PERGAMON

Engineering Applications of Artificial Intelligence 13 (2000) 311–322

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

Development of an intelligent technique for traffic network incident detection

Dipti Srinivasan[a], Ruey Long Cheu[b,*], Young Peng Poh[a], Albert Kim Chwee Ng[c]

[a]*Department of Electrical Engineering, National University of Singapore, Singapore 119260, Singapore*
[b]*Department of Civil Engineering, National University of Singapore, Singapore 119260, Singapore*
[c]*Navigation Systems Department, CET Technologies Pte Ltd., Singapore 609602, Singapore*

Abstract

Automated incident detection and alternative path planning form important activities within a modern expressway traffic management system which aims to ensure a smooth flow of traffic along expressways. This is done by adopting efficient technologies and processes that can be directly applied for the automated detection of non-recurrent congestion, the formulation of response strategies, and the use of management techniques to suggest alternative routes to the road-users, resulting in significant overall reductions in both congestion and inconvenience to motorists. A delicate balance has to be struck here between the incident detection rate and the false-alarm rate. This paper presents the development of a hybrid artificial intelligence technique for automatically detecting incidents on a traffic network. The overall framework, algorithm development, implementation and evaluation of this hybrid fuzzy-logic genetic-algorithm technique are discussed in the paper. A cascaded framework of 11 fuzzy controllers takes in traffic indices such as occupancy and volume, to detect incidents along an expressway in California. The flexible and robust nature of the developed fuzzy controller allows it to model functions of arbitrary complexity, while at the same time being inherently highly tolerant of imprecise data. The maximizing capabilities of genetic algorithms, on the other hand, enable the fuzzy design parameters to be optimized to achieve optimal performance. The results obtained for the traffic network give a high detection rate of 70.0%, while giving a low false-alarm rate of 0.83%. A comparison between this approach and four other incident-detection algorithms demonstrates the superiority of this approach.

Keywords: Hybrid AI systems; Fuzzy logic; Genetic algorithms; Incident detection; Traffic engineering

1. Introduction

Incident detection has become an important and sophisticated task in today's complex engineering environment, spanning areas ranging from air navigation and traffic networks to power systems and computers. Incidents on urban expressways include accidents, disabled vehicles, spilled loads, maintenance, detector malfunctions and other activities that disrupt normal traffic flow, causing delays to motorists and degraded road safety conditions. For more effective traffic management systems, an automated incident-detection algorithm that is reliable and quick at detecting incidents is essential. Early detection of incidents is vital for the formulation of effective response strategies and the provision of real-time information to motorists.

Research in automated incident-detection techniques started in early 1970s with the implementation of interstate freeway systems. These techniques typically divide the freeway zones into sections of 500–1000 m in length. Inductive loop detectors are placed in individual lanes at the boundary of each section to collect traffic volume, occupancy and average speed. The data

* Corresponding author. Tel.: +65-874-2153; fax: +65-779-1635.
E-mail address: cvecrl@nus.edu.sg (R.L. Cheu).

PII: S0952-1976(00)00011-7

collected from upstream and downstream detector stations of each section is transmitted to a traffic management centre at regular intervals of 30 s for analysis and detection. Fig. 1 shows the traffic flow for a section of an expressway.

1.1. Incident-detection algorithms

The expressway incident-detection algorithms in the past have included techniques such as decision trees for pattern recognition (Payne et al., 1976), time series analysis (Ahmed and Cook, 1982) and Kalman filters (Willsky et al., 1980). The three main incident-detection algorithms are briefly described below.

1.1.1. California algorithm

The California algorithm consists of a family of 11 algorithms that detect incidents based on discrepancies in occupancy values (Payne et al., 1976). Typically, the algorithms use 60-s average occupancy data from two adjacent detector stations along the freeway, on the upstream and downstream sides, to compute absolute, relative and temporal differences in lane occupancy. A binary decision tree structure, which classifies traffic conditions between the two stations into one of several states, provides the thresholds against which the input features are then compared. One of the better-known algorithms, the California algorithm No. 8, uses a 5-min roll-wave suppression logic that helps to reduce false alarms due to shock waves from downstream sectors. The four input features and five threshold values in the algorithm are calibrated with historic incident data before application (Cheu, 1994).

The main disadvantage of California algorithm No. 8 is that it uses only occupancy-related variables as inputs. Volume-related data is never taken into account. The other disadvantages include its input features and a 60-s application interval, which results in a minimum time lag of at least 2 min to detect an incident. The main unique advantage of the California algorithm No. 8 is a low false-alarm rate(FAR).

1.1.2. McMaster algorithm

The McMaster algorithm is based on catastrophe theory (Persaud et al., 1990). Traffic volume and occupancy values from the fast lane of a detector station are used as its inputs. The algorithm operates by comparing field data with station-specific volume-occupancy templates. Specifically, two different volume-occupancy templates, one for detector stations with normal traffic and the other for stations with recurring congestion, are used. If an incident condition is found to exist for three consecutive intervals, an alarm is sounded.

The McMaster algorithm has some distinct advantages in comparison to the California algorithm No. 8. In this algorithm, malfunction of a downstream detector does not affect incident detection, unlike in California algorithm No. 8. It uses volume as an input in identifying possible incidents, unlike California algorithm No. 8, which takes only occupancy inputs into account. The mean time to detect (TTD) an incident is 30 s faster than in California algorithm No. 8. The McMaster algorithm takes recurring congestion into account in identifying incidents, leading possibly to a lower FAR.

The main disadvantage of the McMaster algorithm is that only data from the fast lane is evaluated. A longer incident-detection time is taken for an incident occurring on the shoulder or the right-most travel lane, which is detected only after it affects the traffic flow in the fast lane.

1.1.3. Minnesota algorithm

The Minnesota algorithm (Stephanedes and Chassiakos, 1993) depends on the occupancy values as input, and operates on 30-s intervals. Specifically, a 1-min average occupancy across all lanes, at the upstream and downstream stations, is used as the input. Its detection concept is based on comparing the average spatial occupancy differences before and after an incident.

The Minnesota algorithm has two main disadvantages. First, it uses only occupancy values as input, and ignores volume values. During low-volume conditions, the lack of volume as input may lead to many false alarms. Second, the detection time is extended to 3 min, as the algorithm uses occupancy values from the past six time intervals.

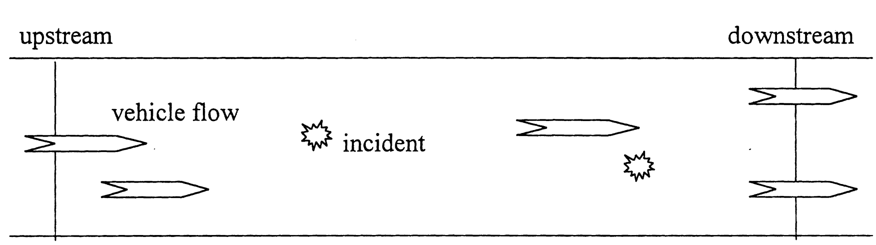

Fig. 1. Pictorial view of traffic flow along an expressway.

1.2. AI-based models

The growing interest in artificial intelligence (AI) techniques has presented new opportunities to provide solutions to the increasing challenges of the future. AI-based techniques such as neural networks, fuzzy logic and genetic algorithms (GAs) (Goldberg, 1989; Haykin, 1992; Bojadziev and Bojadziev, 1995) are highly adaptive and can devise solutions to problems where traditional methods cannot be effectively applied. AI-based techniques have shown great potential in the development of automated incident-detection algorithms (Cheu, 1994) with the promise to give high incident-detection rates and low FARs. Fig. 2 illustrates the automated incident-detection process using AI techniques.

The application of artificial neural networks (ANNs) for freeway incident detection was first investigated in (Cheu et al., 1991). Among the ANNs tested, a multilayer feed-forward network performed well by giving a high DR rate and a low FAR (Cheu and Ritchie, 1995). The inputs to the neural network consisted of 30-s occupancy and volume figures for five consecutive time intervals at the upstream station, as well as the data for three consecutive intervals from the downstream station. The single output neuron was used to classify the traffic condition as 'incident' or 'incident-free'. The neural network was trained on simulated incident data, and tested on simulated as well as real incident data obtained from a freeway in California.

Abdulhai (1996) used probabilistic neural networks (PNNs) for the incident-detection problem. The neural network developed there was tested on real as well as simulated data. The results indicated that PNN developed in (Abdulhai, 1996) had a higher DR and a higher FAR which is undesirable) for the same set of test data used in (Cheu, 1994), and offered relatively high adaptability from one site to another. The main limitation of the PNN was a large memory and computation time required, due to the large size of PNN pattern layer.

1.3. Main objectives

Despite extensive research, many traffic-management agencies would still prefer to have higher DR and lower FARs and incident-detection times, than those that can be offered by existing techniques. This has motivated the authors to develop a hybrid technique, using a genetic algorithm (GA) in a fuzzy system framework, for speedy and accurate incident detection. To the authors' knowledge, it is the first time that such hybrid fuzzy-GA combination has been used for traffic network incident detection.

The main objectives for this automated incident-detection algorithm are as follows:

1. to determine incident occurrence using different traffic indices such as average occupancy and volume;
2. to have a high incident-detection rate;
3. to have a low FAR;
4. to have a low average mean TTD after an actual incident occurrence.

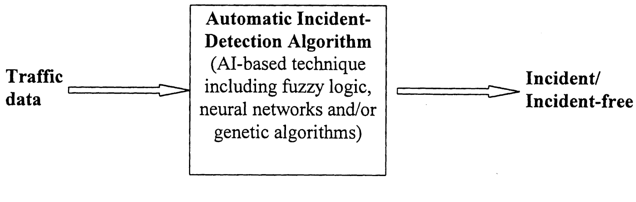

Fig. 2. Traffic data processing for automated incident detection.

2. Fuzzy logic and genetic algorithm

This section provides the relevant theoretical concepts and equations used in the development of the hybrid fuzzy-logic/GA-based approach for network incident detection. AI techniques can be classified into traditional AI, i.e. expert systems and new AI techniques (Bojadziev and Bojadziev, 1995; Haykin, 1992; Goldberg, 1989) that can handle numerical computations such as fuzzy logic, GAs and neural systems.

2.1. Fuzzy logic

Fuzzy logic is a human-inference-oriented AI technique that incorporates the uncertainty and abstract nature inherent in human decision-making into intelligent control systems. It captures the approximate and qualitative boundary conditions of system variables by fuzzy sets with membership functions. A fuzzy system implements functions in near-human terms, i.e. IF-THEN linguistic rules, with reasoning by fuzzy logic (Bojadziev and Bojadziev, 1995).

2.1.1. Operations on fuzzy sets

The difference between classical and fuzzy set theory is that classical theory allows only the crisp, binary values 1 or 0 (true or false), whereas fuzzy logic allows partial set memberships. The extent to which an element x belongs to a fuzzy set A is characterized by its degree of membership, $\mu_A(x)\rightarrow[0,1]$. The three most commonly used fuzzy operators, AND, OR and NOT are defined as follows.

Union of two sets; $A \cup B$, corresponds to the OR operation.

$$\mu_{A\cup B}(x) = \max)\mu_A(x), \mu_B(x)) \quad (1)$$

Intersection of two sets; $A \cap B$, corresponds to AND operation.

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x)) \tag{2}$$

Complement of a set; $\overline{A}$, corresponds to the NOT operation.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{3}$$

2.1.2. Fuzzy control rules

A fuzzy controller consists of a set of control rules. Each rule is a linguistic statement about the control action to be taken for a process condition given by the following rule structure:

IF ⟨condition⟩ THEN ⟨control action⟩.

The ⟨condition⟩ is termed as the antecedent and the ⟨control action⟩ is the consequence. In linguistic approximation by fuzzy logic,each of these terms is represented by a preference fuzzy membership function to establish a value in the interval [0,1].

2.1.3. Fuzzification, rule bases and defuzzification

The fuzzification of a crisp value to a fuzzy terminology is characterized by a scaling factor and a quantization process. After quantization (Bojadziev and Bojadziev, 1995), a degree-of-membership function is applied to derive its membership value, or 'belongingness', in each of the fuzzy linguistic sets. Typically, triangular or Gaussian membership functions like those shown in Fig. 3 are used.

The membership value μ_x of an element x in a fuzzy set is calculated for triangular fuzzy sets as:

$$\mu_x = \begin{cases} 1 & \text{if } x = B \\ (x-A)/(B-A) & \text{if } B > x > A \\ (C-x)/(C-B) & \text{if } B < x < C \\ 0 & \text{if } x \geq C \text{ or } x \leq A \end{cases} \tag{4}$$

and for Gaussian fuzzy sets as:

$$\mu(x) = \exp\left[\frac{-|x-\alpha|^{\beta}}{\sigma}\right], \tag{5}$$

Here α is the center of the Gaussian field corresponding to a membership value of 1, β is a positive number, typically chosen to be 2, and σ is the width of the Gaussian field.

After the input variables have been fuzzified, they are fed into a two-dimensional fuzzy decision table to derive the output variable. Using the look-up table, fuzzy control actions are computed using the min-max functions in a fuzzy control system. The min operation is performed on the antecedents of the rule, followed by the max operation on the consequences to determine the final control actions. The control actions are used in the defuzzification process, where a crisp executable value is computed.

There are four major defuzzification rules; which consist of the maximizer technique, the weighted average method, the centroid method and the Singleton method (Bojadziev and Bojadziev, 1995). The Singleton method represents each fuzzy output set as a single output value by using a weighted average to combine multiple output actions. It treats each output degree-of-membership function as a rectangle, and hence considerably reduces the computation. Once a single output value has been obtained, this is then multiplied by an output-scaling factor to obtain a corresponding crisp, executable control action.

2.1.4. Suitability of fuzzy logic for incident detection

Fuzzy logic is popular for diagnosis and detection kinds of applications because it is conceptually easy to understand, since it is based on natural language. It is tolerant of imprecise data, and is therefore more robust as compared to conventional controllers. It can also model functions of arbitrary complexity, and is very adaptive in nature. The disadvantages of using fuzzy logic are that it may be difficult obtaining good membership function graphs representing the actual control parameters to be modelled. The rule-base for the fuzzy decision table may be difficult to dictate, and

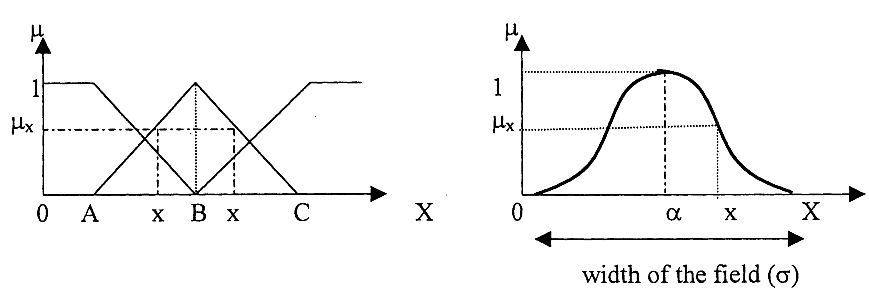

Fig. 3. Example of a triangular and a Gaussian membership function.

experience or an expert's knowledge rules may not always give the optimised solutions.

2.2. Genetic algorithms

GAs are search algorithms based on the mechanics of natural selection and Darwinian survival of the fittest. A GA uses coded strings (chromosomes) of binary numbers (genes) for the search process. Each chromosome is termed as an individual and a population of individuals evolves from generation to generation, with only the most suited individuals likely to survive and generate off-spring, thereby transmitting their genetic material to the next generation. GA are essentially performed by three operators, namely: reproduction, crossover and mutation (Goldberg, 1989).

2.2.1. Reproduction, crossover and mutation

The initial population size in genetic algorithms, typically between 30 and 200, has a substantial effect on the ultimate performance and efficiency in a genetic search. Reproduction occurs when new individuals are produced, whereby a new generation is formed by randomly selecting the fittest individuals from an existing generation, to breed. The selection procedure (Goldberg, 1989) generates a probability that the individuals with higher fitness values will be selected to reproduce within a fixed-size population in each generation, resulting in individuals with higher fitness values in the new generation. Typically roulette-wheel or tournament selection schemes are used.

The crossover operator is used to produce offsprings that are different form their parents, yet inherit portions of their parents' genetic material. A selected chromosome is split into two or more parts (*multiple-point-crossover*) and recombined with another selected chromosome which has also been split at the same crossover point(s) to produce two new offspring which will replace weaker individuals in the population. Crossover operations provide and introduce new search spaces for further testing within the existing hyperplanes into the new population. Mutation in a chromosome is used to provide new genetic materials by randomly selecting bits (genes) to be mutated and subject them to inversion of values. The mutation operator contributes by discovering new or restoring lost genetic materials. Each GA generation is performed by carrying out these three operators until a satisfactory result is found, or the maximum number of generations is reached.

2.2.2. Objective fitness functions and coding schemes

The application of GAs to optimisation problems depends on the choice of the fitness function and coding schemes used to code the design parameters. The fitness function differs for individual problems and depends essentially on the factors to be optimised or minimised (Homaifar and McCormick, 1992). The coding of the parameter set typically involve binary coding, but decimal coding is more efficient and flexible (Ng and Li, 1994), with shorter chromosome lengths and reduced run-times of the GA. The formula for decimal coding is given as:

$$C = C_{\min} + \frac{a_{p-1}b^{p-1} + \cdots + a_0 b^0}{b^p}(C_{\max} - C_{\min}), \qquad (6)$$

where $C \in [C_{\min}, C_{\max}]$ is the decimal value being coded,$[C_{\min}, C_{\max}]$ denotes the decoding range,b is the base value for coding, $a_p \in [0, b-1]$ is an unsigned integer code and p is the number of digits used in the coding which indicates the compromise between accuracy and speed in the evolution process.

2.2.3. Features of GAs

GAs are powerful because they consider a population of points in the search space simultaneously, and permit the optimisation of the whole parameter set.They use objective functions to guide the search and are therefore more robust in comparison to traditional search-and-optimisation techniques in achieving the optimal solution. GAs use probabilistic rules to make decisions, and this has introduced an intellectual capability in them. However, it may be difficult to obtain a suitable objective fitness function for optimal performance if binary coding were to be used. In applications where the variables are real numbers or integers, the importance of a gene is determined by its location on the chromosome. Thus, mutation on one part of the bit string will have a different impact on he variable settings than a mutation on another part of the bit string. This problem, which is unavoidable in these situations and is undesirable, has been termed the 'Hamming Cliff' effect (Homaifar and McCormick, 1992). It may drastically reduce the convergence time of the GA.

3. Design of automated incident-detection algorithm for traffic network

This section deals with the design of the automated incident-detection methodology developed to test incidents along an expressway in California. It also covers some aspects of the computer simulations. Simulated detector data was generated using the INTRAS (Integrated Traffic Simulation) model (Cheu, 1994).

3.1. System overview and traffic data

The automated incident-detection algorithm devel-

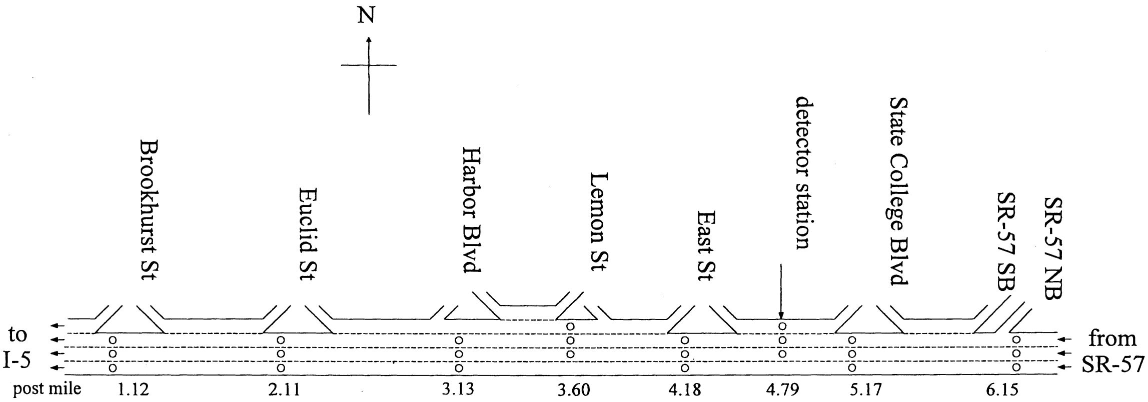

Fig. 4. The study site.

oped here adopts a macroscopic, section method which provides observations at two different, adjacent sites, upstream and downstream, at several segments, along a stretch of expressway with data comparisons between them; it works with aggregated information.

3.2. The study site

An expressway section in the westbound direction of the SR-91 Riverside Freeway in Orange County, California, between the SR-57 and Interstate 5 Freeways, of approximately 5.0 miles in length, was selected as the study site. The entire site has eight detector stations, and the spacing of the detector stations varies from 0.34 to 1.02 miles.

A schematic showing the lane geometry, together with the spacing of the detector stations, is shown in Fig. 4. Lane-specific volume and occupancy values accumulated over 30-s intervals is used as the detector data. The study site was divided into seven segments using the detector stations as boundaries. These segments, numbered from 1 to 7 in the direction of the traffic flow, are shown in Fig. 4.

Segment 0 was the segment upstream of the study site, while segment 8 was that on the downstream side of the study site. Upstream and downstream detector stations, at the upstream and downstream ends, bound every segment. For the ith freeway segment (where $i = 1, \ldots, 7$), segments $i - 1$ and $i + 1$ were referred to as the upstream and downstream segments of segment i respectively.

3.3. Simulation data

The INTRAS model (Cheu, 1994), a microscopic freeway traffic-simulation model written to evaluate different incident-detection algorithms and ramp-control strategies, was used to generate detector data for this study.

Detector outputs from the INTRAS simulation runs were grouped into two independent data sets (i.e., set1.dat and set2.dat). For each data set, the number of incidents in each expressway segment is listed in Table 1.

The traffic indices to be measured in the upstream and downstream portions of the expressway are the average occupancy and volume. For every set of traffic data, there are five occupancy values upstream and three occupancy values downstream, at time-intervals of 30 s each. The same applies for the volume inputs, giving a total of 16 inputs per data set. There are 35,000 data vectors in each of the two files, set1.dat and set2.dat. Besides this, a smaller data file of 3000 data sets was created from set1.dat for training.

The traffic data vectors would be input into the automatic incident-detection algorithm of the fuzzy-GA hybrid, and a fuzzy output obtained for each data vector. For simplicity, the final fuzzy output is converted to give one of two output states, STATE 1 for an incident detected or STATE 0 for the incident-free state.

3.4. Implementation using fuzzy logic

In order for the program to be comprehensive and thorough, the incident-detection algorithm reads in 12 out of the 16 columns of available data set for by considering the current data and two sets of past traffic

Table 1
Number of incidents in each expressway segment in INTRAS simulation runs

Data sets	Segment numbers									
	0	1	2	3	4	5	6	7	8	Total
set1.dat	25	50	50	50	50	50	50	50	25	400
set2.dat	25	50	50	50	50	50	50	50	25	400

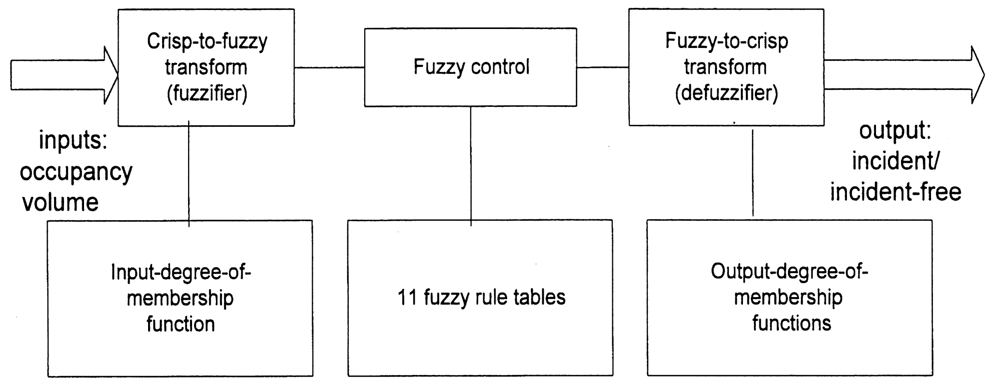

Fig. 5. Block diagram showing the fuzzy process for the incident detection.

data at times (t), ($t-1$), ($t-2$) respectively at an interval of 30 s for a total of 35,000 data vectors in the upstream and downstream portions of each of the eight segments.

The processes of fuzzification, fuzzy control and defuzzification are applied in this algorithm, and eleven fuzzy decision tables are employed in this case. The block diagram for the fuzzy control is shown in Fig. 5.

The algorithm allows the user to perform traffic incident detection using (1) occupancy, (2) volume, and (3) both occupancy and volume. The program basically reads in the set of required input traffic data, applies fuzzy logic to the inputs and evaluates the performance of the fuzzy process. The flowchart of the fuzzy process is shown in Fig. 6. The output of the program indicates whether an incident has been detected (STATE 1: incident) or not (STATE 0: incident-free state).

3.4.1. Multiplexer layer design

The fuzzy algorithm resembles a *cascaded multiplexer design*, and is divided into four different layers, simulating that of a *neural network* with an input layer, two hidden layers and an output layer. The 12 data inputs are first fuzzified and input into the first fuzzy layer. The fuzzified outputs from this layer are subsequently fed into the next layer, and the same continues through to the last layer. The algorithm using fuzzy set theory is shown in Fig. 7. Here 'US' represents upstream and 'DS' represents the downstream side of the expressway segment. The final output is obtained by applying the Singleton method to the 11th fuzzy decision table in the 4th layer.

3.4.2. Fuzzification process

Each fuzzy decision table takes in two inputs (either occupancy or volume); one value is from the upstream portion and the other is from the downstream portion of the expressway at the same time. Each input is fuzzified into one of seven different fuzzy linguistic sets: ZERO (ZO), POSITIVE SMALL-ZERO (PSZ), POSITIVE SMALL (PS), POSITIVE SMALL-MEDIUM

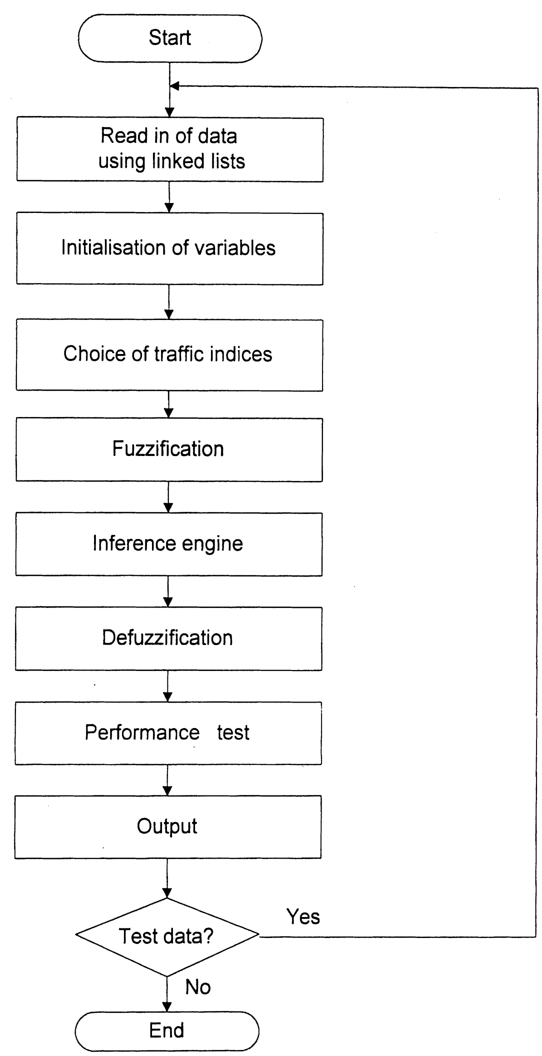

Fig. 6. Flowchart showing the incident-detection algorithm.

(PSM), POSITIVE MEDIUM (PM), POSITIVE MEDIUM-BIG (PMB) and POSITIVE BIG (PB), with their corresponding degree-of-membership values.

This results in a 7×7 fuzzy decision table with 49 possible outputs and degrees-of-membership. As can be seen from the multiplexer design, inputs at three different times, t, $t - 1$ and $t - 2$ are taken for each traffic index. To arrive at the output degree-of-membership, the AND operator (Eq. (1)) is first applied to the two inputs (antecedents) in each fuzzy set to get the minimum value for the consequence. This is then followed by applying the OR operator (Eq. (2)) to get the maximum value among the consequences in each of the seven different fuzzy sets.

For this fuzzy controller, the seven outputs from each of the seven sets from each fuzzy decision table in each layer are then fed accordingly as inputs into the subsequent layer of fuzzy decision tables, and the process is repeated until it reaches the fourth and final layer. At this point, the Singleton method is applied to defuzzify the outputs of the fuzzy sets of the last table to produce a single output control action.

3.4.3. Decision table and membership functions

In each fuzzy decision table, each of the 49 values is to be tuned by the GA to obtain a 'near-optimal' design. A typical decision table for the algorithm is shown in Fig. 8. There are eleven such decision tables in the fuzzy algorithm, arranged in the *layer-like* structure shown in Fig. 7. Triangular and Gaussian membership functions are used in this algorithm.

3.5. Coding scheme using a GA

Fig. 9 shows one chromosome of the genes to be tuned by genetic algorithm. There are 657 unknown fuzzy parameters, of which 539 belong to the 11 decision tables. To simplify the process and let the GA converge, these 539 parameters are manually tuned. The rest of the 118 fuzzy parameters include the centroid values of the fuzzy parameters. Some of the parameters use one GA gene while some use two genes, resulting in a total of 719 genes in all. Non-binary coding is used for this process and since a base 7 decision table is used, the values of the genes vary from 0 to 6.

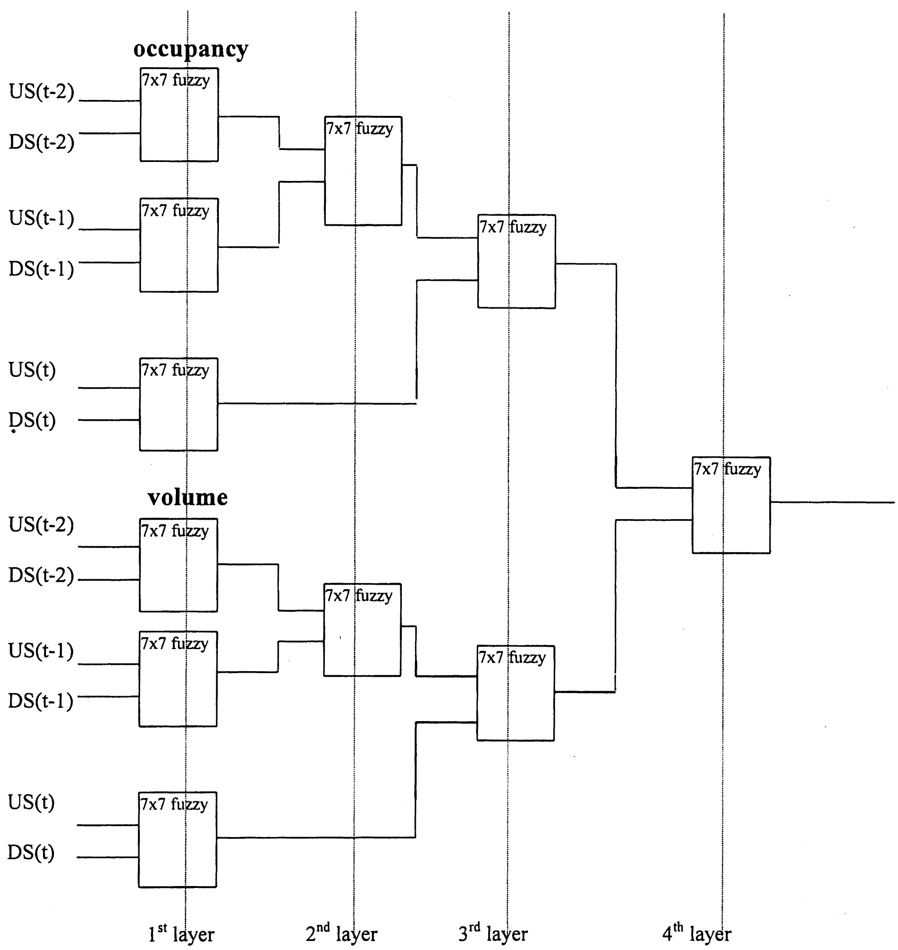

Fig. 7. Block diagram for the fuzzy technique.

		US						
		ZO	PSZ	PS	PSM	PM	PMB	PB
	ZO	ZO	PSZ	PSZ	PM	PMB	PB	PB
	PSZ	ZO	ZO	PSZ	PSZ	PM	PMB	PB
	PS	ZO	ZO	ZO	PSZ	PS	PM	PMB
DS	PSM	ZO	ZO	ZO	ZO	PSZ	PS	PS
	PM	ZO	ZO	ZO	ZO	ZO	PSZ	PSZ
	PMB	ZO	ZO	ZO	ZO	ZO	ZO	PSZ
	PB	ZO	ZO	ZO	ZO	ZO	ZO	ZO

Fig. 8. A typical fuzzy decision table.

1. TetaUS and TetaDS are used for selecting the different membership functions, either triangular or Gaussian membership functions.
2. BetaUS, BetaDS, BL1 and BL2 are used for calculating the shape parameters and scale parameters of the fuzzy Gaussian membership functions.
3. ESZ, EPM, ESM, EPM and EMB are used for calculating the centroid values of the fuzzy Gaussian membership functions during fuzzification.
4. cal, cal1, cal2, cal3 and cal4 are used for calculating the centroid values of the membership functions during defuzzification.
5. ISF is used to change the input scaling factors for the normalized fuzzy values.
6. The rest of the 539 genes are used to tune the values in the fuzzy decision tables.

3.5.1. Training by the GA

In the algorithm, genes for the chromosome are generated by a random-number generator with statistically uniform deviations. These genes are input into a decoding algorithm to obtain the fuzzy parameters, which are fed into the fuzzy-set equations used subsequently to arrive at results for the automated incident-detection algorithm. This process is called 'training by the genetic algorithm' (on-line GA) whereby the algorithm proceeds until a specified number of generations, indicated by the user, are reached, e.g. 100. When the best chromosome is obtained, the genes are stored in another file (fldata.txt) whereby they are then input into another algorithm (off-line GA) to obtain the desired optimal results. The flowchart for the on-line GA for one iteration is shown in Fig. 10.

4. Performance measures

The performance measures evaluate the average DR, the FAR, the false-alarm interval, the current performance and the average time-to-detection of the incident detection.

4.1. Average detection rate, mean time-to-detect and current performance

The incident-detection algorithm is designed such that the triggering of the fuzzy output pulse must come later than the actual output pulse rise, in order an incident block be considered to have been detected (Fig. 11). The delay in detecting the actual incident from the fuzzy output is called the TTD. The longest time to detect is the longest TTD, while the average time to detect is the average TTD.

The average detection rate AvgDR, is defined as:

$$\text{AvgDR} = \frac{100.0 \times (\text{No. of detected blocks})}{\text{Total no. of actual incident blocks}} \tag{7}$$

Another factor to be considered is the current performance rate:

$$\text{Current performance} = \frac{100.0 \times (\text{DR} + \text{DV}\ )}{\text{No. of input} - \text{output pair}} \tag{8}$$

The DR is incremented whenever the detected fuzzy output for an incident corresponds to the actual target output. DV is incremented when the detected fuzzy output for the incident-free condition corresponds to the actual target output.

4.2. False-alarm rate calculation

A false-alarm interval occurs when the fuzzy output is triggered to indicate an incident when the actual target output is zero (no incident). Fig. 12 shows the

<table>
<tr><td>BetaUS
14x1</td><td>BetaDS
14x1</td><td>TetaUS
14x1</td><td>TetaDS
14x1</td><td>BL1
14x2</td><td>BL2
14x2</td><td>ESZ
4x2</td><td>EPS
4x2</td><td>ESM
4x2</td><td>EPM
4x2</td><td>EMB
4x2</td></tr>
</table>
<table>
<tr><td>cal
2x2</td><td>cal1
2x2</td><td>cal2
2x2</td><td>cal3
2x2</td><td>cal4
2x2</td><td>ISF
4x2</td><td>Fuzzy decision table
11 x 49 = 539</td></tr>
</table>

Fig. 9. A typical chromosome used in the algorithm.

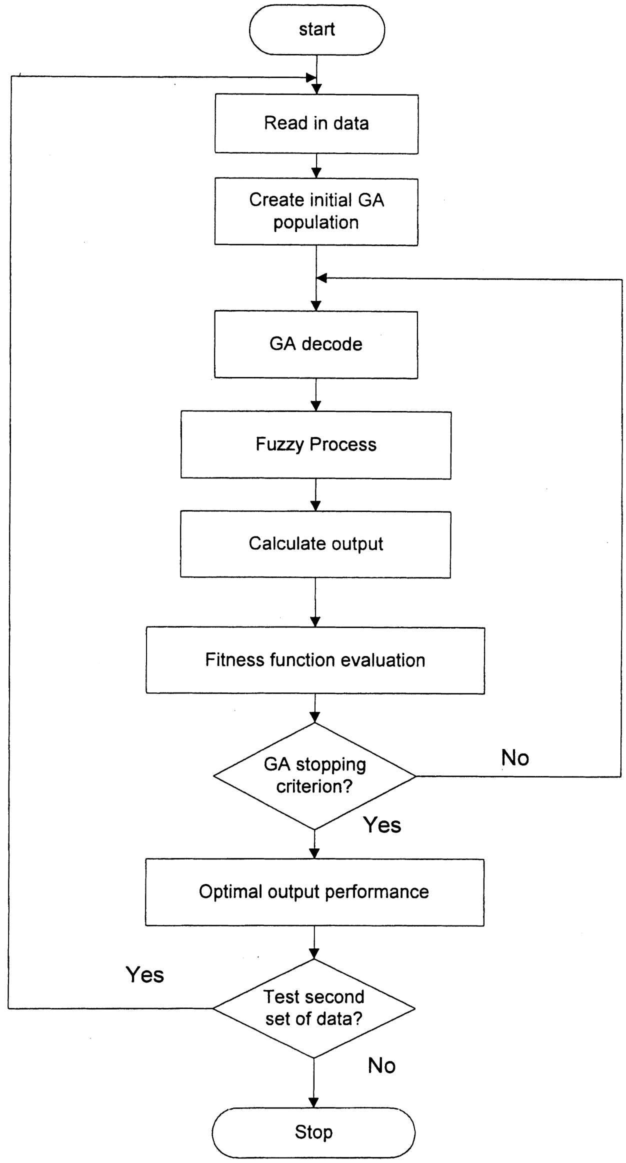

Fig. 10. On-line training by the genetic algorithm.

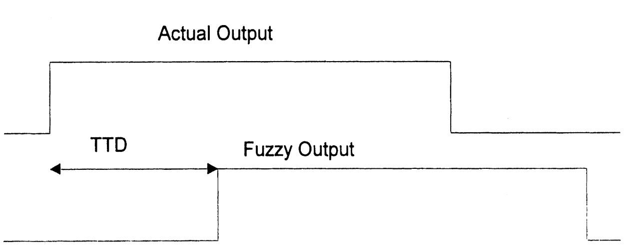

Fig. 11. Block diagram showing the TTD calculation.

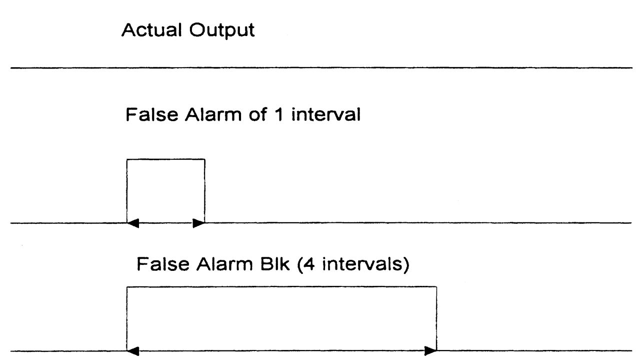

Fig. 12. False-alarm block calculation.

actual output, with no incidence in and two cases of fuzzy output values in the second and third diagrams. A false-alarm block occurs only if one or more false-alarm intervals happen continuously; therefore both cases constitute a false-alarm block. A FAR that is 1% and below is within acceptable limits.

$$\text{PercentFAR\_BK} = \frac{100.0 \times (\text{no. of detected false alarm blocks})}{\text{no. of incident} - \text{free intervals}} \quad (9)$$

The whole incident-detection algorithm for the traffic network is coded using C++ and a user-friendly graphical interface has also been developed.

5. Simulations and results

The simulations for this automated incident-detection algorithm were carried out on a Pentium 166 MHz personal computer using the primary data set. The whole algorithm for the traffic network was coded using C++, and a user-friendly graphical interface has also been developed.

Due to a large number of variables (657) and genes (719) used, each iteration took approximately 10 min. In the cascaded multiplexer design, there are a total of 11 decision tables for the four layers. Each decision table consists of 49 linguistic variables. The variables were tuned by the GA, which used a set of maintenance rules. The Singleton method was used for the defuzzification process, and a crisp output value was obtained for each set of data vectors for performance evaluation. The AvgDRs, FARs and the fitness values are given in Fig. 13 for the two sets of data. On average, for the fault cases considered, a high DR of 70.0% and a low FAR of 0.83% were obtained. These results are found to be very good for traffic network incident detection.

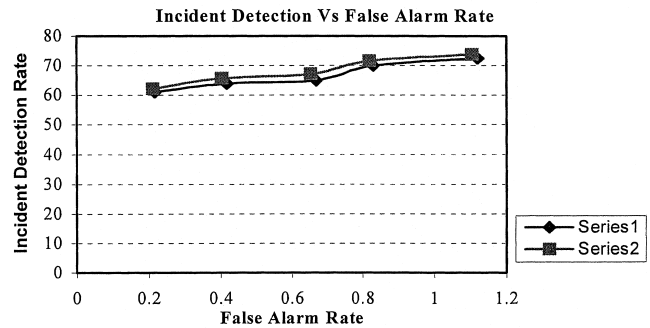

Fig. 13. Graph of incident detection vs. false-alarm rate for the two data sets.

The incident-detection performance of the developed fuzzy-GA algorithm was compared with an artificial neural network approach (Cheu and Ritchie, 1995), and with the California, McMaster and Minnesota algorithms on a similar set of data. Persistence tests (Cheu and Ritchie, 1995) were added to reduce false alarms. With a persistence test of one interval, an incident alarm was issued if the output of the fault-detection algorithm changed from one state to another (e.g., from incident-free to incident) and remained in the changed state for another 30 s. Persistence tests of up to three intervals were used. Table 2 shows the results obtained. In case of California no. 8 and the Minnesota algorithm, several sets of threshold values were obtained and used for calculating the DR, FAR and TTD. The threshold set that gave the lowest FAR is listed in the table.

From the results obtained, it is noted that the fuzzy-GA approach developed here performs generally better

Table 2
Comparison of incident-detection performance

Algorithm	Persistence test	DR (%)	FAR (%)	TD (s)
Fuzzy-GA	0	73.7	1.104	240
	1	71.7	0.818	206
	2	66.6	0.625	210
	3	63.4	0.239	273
ANN (Cheu and Ritchie, 1995)	0	78	1.503	206
	1	65	0.442	252
	2	56	0.218	287
	3	46	0.177	311
California no. 8	NA	49	0.571	255
	NA	7	0.027	431
McMaster	0	37	2.075	260
	1	27	0.612	295
	2	22	0.299	316
	3	17	0.163	346
Minnesota	NA	76	3.048	229
	NA	32	0.197	320

than the ANN approach for persistence tests 1–3 by giving a better DR and smaller TTD. However, the FAR is generally higher than for the ANN. The results also indicate that both these AI-based approaches are superior compared to non-AI algorithms.

6. Conclusions

In this paper, a hybrid fuzzy-GA framework for automated incident detection on expressways has been presented. The algorithm uses a cascading framework of 11 fuzzy decision tables, arranged into four layers in a multiplexer-like design. The inputs to this incident-detection algorithm are the average occupancy and average volume for 30 second intervals at the upstream and downstream portions of an expressway in California. A GA was used for optimising the 657 unknown fuzzy variables. The algorithm was tested on a simulation data set containing 35,000 data vectors obtained from INTRAS. The best chromosome yields a high AvgDR of 70.0%, and a low FAR of 0.83%. This algorithm was found to give superior performance compared to three non-AI algorithms for a similar set of data. In comparison to an artificial neural network, this algorithm used a smaller time to detect an incident, and generally gave a higher DR.

The simulation results have shown that the algorithm using a combination of fuzzy logic and GAs gives a high DR and a low alarm rate, and is very promising for expressway incident detection. The authors believe that the performance of this algorithm could be further improved by using fuzzy maintenance rules to converge the values of the fuzzy decision variables. Future work on this algorithm will involve the automatic tuning of the decision tables to give the algorithm more adaptability and flexibility, and the ability to learn intelligently.

References

Abdulhai, B., 1996. A neuro-genetic-based universally transferable freeway incident detection framework. Ph.D. Dissertation, University of California, Irvine.

Ahmed, S.R., Cook, A.R., 1982. Application of time-series analysis techniques to freeway incident detection. Transportation Research Record 841, 19–21.

Bojadziev, G., Bojadiev, M., 1995. Fuzzy Sets, Fuzzy Logic, Applications, 2nd ed. McGraw-Hill, New York, pp. 23–65.

Cheu, R.L., 1994. Neural network models for automated detection of lane-blocking incidents on freeways. Ph.D. Dissertation, University of California, Irvine.

Cheu, R.L., Ritchie, S.G., 1995. Automated detection of lane-blocking freeway incidents using artificial neural networks. Transportation Research 3C (6), 371–388.

Cheu, R.L., Ritchie, S.G., Recker, W.W., Bavarian, B. 1991. Investigations of a neural network model for freeway incident detection. In: Topping, B.H.V. (Ed.), Artificial Intelligence and Civil and Structural Engineering. Civil-Comp Press, Edinburgh, UK, pp. 267–274.

Goldberg, D.E., 1989. Genetic Algorithms in Search Optimisation, and Machine Learning. Addison-Wesley, Reading, MA.

Haykin, S., 1992. Neural Networks: A Comprehensive Foundation. Prentice Hall, Englewood Cliffs, NJ.

Homaifar, A., McCormick, E., 1992. Full design of fuzzy controllers using genetic algorithms. Proceedings of SPIE Conference On Neural and Stochastic Methods in Image and Signal Processing 1766, 393–404.

Ng, K.C., Li, Y., 1994. Design of sophisticated fuzzy logic controllers using genetic Algorithms. Proceedings of 3rd IEEE International Conference on Fuzzy Systems 3, 1708–1712.

Payne, H.J., Helfenbein, E.D., Knobel, H.C., 1976. Development and testing of incident detection algorithms, vol. 2: research methodology and results. Report No. FHWA-RD-76-20, Federal Highway Administration.

Persaud, B.N., Hall, F.L., Hall, L.M., 1990. Congestion identification aspects of the McMaster incident detection algorithm. Transportation Research Record 1287, 167–175.

Stephanedes, Y.J., Chassiakos, A.P., 1993. Application of filtering techniques for incident detection. Journal of Transportation Engineering 119 (1), 13–26.

Willsky, S., Chow, E.Y., Gershwin, S.B., Greene, C.S., Houpt, P.K., Kurkjian, A.L., 1980. Dynamic model-based techniques for the detection of incidents on freeways. IEEE Transactions on Automatic Control 25 (3), 347–360.