size of the rulebase could impact heavily on the performance of the GA optimiser and hence limits on the possible number of fuzzy sets may have to be imposed. There are many instances of genetic algorithm-optimisation of fuzzy logic systems but none of those reviewed in the literature were applied to traffic signal control.

The suggested approach is to classify traffic and network user demand situations so that a different FL setup could be applied to each. Optimisation - which would include the broad range of objectives required of future signal systems - would then aim to find the fuzzy logic parameters and processing choices for each of these classes. Iterative optimisation of the elements of the fuzzy logic system is suggested.
Optimising fuzzy logic traffic signal control systems

The choice of using simulation or analytical techniques depends on the accuracy of the affordable models and the ease with which the chosen option can be implemented as the objective function of a genetic algorithm.

References

Brubaker, DI and Sheerer, C (1992) Fuzzy-logic system solves control problem *EDN* 18 June pp 121-7


Clement, SJ Bell, MGH Cassin, C and Grosso, S (1997a) Experiences with the Path Flow Estimator on a Leicester City street network, pp 455-68 of *Proceedings of the 21st Australasian Transport Research Forum* Adelaide: University of South Australia

Clement, SJ (1997b) Genetic algorithms in dynamic traffic signal timing: the Cabal model Master of Engineering (Civil) thesis, University of South Australia, Adelaide

Clement, SJ (1997c) The feasibility of genetic algorithms and fuzzy logic applied to adaptive traffic signal control Report to the Department of Industry, Science and Tourism, Australia


Dem, R and Jumar, U (1994) An advanced fuzzy controller for traffic lights, pp 67-72 *Proceedings of the IFAC Conference on Artificial Intelligence in Realtime Control*, Spain


Pappis, CP and Mamdani, EH (1977) A fuzzy logic controller for a traffic junction IEEE Transactions on Systems, Man, and Cybernetics Vol SMC-7 (10), pp 707-17, October


Senoner, R (1997) Traffic control system for an intersection group based on fuzzy logic. Thesis prepared in partial fulfilment of the MSc degree Thesis prepared at Transport Operations Research Group, University of Newcastle upon Tyne, UK


Woolley JE (1997) A tool for planning and evaluating the effects of noise from road traffic networks. PhD thesis; School of Geoinformatics, Planning and Building; University of South Australia, Adelaide
