# Application of Genetic Programming to Motorway Traffic Modelling

**Daniel Howard and Simon C. Roberts**
Software Evolution Centre
Building U50, QinetiQ
Malvern, Worcs. WR14 3PS
United Kingdom
dhoward@qinetiq.com
Phone: +44-1684-894480

## Abstract

This paper describes two innovative projects that the investigators are undertaking for the Highways Agency network operator. Both projects use genetic programming to develop traffic management systems. The first project addresses the reliable prediction of motorway journey times for high-flow low-speed conditions, i.e. congested periods. The second project concerns the detection of motorway incidents in low-flow high-speed conditions, i.e. late at night. Genetic programming manipulates traffic readings from the Motorway Incident Detection and Automatic Signaling system to arrive at solutions to both real world problems.

## 1 Introduction

The UK Highways Agency (HA) has invested heavily in the Motorway Incident Detection and Automatic Signaling (MIDAS) system primarily for the purpose of queue protection. Its secondary function is to provide, at zero cost, one-minute averaged traffic data from sites spaced at 500m intervals. On the M25 motorway which encircles London, MIDAS produces readings of traffic velocity, flow, occupancy, headway and flow categorized by vehicle length. These quantities are measured directly by loop sensors which are incorporated into the road surface in each motorway lane.

Ideally, a traffic system would be controlled by obtaining a comprehensive understanding of its dynamics. For example, by modelling driver behaviour characteristics in different traffic states, and by using number plate recognition to monitor lane changing. However, a detailed understanding requires an extensive development cycle which is incompatible with the HA's demand for short delivery times. Rather than seek a thorough understanding of the traffic system, computational techniques are being investigated to develop control functions that learn and operate on the available MIDAS data, e.g. artificial neural networks and genetic programming (GP).

## 2 Journey Time Prediction

The HA needs to produce highly accurate short-term journey time predictions to: allow tactical control systems such as ramp metering to be pro-actively deployed; give motorists advance warning of when and where a tactical control system will be operational on the network; improve the performance of real-time route assignment models; and signal predicted journey times to motorists.

This GP investigation aimed to discover mathematical and logical relationships between predicted journey times and recent MIDAS measurements. For this purpose the journey time, JT, was defined as the average time taken by a driver already on the motorway to cover the distance between two junctions, e.g. junction A merging and junction B diverging.

### 2.1 Forecasting options

Two alternative strategies for journey time prediction were investigated:

- the direct prediction of journey time.
- the prediction of velocity at each site along the journey. The journey time was then computed by integrating these velocity predictions along the path of the virtual journey into the future.

More GP evolution work was involved in the latter strategy because velocity predictions were carried out at each site of the virtual journey. In contrast, the former strategy makes one forecast for the entire journey.

The latter strategy required a forecasting period to be chosen which was commensurate with the maximal expected JT. This allowed the predicted velocity at the terminating site to always be available and thus the time of the complete journey to be calculated. However, predictions were less accurate when the forecasting period was increased. Hence, a balance was achieved by setting the period to 15 minutes and by using a *naive* prediction (defined in Section 2.4) when JT exceeded this period.

Both strategies processed present and past traffic quantities for current and *downstream* sites. It was crucial to use downstream information to anticipate the future traffic state at the current site. This was especially important during high-occupancy periods because traffic queues building downstream tended to produce congestion waves which propagated upstream.

### 2.2 Transforming MIDAS data into a GP terminal set

MIDAS records minute-averaged traffic quantities $\phi_t^{sl}$ for site $s$ on lane $l$ at minute $t$:

$$\phi_t^{sl} = (V_t^{sl}, O_t^{sl}, F_t^{sl})$$

where $V_t^{sl}$ is velocity in km/h, $O_t^{sl}$ is the percentage lane occupancy and $F_t^{sl}$ is the flow rate in vehicles per minute. A correction algorithm was devised to interpolate missing MIDAS data. This was invoked when measurements were simply absent and when no vehicles happened to pass a MIDAS sensor for at least one minute. In the latter case, the values $F_t^{sl} = 0$ and $O_t^{sl} = 0$ were valid, but $V_t^{sl}$ must be estimated to avoid discontinuities with the neighbouring measurements (in time and location).

Lanes are numbered from the offside fast lane[1]. The prediction inputs combined quantities across the offside and the two adjacent lanes as follows. The velocities at each site were averaged across the lanes and weighted by the flow in each lane:

$$V_t^s = \frac{F_t^{s1}V_t^{s1} + F_t^{s2}V_t^{s2} + F_t^{s3}V_t^{s3}}{F_t^{s1} + F_t^{s2} + F_t^{s3}}$$

The input flow was the total across the lanes and the input occupancy was averaged:

$$F_t^s = \sum_{l=1}^{3} F_t^{sl} \qquad O_t^s = \frac{1}{3}\sum_{l=1}^{3} O_t^{sl}$$

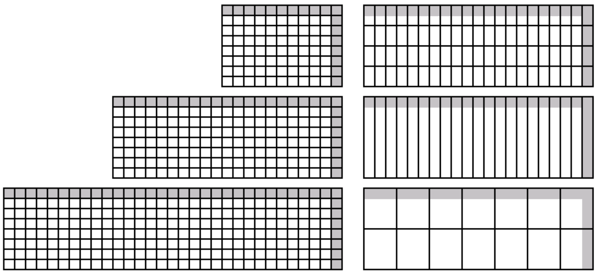

Figure 1: Examples of raw and box-averaged input windows. Time is shown horizontally and site is shown vertically. The grey cells are at the current time or the current site. Each square represents either a raw value or a box-averaged value.

The lane-independent traffic quantities were transformed into a GP terminal set by defining a window comprising previous minutes and downstream sites. At minute $t$ and site $s$, the window spanned back to minute $t-T$ and down to site $s+S$. Journey time predictors simultaneously processed all sites in the journey, i.e. $S = 27$, and typically used $T = 15$. Velocity predictors typically used $S = 7$ and a wide variety of $T$ values were investigated.

Figure 1 illustrates the different types of input windows. Clearly, when $T$ and $S$ were increased the size of the GP terminal set could become excessive and result in an over-complicated search space. Thus data reduction techniques were investigated. The most effective was found by dividing the input window into boxes of size $b_t \times b_s$, and averaging the raw values in each box to give a single value per box for each traffic quantity.

The most recent data was found to possess a stronger predictive power, as expected, but time windows which extended tens of minutes into the past were also found to be beneficial. To strike a balance between these extremes, overlapping boxes which extended further into the past were investigated as shown in Figure 2. Each box reached back from minute $t$ to $t - b_t$ where $b_t$ was initialized to 1 and then increased for each subsequent box. This produced superimposed multiscale time windows which gave a crisp short-term view but evermore blurred longer-term views. This box-averaging technique produced among the best predictors.

Data reduction was also investigated by fusing quantities based on flow-velocity plot descriptors, namely the flow-velocity gradient $G$ or the flow-velocity magnitude $M$.

Some experiments also investigated feeding previous velocity predictions, $\hat{V}$, back in as input data. Fig-

[1] Sites between diverging and merging junctions have three lanes while all other sites have four lanes. The fourth lane is used to enter and exit the motorway.

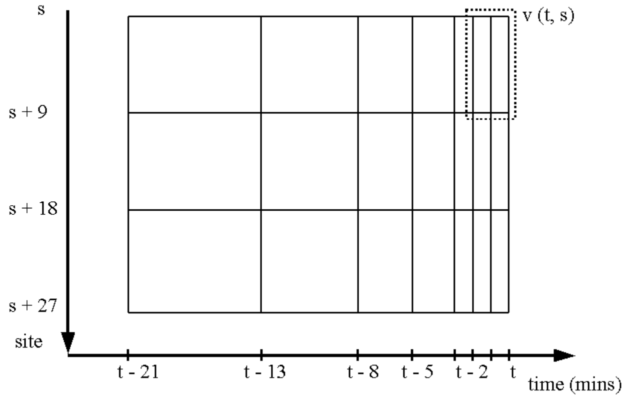

Figure 2: Multiscale input windows superimposed in the time dimension. The window size $b_t$ increased based on the Fibonacci sequence.

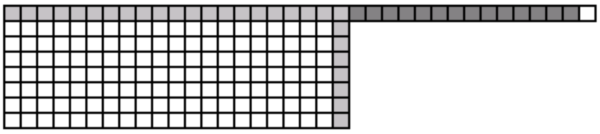

Figure 3: Velocity prediction $\hat{V}^s_{t+15}$ which uses the 14 former predictions of velocity, shown as dark grey cells.

ure 3 illustrates this special case where $(\hat{V}^s_{t+1}...\hat{V}^s_{t+14})$ was used in a form of autoregressive prediction. Journey time predictors could also exploit the recent known JT values, e.g. the JT of the most recent journey completed by time $t$. The experiments investigated which data types gave the most predictive power from:

$$\phi = \left(V, O, F, M = (F^2 + V^2), G = \frac{60F}{V}, \hat{V}\right)$$

Two years of historical MIDAS data was inspected on the counter-clockwise carriageway of the M25 between junctions 15 and 11. This identified days when the sensors were fully operational and the traffic involved congestion waves. Figure 4 illustrates the MIDAS data showing different traffic quantities on different lanes. The sub-image for each given quantity and lane shows progressive time as left to right and progressive sites as top to bottom. Brighter cells represent lower quantity values, e.g. congestion appears as low speed and high occupancy. Journey times between junctions 15 and 11 take between 6 minutes to over 25 minutes. For the experiments reported in this study, GP processed the period 1300h to 2200h.

### 2.3 Experimental GP study

Many GP runs explored different evolution options: a velocity predictor; a velocity-gradient predictor; a journey time predictor. The first two options were driven by velocity errors or JT errors via the fitness function. The third option could only be driven by JT error.

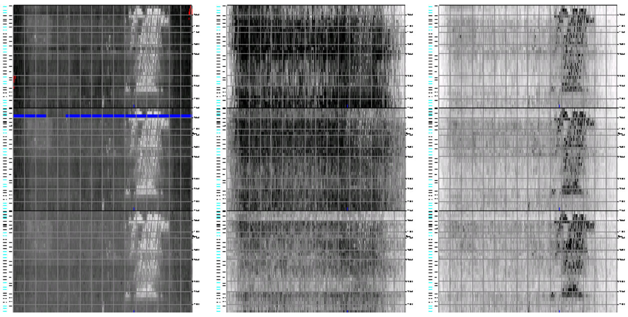

Figure 4: MIDAS data: 0600h to 2200h on 2nd September 1999, M25 counter-clockwise carriageway, junctions 15 to 11. From left to right: velocity, flow and occupancy. From top to bottom: offside (fast) lane, offside-1 lane and offside-2 lane.

Each option involved at least 15 independent GP runs continued for 50 generations. They used the steady-state GP method with a typical population size of 2000 individuals, 90% cross-over, 10% mutation, a breed tournament size of 4, a kill tournament size of 2 and the function set: $+$, $-$, $*$, protected division, min, max and *if-less-then-else*.

Both root mean square error, $\|\cdot\|$, and max error, $\max(\cdot)$, were applied to measure absolute error, $e_a$, and relative error, $e_r$. However, the experiments investigated various fusions of $\|\cdot\|$ and $\max(\cdot)$ errors of $\phi$, the quantity whose error was to be minimized[2]. The errors were calculated as follows, where $\hat{\phi}$ was the evolved prediction and $e_n$ was the error relative to a naive prediction, $\phi_N$:

$$\begin{aligned} e_a &= |\hat{\phi} - \phi| \qquad e_r = \frac{e_a}{\phi} \\ e_n &= |\hat{\phi} - \phi_N| \qquad \text{e.g. } \|e_a\| = \sqrt{\frac{1}{n}\sum e_a^2} \end{aligned}$$

Various fitness measures, $f$, were investigated:

$$f = -\Big(\|e_a\|\|e_r\| + \max(e_a)\max(e_r)\Big) \ ;$$

$$f = -\Big(\|e_a e_r\| + \max(e_a e_r)\Big) \ ;$$

$$\text{or by defining } e_c \text{ as: if } \left(\frac{e_a}{e_n}\right) > 1$$

$$\text{then } e_c = \left(\frac{e_a}{e_n}\right)^2 \text{ else } e_c = \left(\frac{e_a}{e_n}\right)$$

[2] $V^s_t$ or $\Delta V^s_t$ or $J_t$.

$$f = -\|e_c\| \; ; \; f = -\left(\|e_c\| + \frac{\max(e_c)}{100}\right)$$

Predictors were compared based on JT prediction errors, regardless of whether the evolution was driven by JT or by velocity errors. Case-wise JT prediction errors were obtained for the 10 fittest predictors from each GP run. The following four error calculations were then used to quantify the overall JT errors: $\max(e_a)$, $\max(e_r)$, $\|e_a\|$ and $\|e_r\|$. This posed a multi-objective optimization problem in four dimensions and thus a *Pareto* ranking scheme was used to compare the different predictors. Pareto ranking was first performed on the $\max(e_r)$ and $\|e_r\|$ dimensions. A second independent Pareto ranking process was then performed on the $\max(e_a)$ and $\|e_a\|$ dimensions. Predictors were then ordered according to $e_r$ rank and then $e_a$ rank. The best predictor for the experiment was then subjectively designated from the top ranking predictors, and the associated JT errors were tabulated.

### 2.4 Naive predictions

Computational solutions find it challenging to perform markedly better than naive predictions. For example, it is difficult to beat the rule "the journey time in the future will be the same as the time of the journey just completed". A method to overcome this problem is to target the predictors at specific traffic states, and this method is currently under investigation.

Thus, five naive formulae were used to judge the performance of the GP evolved predictors. The simplest naive formula was $\hat{\phi}_{t+p} = \phi_t$ where $p$ is the prediction period. This formula produced the best naive predictions and hence the presented naive results pertain only to this formula.

$\phi_t$ was either velocity at the current site or JT of the most recent journey *completed* by time $t$. For example, let the virtual journey which commenced 9 minutes ago take $JT_{t-9}$ minutes to complete, and that which commenced 8 minutes ago take $JT_{t-8}$ minutes to complete. If $JT_{t-9} = 8.5$ and $JT_{t-8} = 8.1$ then $\phi_t$ was set to $JT_{t-9}$. However, if instead $JT_{t-8} = 7.9$ then $\phi_t$ was set to $JT_{t-8}$.

The prediction period, $p$, was 15 minutes for velocity predictors as discussed in Section 2.1. JT predictors were required to predict the time of the journey *starting* on the next minute and thus $p = 1$.

Table 1: JT errors for naive and evolved predictors of velocity and journey time. Absolute errors are in seconds.

| **Scheme** | $\|e_a\|$ | $\|e_r\|$ | $\max(e_a)$ | $\max(e_r)$ |
|---|---|---|---|---|
| naive $V_{t+15}$ | 99.4 | 0.121 | 343.3 | 0.499 |
| evolved $V_{t+15}$ | 78.7 | 0.099 | 256.3 | 0.35 |
| naive $JT_{t+1}$ | 110.6 | 0.122 | 352 | 0.463 |
| evolved $JT_{t+1}$ | 65.5 | 0.074 | 250.7 | 0.25 |

### 2.5 Results summary

More than 100 sets of experiments were run under different options. Each run processed traffic from 4 weekdays in August 1998. Table 1 shows the JT errors for the best prediction schemes, where $V_{t+15}$ is the prediction of velocity in 15 minutes time, and $JT_{t+1}$ is the direct prediction of the time of the journey starting on the next minute. It can be seen that the evolved predictors consistently gave better accuracy than the naive predictors, for both the prediction of velocity and JT. Furthermore, the evolved JT predictor gave the lowest error for each error type, despite the naive JT predictor being marginally worse than the naive velocity predictor.

The following general recommendations were drawn from the overall comparative study.

Predictor type: Predicting $JT_{t+1}$ gave lower JT errors than predicting $V_{t+15}$. JT predictors were also much faster to evolve and apply than velocity predictors because velocity must be predicted at each site in the journey. However, JT predictors may not generalize across journeys traversing a different number of sites.

Data type: All traffic quantities ($V$, $O$ and $F$) were required whereas the fused quantities ($M$ and $G$) gave greater errors. Feeding intermediate predictions, $\hat{V}$, back into the input tended to reduce prediction accuracy, probably because this "autoregression" complicated the search space by increasing the size of the terminal set. Box averaging proved to be the most beneficial data reduction method.

Error drivers: Recall that various prediction errors drove the evolution via the fitness function. JT errors were better drivers than velocity errors. Both absolute and relative errors were needed and $\|\cdot\|$ errors had to be weighted stronger than $\max(\cdot)$ errors.

Table 2: JT errors for naive and evolved $JT_{t+1}$ predictors on training and test data. Absolute errors are in seconds.

| **Training:** 4853 cases over 15 weekdays | | | | |
|---|---|---|---|---|
| $b_s$ | $\lVert e_a \rVert$ | $\lVert e_r \rVert$ | $\max(e_a)$ | $\max(e_r)$ |
| (naive) | 68.4 | 0.086 | 358 | 0.503 |
| 9 | 45.5 | 0.057 | 213.2 | 0.229 |
| 3 | 51.3 | 0.066 | 228.2 | 0.271 |
| **Testing:** 2008 cases over 5 weekdays | | | | |
| $b_s$ | $\lVert e_a \rVert$ | $\lVert e_r \rVert$ | $\max(e_a)$ | $\max(e_r)$ |
| (naive) | 65.1 | 0.089 | 269.4 | 0.546 |
| 9 | 45.9 | 0.06 | 217.9 | 0.269 |
| 3 | 52.3 | 0.072 | 229.3 | 0.398 |

The $JT_{t+1}$ predictor was scaled-up by using more data from August 1998 and the results are shown in Table 2. Box averaged inputs were used with various $b_s$ (Section 2.2) and $b_s = 9$ proved to be the optimum. The naive predictions are indicated by the $b_s$ column in the table. It can be seen that the evolved predictor consistently gave better accuracy than the naive predictor. Furthermore, comparison of the results on training and testing shows that the evolved predictor generalized well.

## 3 Incident Detection

The HA's MIDAS system uses the HIOCC algorithm (Collins et al., 1979) to detect incidents. The HIOCC (high occupancy) algorithm is essentially a queue detection algorithm. It is very capable of detecting incidents during the peak and interpeak periods, but it gives poor results at night because the traffic flows are too low to allow queues to form. HIOCC or California (Payne et al., 1975) algorithms cannot detect late-night incidents because of the low-occupancy traffic states.

The Staged GP (Howard and Roberts, 1999) method had proved successful in the detection of objects of large variability in poorly specified domains. For this reason, Staged GP was selected to detect subtle anomalies in the late-night MIDAS data, in order to warn for the presence of poorly specified incidents. Success would be much cheaper than alternative approaches such as the capital cost required for upgrades to the traffic monitoring infrastructure.

The sought incident detector should detect the incident as early as possible to allow the human MIDAS operator to take action. For example, the operator could react by positioning a motorway camera to examine the scene of the incident, or by alerting the police who need a maximal response time. Late-night incidents have many causes, e.g. driver fatigue, roadworks, bad weather or animals on the motorway, resulting in a high variability in the characteristics of the incident onsets.

GP was used in a supervised learning role. Hence, it required a "truth" from which to learn to detect incident onsets. Unfortunately, until now very few incidents had been archived, and even where they had their reasons were not always correctly recorded. This investigation manually scrutinized the MIDAS data and marked up an incident "truth". The data was analyzed by inspecting the velocity in all lanes and identifying the approximate time and site of each incident onset. Incidents were then graded by a subjective measure of their significance. Grade 1 incidents were the most obvious type, corresponding to serious accidents which resulted in sustained lane closures. Grade 5 incidents were the most subtle type and appeared only as blips in the viewed MIDAS data. Major incidents were rarer than minor incidents and thus the significance grading was incorporated into the GP fitness measure, to counteract the bias towards detecting simply the most common types of incident (Howard and Roberts, 1999).

The detection task was tackled by a two-stage evolution strategy (Howard and Roberts, 1999). The first stage evolved for itself which traffic data best characterized an incident onset, by distinguishing data proximate to incidents from a sample of non-incident data. The second stage was then required to minimize the false alarm rate whilst retaining at least a single detection point per incident.

### 3.1 Traffic data input to GP

Currently we are evolving an incident detector with a training, validation and test set that we have painstakingly derived from three years worth of MIDAS data. However, in this paper we report on a smaller earlier study. Traffic data was taken from 60 consecutive sites along the M25 between sites 4727 and 5010. Both carriageways were used for 7 nights in August 1998, which comprised 32 incidents of grade 1 to 4. Grade 5 incidents were ignored.

### 3.2 Fitness measure

A GP chromosome was said to have detected an incident when it output a positive value. In other words, the traffic data being processed at the given time and site was deemed to represent an incident. If an actual incident had occurred then the output was called a *true positive*, otherwise it was called a *false positive* or a false alarm. After a chromosome had processed all the training cases, let TP denote the number of resulting true positives and FP denote the number of false positives.

The task was to distinguish incidents of grade 1 to 4 from non-incidents. Furthermore, lower numbered incident grades were more important than higher numbered grades. These issues were captured by the following fitness measure which drove the evolution runs,

$$\text{fitness} = \frac{TPS}{(TPS_{max} + \beta FP)} \qquad (1)$$

where $TPS$ was the score for the incidents detected and $TPS_{max}$ was the score obtained when all incidents were detected. $TPS$ was calculated as follows to weight the incidents according to grade.

$$TPS = \sum_{i=0}^{i<TP} 5 - \text{grade of incident}_i \qquad (2)$$

The variable $\beta$ was used to balance the importance of incident detection against the expense of detecting false alarms. For example, as $\beta$ tended to zero the number of false alarms became irrelevant.

The *figure of merit* (FOM) was used to compare different GP runs. The FOM is the same as the fitness when $\beta = 1$. Note that the minimum FOM is 0 for the case when no incidents are detected or when the number of false alarms approaches infinity, and the maximum FOM is 1 when all incidents are detected with no false alarms.

### 3.3 Evolving first stage detectors

Even though incidents happen at a single time and location, the incidents generally manifest themselves in the traffic data over multiple minutes and a number of sites (more upstream sites than downstream). Consequently GP was trained to detect the onset of each incident by sweeping from 5 minutes before to 5 minutes after the incident time, and by sweeping from 4 upstream sites to 4 downstream sites for each minute. However, an incident was said to be detected if a GP chromosome returned a positive output for any one of these times and locations. Therefore, TP represented the number of incidents detected and had a maximum value of 32. Non-incident training data was sampled at 10 minute steps using all possible sites at each minute, giving a total of 21,181 non-incident training cases.

The GP terminal set was taken from an input window similar to those illustrated in Figure 1. However, the window encompassed only local traffic data by using $T \leq 8$ previous minutes and one downstream site, i.e. $S = 1$. Furthermore, the input data consisted of the *lane-specific* MIDAS quantities: velocity, flow, occupancy, headway and flow categorized by vehicle length.

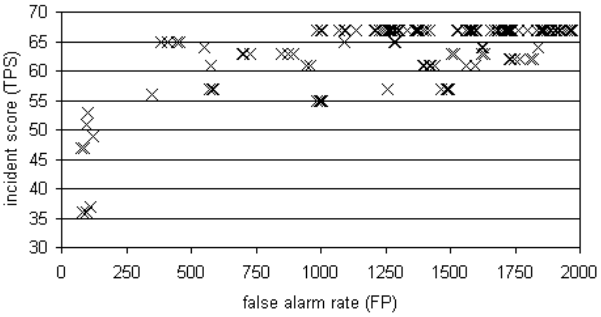

Figure 5: Incident score against false alarm rate for the first stage detectors.

Approximately 60 GP runs were conducted for the first stage each using $T = 3$. A first stage detector's task was to detect *all* incident onsets whilst producing a minimal false alarm rate. The fitness variable $\beta$ was thus set to low values and the range 0.01 to 0.1 tended to give the best results. Higher settings caused incidents to be missed whilst lower settings tended to give an excessive false alarm rate. The population size was set to 1000 and the other GP parameters were the same as those listed in Section 2.3.

### 3.4 Validating first stage detectors

The evolved first stage detectors were validated on the 32 incidents and a maximum score of 67 was achieved if all incidents were detected (i.e. $TPS_{max} = 67$). The detectors processed all non-incident traffic data in validation which totalled to 200,010 cases. Figure 5 plots the incident score against false alarm rate for the 10 fittest detectors from each GP run.

The lowest false alarm rate when all incidents were detected was 987 out of the 200,010 non-incident cases. However, another first stage detector was judged to be the best because it produced 142 hits distributed across 30 incidents and it gave only 382 false alarms, and sacrificed only two grade 4 incidents in order to achieve this (i.e. $TP = 30$ and $TPS = 65$). Even by careful manual inspection it was difficult to determine whether these grade 4 incidents were actually true.

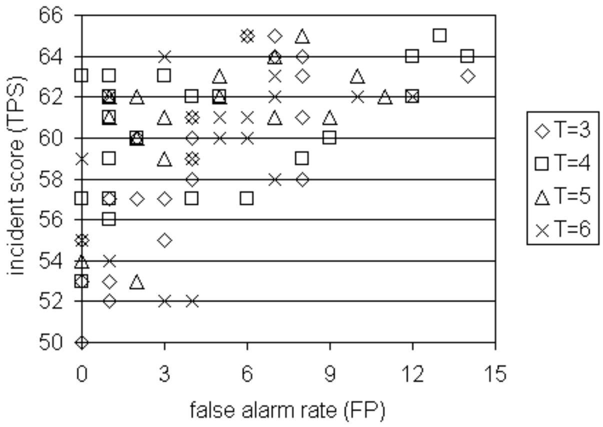

Figure 6: Incident score against false alarm rate for the second stage detectors.

### 3.5 Evolving second stage detectors

Second stage detectors were trained to reduce the false alarms whilst retaining at least a single hit per incident. Note that $TPS_{max}$ reduced to 65 because of the two grade 4 incidents missed by the first stage detector.

The parameters $\beta$ and $T$ were optimized by conducting at least 50 GP runs for various settings. The detection performance was largely insensitive to $T$ between values of 2 and 8. The results showed that $\beta$ must exceed 0.1 in order to achieve no false alarms, and that a good balance between maximizing detected incidents and minimizing false alarms was achieved by setting $\beta$ to 0.5.

More GP runs were conducted with $\beta = 0.5$ and a population size of 4000. Figure 6 plots the resulting incident scores against false alarm rate for the best detector from each GP run for various $T$ values.

Recall that two grade 4 incidents were missed at the first stage and so a second stage detector could identify a maximum of 30 incidents with a TPS of 65. This was achieved at the expense of giving 6 false alarms, i.e. less than a single false alarm per night on average. These false alarms are arguably grade 5 incidents and two of them actually refer to the same event which was detected at adjacent minutes and sites. The maximum FOM was achieved when two further grade 4 incidents were missed to bring TPS down to 63, but this detector abolished all false alarms.

The incidents detected by the best second stage detector (i.e. the one which gave the maximum FOM) are shown in Table 3. The *lag* column gives the lag in minutes between the marked incident onset and the time of detection. This was negative when the incident started to manifest itself in the traffic data before the manually marked onset, e.g. because of a gradual onset spread across lanes. The column shows that most incidents were detected within two minutes. The last column gives the number of hits per incident and shows that grade 1 incidents received multiple hits but higher grades tended to be hit only once.

Table 3: Incidents detected by the best second stage detector. The lag is in minutes and the number of detections per incident is given.

| date | time | site | grade | lag | no. of hits |
|---|---|---|---|---|---|
| 3 | 2320 | 4989a | 1 | -1 | 5 |
| 4 | 0319 | 4927a | 4 | 0 | 1 |
| 9 | 2253 | 4912a | 3 | 0 | 4 |
| 10 | 0203 | 4935a | 3 | 0 | 1 |
| 11 | 2200 | 4742a | 1 | 0 | 3 |
| 11 | 2232 | 4762a | 2 | 0 | 1 |
| 12 | 0540 | 4912a | 3 | 1 | 1 |
| 28 | 0035 | 4932a | 3 | 0 | 1 |
| 5 | 2334 | 4955b | 3 | 2 | 1 |
| 5 | 2337 | 4940b | 4 | -1 | 1 |
| 6 | 0040 | 4757b | 2 | 1 | 2 |
| 6 | 0336 | 4955b | 4 | 5 | 1 |
| 6 | 0515 | 4955b | 4 | 2 | 2 |
| 9 | 2243 | 4945b | 4 | 0 | 3 |
| 10 | 0015 | 4935b | 3 | 0 | 2 |
| 11 | 0346 | 5002b | 4 | 0 | 1 |
| 11 | 0431 | 4949b | 2 | 5 | 1 |
| 11 | 2258 | 4949b | 2 | 1 | 1 |
| 11 | 2340 | 4949b | 2 | 1 | 1 |
| 12 | 0050 | 4945b | 3 | 0 | 1 |
| 12 | 0128 | 4752b | 3 | 3 | 1 |
| 12 | 0227 | 4949b | 3 | -4 | 4 |
| 26 | 2346 | 4888b | 2 | -5 | 1 |
| 26 | 2352 | 4797b | 2 | 5 | 1 |
| 27 | 0227 | 4949b | 4 | 0 | 1 |
| 27 | 0311 | 5002b | 3 | 0 | 1 |
| 27 | 2151 | 4832b | 1 | 0 | 6 |
| 28 | 0436 | 4737b | 2 | 3 | 1 |

### 3.6 Results summary

The task was to detect incidents on the M25 at periods of low traffic occupancy. This was approached by training GP to detect the onset of incidents which occurred during the night (approximately between 2200h and 0600h) whilst producing a near-zero false alarm rate.

The precise onset of incidents was often poorly defined and so the task was tackled by a two-stage evolution strategy. The first stage evolved for itself which traffic data best characterized an incident onset by distinguishing data proximate to incidents from a sample of non-incident data. The second stage was then required to minimize the false alarm rate whilst retaining at least a single detection per incident.

The best first stage detector missed two grade 4 incidents (the least obvious incidents) and gave 382 false alarms from 7 nights worth of data using both carriageways. The best second stage detector missed another two grade 4 incidents but abolished all the false alarms. Other second stage detectors retained all incident detections but gave 6 false alarms.

### 3.7 Current work

The activity of manually marking more than 1000 incidents over three years worth of traffic data has shed light into the nature of the incidents, which in turn has increased the subjective threshold as to what constitutes an incident. This has improved the quality of the "truth" in the scaled-up project, and it is hoped that this will be reflected in the detection performance. Tests are currently being undertaken to assess the generalization of the detection performance. Other modelling parameters such as the effect of weather, e.g. rain and fog measurements, may be incorporated as inputs to future detectors.

## 4 Conclusions

This paper describes two projects that apply genetic programming to real world problems proposed by the UK Highways Agency. The first project evolved motorway journey time predictors which were required to be reliable during high-flow low-speed conditions. These peak travel periods contained the most variable traffic characteristics and notably included congestion waves. The evolved predictors processed lane-independent traffic quantities and improved on the accuracy of equivalent naive predictors. Naive predictions are typically difficult to out-perform due to the inherent irregularities in the traffic data during congested periods.

The second project evolved motorway incident detectors for low-flow high-speed conditions, i.e. late at night. A Staged GP method was employed to identify the subtle anomalies which correspond to late-night incident onsets. The evolved detectors processed lane-specific traffic quantities and achieved near-zero false alarm rates whilst only missing very few minor incidents. Work is currently underway to scale this project up to extensive traffic data sets and to assess detection generalization.

It is hoped that an insight can be gained into the principles underlying the two projects by interpreting the structures of the evolved processors. Further understanding could be obtained by analyzing the behaviour of the evolved processors in different traffic states.

**Acknowledgment**

The authors wish to thank the UK Highways Agency for facilitating the data for this study which is highly specific to MIDAS and to the M25 motorway.

**References**

[Beale, 2002] Beale, S. (2002). Traffic Data: Less Is More. HA Internal Report, Highways Agency, Bristol, UK.

[Collins et al., 1979] Collins J. F., Hopkins C. M. and Martin J. A. (1979). Automatic incident detection - TRRL algorithms HIOCC and PATREG. TRL Report SR 526, Transport Research Laboratory, Cowthorne, UK.

[Howard and Roberts, 1999] Howard D. and Roberts S. C. (1999). A Staged Genetic Programming Strategy for Image Analysis. In Banzhaf, Daida, Eiben, Garzon, Honavar, Jakiela and Smith (eds), *Proceedings of the Genetic and Evolutionary Computation Conference, 1047–1052*, Morgan Kaufmann.

[Mahalel and Hakkert, 1985] Mahalel D. and Hakkert A. S. (1985). Time Series Model for Vehicle Speeds. Transportation Research Vol. 19B(3), 217–225.

[Moorthy and Radcliffe, 1988] Moorthy C. K. and Radcliffe B. G. (1988). Short term traffic forecasting using time series methods. Tansportation Planning & Technology, Vol. 12, 45–46.

[Payne et al., 1975] Payne H. J., Goodwin D. N. and Teener M. D. (1975). Evaluation of existing incident detection algorithms. Technology Service Corporation, Santa Monica, California.