Technical Report Documentation Page

1. Report No. FHWA/TX-08/0-5422-1	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Ramp Closure Strategies for Incident Management		5. Report Date October 2007 6. Performing Organization Code
7. Author(s) Dr. S. Travis Waller, Stephen Boyles, David Fajardo, Dr. Ampol Karoonsoontawong		8. Performing Organization Report No. 0-5422-1
9. Performing Organization Name and Address Center for Transportation Research The University of Texas at Austin 3208 Red River, Suite 200 Austin, TX 78705-2650		10. Work Unit No. (TRAIS) 11. Contract or Grant No. 0-5422
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Implementation Office P.O. Box 5080 Austin, TX 78763-5080		13. Type of Report and Period Covered Technical Report September 2005 - August 2007 14. Sponsoring Agency Code

15. Supplementary Notes

Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.

16. Abstract

Non-recurring congestion accounts for a significant portion of freeway delay experienced by motorists in both rural and urban areas. Because access to these facilities is controlled by onramps, these can play a major role in minimizing the impact that unexpected incidents have on system performance. This project considered both methodological and practical considerations in using ramp closure as an incident management strategy. Novel methods for predicting incident severity were developed using probabilistic analysis, as well as a two-phase model for recommending which ramps should be closed, and for how long. These models were designed to be integrated with operations at a traffic management center, where responsibility for ramp closure is likely to be located. Further, an overview of closure devices is provided, along with their comparative advantages and disadvantages.

17. Key Words Incident management, ramp treatment, traffic assignment, microsimulation, incident modeling	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161; www.ntis.gov.

19. Security Classif. (of report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	88	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

RAMP CLOSURE STRATEGIES FOR INCIDENT MANAGEMENT

S. Travis Waller
Stephen Boyles
David Fajardo
Ampol Karoonsoontawong

CTR Technical Report:	0-5422-1
Report Date:	October 2007
Project:	0-5422
Project Title:	Ramp Treatment and Dynamic Closure Strategies for Incident Traffic Management
Sponsoring Agency:	Texas Department of Transportation
Performing Agency:	Center for Transportation Research at The University of Texas at Austin

Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.

Center for Transportation Research
The University of Texas at Austin
3208 Red River
Austin, TX 78705

www.utexas.edu/research/ctr

Disclaimers

Author's Disclaimer: The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official view or policies of the Federal Highway Administration or the Texas Department of Transportation (TxDOT). This report does not constitute a standard, specification, or regulation.

Patent Disclaimer: There was no invention or discovery conceived or first actually reduced to practice in the course of or under this contract, including any art, method, process, machine manufacture, design or composition of matter, or any new useful improvement thereof, or any variety of plant, which is or may be patentable under the patent laws of the United States of America or any foreign country.

Notice: The United States Government and the State of Texas do not endorse products or manufacturers. If trade or manufacturers' names appear herein, it is solely because they are considered essential to the object of this report.

Engineering Disclaimer

NOT INTENDED FOR CONSTRUCTION, BIDDING, OR PERMIT PURPOSES.

Research Supervisor: S. Travis Waller

Acknowledgments

The authors would like to express their appreciation towards the Texas Department of Transportation for support of this research. In particular, thanks are due to Carlos Chavez, Charles Koonce, Edgar Fino, Gus Morgan, Wade Odell, and John Windlow for their assistance and leadership throughout this project.

Products

This report also contains product 0-5422-P1 (guidelines and best practices for ramp closure) and the documentation for product 0-5422-P2 (incident duration prediction software). Chapters 4, 5, 6, and Appendix C together form 0-5422-P1, and the 0-5422-P2 documentation is found in Appendix A.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Project Overview

With ever-increasing congestion levels throughout the nation, it comes as no surprise that huge amounts of research, political, and technical effort have been invested in alleviating its negative impacts. A substantial amount of freeway congestion, if not the majority, is due to nonrecurring events such as incidents [1], which has spurred significant innovation in incident management techniques and practices designed to minimize the impact of these unexpected events.

A large variety of incident management strategies have been developed, with different aims: for instance, information dissemination with variable message signs (VMSs) or highway advisory radio, reducing the clearance time of incidents with dedicated incident-response vehicles or policy guidelines for emergency responders, and operational strategies such as temporary opening of high-occupancy vehicle lanes to general traffic have all been proposed to mitigate the impact of incidents. These efforts typically focus on freeway incidents, because these facilities carry higher volumes of traffic, and because their limited-access nature restricts motorists' opportunities to detour when encountering congestion.

One recently proposed strategy for severe or extended incidents is to close selected freeway onramps in the vicinity of the incident. This strategy is logical, given that access to freeways is completely controlled by these ramps, and that ramp treatment strategies have shown success in reducing recurring congestion. Ramp metering is widely used to this end, but even complete ramp closure has been employed as a regular tool for congestion management.

The idea of closing onramps to improve traffic flow dates back at least four decades to Miesse [2], who conducted basic microsimulation analysis to study the improvement attained from ramp closure, and to experiments conducted by the Michigan Department of Transportation in the early 1960s on peak-hour ramp closure in Detroit, finding substantial increases in freeway throughput (up to 13.7%) and freeway speed (typically 10 mph) without creating excessive problems on arterials [3]. The Federal Highway Administration's Freeway Management Handbook [4] lists several other successful deployments of ramp closure for this purpose (for instance, in Houston, Los Angeles, San Antonio, and Fort Worth, as well as in Osaka and Tokyo, Japan). In the literature, ramp closure has not been specifically tested as an incident management strategy, although it is sometimes performed on an ad hoc basis by on-site emergency responders.

Both common sense and experience indicate that closing onramps does not always lead to a net system benefit: a study conducted by Prevedorous in Hawaii showed only modest increases in travel speeds at some locations, with decreases in travel speed being observed at other locations [5]. Although this may have been due to specific experimental factors (such as the two-week study period being insufficient for motorists to adapt to the ramp closure, or only considering travel speed as a measure of effectiveness), this nevertheless suggests that a rigorous analysis framework for evaluating the impact of ramp closure is needed, especially in the context of incident management where the closed onramps will probably be unexpected by motorists.

1.2 Ramp Closure Framework

In this project, making a decision about ramp closure is formulated as a three-step process:

1. Is the incident severe enough to consider ramp closure?
2. If so, *which* ramps (if any) should be closed?
3. If any ramps are closed, for *how long* should they remain closed?

This division is based on several practical considerations. First, from an operational standpoint, closing a ramp is not an easy task. Regardless of the specific type of access control (automatic gates, manual barriers, etc.), safely closing an onramp requires time and diversion of resources away from the incident itself. Thus, some sort of pre-screening should be available before any ramp closure policy is seriously considered. Secondly, dividing the questions of *which* ramps and for *how long* carries a major implementation advantage: answering the latter question requires dynamic models, which require more computation time than is needed to answer the former. Because time is of the essence in any incident management application, and because the question of *how long* does not need to be answered immediately, it makes sense to answer the two questions separately.

Thus, we propose a **Phase I** model to first answer the question of which ramps should be closed (leaving open the possibility that the best option is not to close any ramps at all). This model runs quickly (in a matter of seconds) and allows the ramp closure to begin with minimal delay. Following this, a **Phase II** model is run to determine when these ramps should be reopened. Of course, the judgment and experience of human operators and responders should be included as well throughout the entire process, as they interpret the results of these models.

Returning to the first step of pre-screening incidents, our goal was to develop a process of quickly identifying how severe an incident is, so that only sufficiently severe incidents would be further examined for ramp closure. Incident severity can be measured in many ways, based on operational factors (total delay caused to motorists or the number of lanes blocked), safety factors (the amount of property damage or the severity of human injuries), the incident clearance time, and so on.

Because the goal of ramp treatment strategies is inherently based in improving system operations (or at least corridor operations), rather than affecting the actual incident itself, the total delay caused by the incident is the most apt measure of incident severity for this research. By measuring the amount of delay caused by an incident relative to base conditions, in addition to providing a prescreening tool, we simultaneously establish a benchmark for measuring different possible ramp closure strategies.

Incident delay is most affected by the incident's duration (clearance time), and by the prevailing traffic volumes. Naturally, incidents that last longer, or that occur during periods of heavier traffic flow, will cause more delay. Traffic flow characteristics can be inferred from historical information, or from intelligent transportation systems (ITS) devices, such as inductive loops; however, the task of estimating incident duration while an incident is in progress is considerably more difficult. Thus, we develop two models to predict incident severity; the first makes a prediction about the incident duration, which in turn is used to estimate the delay which the incident will cause. This delay figure will then be used to see if the incident is severe enough to warrant further examination regarding ramp closure. The specific threshold should be determined based on the specific corridor where the incident occurs, characteristics of onramps in its vicinity, availability of agency resources for ramp closure analysis, and the judgment of the agency implementing such a procedure.

1.3 Research Tools

The previous section outlines the models that were developed and used in this project. In particular, four distinct tasks need to be accomplished: estimating incident duration, estimating incident delay, deciding which ramps to close, and deciding how long they should remain closed.

We develop a novel incident duration prediction model based on the naive Bayesian classifier; Chapter 2 explains this model in full detail. As part of this project, we created NBIDS (Naive Bayesian Incident Duration Software), a software program which can be applied for this purpose. This program and its documentation are provided as deliverable P2 for this project.

Our survey of the incident delay prediction literature revealed that none of the models account for the inherent uncertainty in incident management. This turns out to be a significant deficiency (Chapter 3 includes a simple example where this leads to inefficient dispatching of an incident response vehicle), so we modify an existing delay prediction model to account for this. This also allows us to fully take advantage of the results of the Bayesian incident duration prediction model, which is stochastic in nature. We develop analytic formulae, and also develop a mesoscopic simulation evaluation technique for scenarios too complex for exact analysis; these simulations are based on the Cell Transmission Model, explained in more detail in Appendix B.

The Phase I ("which ramps") model is based on the principles of macroscopic traffic assignment, allowing us to examine the system-wide effects of a ramp closure policy. This is needed because closing onramps will result in re-routing and changes in flows throughout the entire system — for instance, some vehicles may choose to follow an alternate arterial path to their destinations, rather than simply merging onto the freeway at the next opportunity. This choice of Phase I model also allows us to build on the significant body of research in traffic assignment, which includes a number of very efficient algorithms for large-scale networks.

The Phase II ("how long") model is based on microsimulation, as a detailed analysis of queuing and other dynamic phenomena is needed to answer this question with any degree of realism. In this project, we use the commercial microsimulator VISSIM (developed by PTV), although any microsimulator can be used in its place.

1.4 Outline

The remainder of this report is organized as follows: Chapter 2 describes a naive Bayesian classifier model for predicting incident duration, and explains its advantages over existing models. Chapter 3 presents a delay prediction model, using the input from the Bayesian duration prediction model. Chapters 4 and 5 are concerned with developing an ideal ramp closure policy, respectively describing models for *where* and *when* ramps should be closed. Chapter 6 then discusses some issues related to the practical deployment of this modeling framework for real-time incident management. Finally, Chapter 7 summarizes the document and reiterates the key conclusions.

2. Incident Duration Prediction

2.1 Introduction

Chapter 1 explains how predicting the duration of an incident is the first step in the pre-screening process in identifying incidents which may warrant closing onramps. A variety of such models exist in the literature to make this prediction based on observable characteristics. Linear regression models are common, using characteristics such as incident type, weather condition, and number of vehicles and lanes involved as independent variables; Khattak et al. [6], Garib et al. [7], and Ozbay and Kachroo [8] use variations of this approach. Jones et al. [9] apply a Poisson regression on similar variables to incidents in the Seattle area. Giuliano [10] aggregates incidents into broad categories and estimates models for each category. Golob [11] and Sullivan [12] both fit lognormal distributions to incident data, and Nam and Mannering [13] use hazard-based models which provide information not only on the total incident duration, but also on the probability that an incident, known to have already existed for a certain period of time, will be cleared in the next small time interval. Ozbay and Kachroo [8] construct decision trees which do not require knowledge of all observable incident characteristics. Smith and Smith [14] also suggest the use of nonparametric regression, where incident duration is estimated based on similar incidents in the past.

Implementation of most of these models is often difficult, as almost all incident duration prediction models, with the exception of decision trees, nominally require complete knowledge of all incident characteristics to make a prediction. In practice, this knowledge is often incomplete and obtained sequentially; for instance, as emergency vehicles arrive, they can report additional information that may not have been apparent when the incident was first reported, such as the number of injured persons. Although techniques exist to make predictions based on interpolated values of unknown independent variables [6], this reduces accuracy and complicates the prediction process. Decision tree models do not suffer from the limitation of needing observation of all independent variables, but these are deterministic and give no information on the reliability or variability in the prediction. Probabilistic predictions give more information on this uncertainty and can be more useful in deciding whether or not a particular incident management strategy should be adopted.

Thus, we develop a probabilistic model based on a naïve Bayesian classifier which gives useful predictions regardless of the amount of information known about the incident. To our knowledge, this is the first application of this type of model to incident duration prediction.

The prime advantages of the naïve Bayesian classifier, which we elaborate on in the following section, are as follows:

1. Its operation is simple and intuitive, relying only on basic laws of probability.

2. It accommodates limited information, as seen in real-world incident management.

2a. As a corollary, this allows a broader set of model parameters to be used, as the model does not require observations for all independent variables.

3. Being explicitly probabilistic, it reports results in a form better suited for incident management policy.

4. It is robust to outliers.

5. It can account for information received at varying points in time.

To our knowledge, no current incident duration model contains all of these features. The Bayesian networks approach (applied by Ozbay and Noyan [15]) is most similar, but requires additional calibration to characterize the dependencies in the model; doing this correctly requires some sophistication in statistical techniques and incident management. In contrast, the naive Bayesian approach requires only a basic understanding of probability to use correctly.

The next section describes the naïve Bayesian classifier, and the development of the incident duration model in greater detail. Section 2.3 describes the NBIDS software developed in this research.

2.2 Model Development

This section briefly describes the naïve Bayesian classifier (NBC) and its adaptation for incident duration prediction. The aim of the NBC, as with other classifiers, is to assign an object I to one of a discrete set of categories $C_1, C_2, \ldots, C_m$ based on its observable attributes $X_1, X_2, \ldots, X_n$. The NBC calculates the probability that I belongs to each category, conditioning on the observed attributes; I is typically assigned to the category with the greatest such probability. This classifier is naïve in the sense that it makes the strong assumption that the attributes are mutually conditionally independent; that is, the conditional probability that I belongs to a particular class given the value of some attribute is independent of the values of all other attributes. Despite this unrealistic assumption, empirical studies demonstrate that this assumption need not significantly compromise the accuracy of the prediction, and NBCs are used in a variety of applications, including document classification [16], medical diagnosis [17], systems performance management [18], and other fields. Domingos and Pazzani [19] prove optimality of the NBC under certain conditions even when the conditional independence assumption is violated.

This probability calculation is straightforward; conditioning on the observed attributes, we want to find the probability that I belongs to each category, that is, $\Pr(I \in C_i \mid X_1, X_2, \ldots, X_n)$. Applying Bayes' Theorem, this is rewritten as

$$\Pr(I \in C_i \mid X_1, X_2, \ldots, X_n) = \frac{\Pr(I \in C_i)\Pr(X_1, X_2, \ldots, X_n \mid C_i)}{\Pr(X_1, X_2, \ldots, X_n)} \tag{2.1}$$

Under the mutual conditional independence assumption, this reduces to

$$\Pr(I \in C_i \mid X_1, X_2, \ldots, X_n) = \frac{\Pr(I \in C_i)\prod_{j=1}^{m}\Pr(X_j \mid I \in C_i)}{\Pr(X_1, X_2, \ldots, X_n)} \tag{2.2}$$

for each category C_i. Because the denominator will be the same for all categories, we need only calculate the numerator for each category i, choosing

$$i^* \in \operatorname{argmax} \Pr(I \in C_i)\prod_{j=1}^{m}\Pr(X_j \mid I \in C_i) \tag{2.3}$$

and assigning I to category C_{i^*}.

In the context of incident duration prediction, the attributes X correspond to observable incident characteristics, such as the number of injured persons, the number of blocked lanes, the location of the incident, weather conditions, and so on. It is also necessary to define discrete categories of incident duration to classify the incidents; these might coincide with local incident management policy. For the sake of example, say a particular agency has decided that a highway advisory radio (HAR) message should be recorded if an incident is expected to last more than half an hour, and that a ramp closure policy should be enacted if the incident is expected to last more than an hour. For this agency, it would be appropriate to designate three classes of incidents: those lasting less than half an hour, those lasting at least half an hour but less than an hour, and those lasting more than an hour. When an incident occurs, the NBC would calculate the probability that the incident's duration will fall into each of these three intervals, and the appropriate policy could be chosen based on what is most likely.

Reporting the probability for each duration category also provides useful information about the certainty of the prediction to incident management staff in an intuitive manner; this contrasts with non-probabilistic models which only report a single "expected" duration. Additionally, by classifying all incidents into broad categories, the resulting model is robust with respect to outliers, which abound in incident duration data sets because of high variability in response times and incident severity.

Table 2.1 shows selected parameters for the NBC model estimated in this work, which are used here to illustrate the application of the NBC. The first row of "base" probabilities contains the values which give the probability that an incident falls into these three intervals without any additional knowledge. The next two rows give the conditional probabilities showing the likelihood of that particular attribute, given that the interval falls into a particular class. (The values in each row need not sum to unity as they condition on different events.) Thus, if we are given an incident involving one vehicle and one fatality, the maximand in (2.3) can be calculated for each of the three categories by multiplying the relevant entries in each column, yielding values of 0, 0.000316, and 0.000183, respectively; thus, Category 2 (30 to 60 minutes) is the most likely duration for this incident.

Table 2.1: Selected NBC parameters for example

Attribute	< 30 mins.	30–60 mins.	> 60 mins.
Base probability	0.36	0.25	0.39
One passenger car involved	0.251	0.421	0.156
One or more fatalities	0	0.003	0.003

This probability calculation has the advantage of using whatever information is available to make the prediction; regardless of how much or how little is known about the incident, a valid prediction will be made based on what is provided, although certainly more information leads to better prediction. One interesting and important side benefit of this property is the ability to use more independent variables, as, unlike traditional models, one does not need an observed value of all variables. Thus, parameter selection is limited only by what is present in a calibration set, not by what is likely to be observed in most incidents. The parameters in this model (that is, the probabilities $\Pr(I \in C_i)$ and $\Pr(X_j \mid I \in C_i)$) may be calibrated from records of past incidents.

It may also be desirable to update these probability calculations as more information becomes available, or if the incident has lasted long enough that it is useful to restrict attention to incidents of this duration or greater. For instance, in the previous example, if a given incident has already lasted forty minutes, the probability that its total duration is less than thirty minutes is clearly zero, and the model should yield the same result. This can be done by conditioning on an additional event T which occurs if an incident's total duration is at least t, where t is the amount of time since the beginning of the current incident. Then we can write the probability calculation as

$$i^* \in \operatorname{argmax} \Pr(I \in C_i \mid T) \prod_{j=1}^{m} \Pr(X_j \mid T, I \in C_i) \tag{2.4}$$

This formulation is flexible in that it can update the most probable category at any time (such as when new information is received); however, it is not without its drawbacks. First, the number of parameters needed grows substantially (multiplied by the number of possible values of T). Although this is not a major obstacle, if a computer program is used to calculate and store these parameters from a large database of incidents, the previous formulation can be completely and succinctly summarized in a single table, which is useful if one wishes to make qualitative statements about the impact of various attributes. Second, if the current time is large, there may be few incidents where duration is greater than t, and the accuracy of the conditional probabilities in (2.4) may suffer due to a small sample size. Nevertheless, the ability to accommodate new information is of importance, and completely absent in nearly all current incident duration models.

One difficulty occurs if any of the conditional probabilities in (2.3) or (2.4) are zero, as the category corresponding to the resulting product will have zero probability regardless of the other factors. It may be undesirable for one zero quantity to invalidate a number of other factors that may strongly suggest one category; for this reason, zero probabilities are replaced by a small positive number when calculating these products. The sensitivity of the classifier to the choice of this number was not examined in this work.

2.3 NBIDS

As part of this project (deliverable 0-5422-P1), a software program was created to assist TxDOT with using the Bayesian classifier to predict incident duration. This software, titled Naive Bayesian Incident Duration Software (NBIDS), was coded using FreeBASIC, and accompanying documentation, including a guided tutorial and source code, is also provided in Appendix A.

Predicting incident duration with NBIDS is a three-step process:

1. Create a historical incident database
2. Initialize the incident database
3. Predict incident durations

Creating the database can be done by obtaining past incident records, organizing them with a spreadsheet program, and saving in a comma-separated value (CSV) text format. Depending on the format in which the data is initially received, little to no actual processing may be required in this step.

Initializing the database is the process of letting NBIDS know which fields in the database are the ones to use for predicting incident duration (e.g., you might select the number of vehicles and a location field, while ignoring a general comments field),defining the categories of incident length, and defining the attributes used in the classifier, as described earlier in this chapter. This is the most labor-intensive part of the process, but the tutorial and documentation explain this step in detail, and this step only needs to be performed once when setting up the database.

Once initialized, predicting incident duration can be accomplished in a matter of seconds. NBIDS prompts the operator for all of the information known about the incident; any unknown information is simply skipped, and the Bayesian prediction is calculated and output to the user. A screenshot of the NBIDS menu is seen in Figure 2.1.

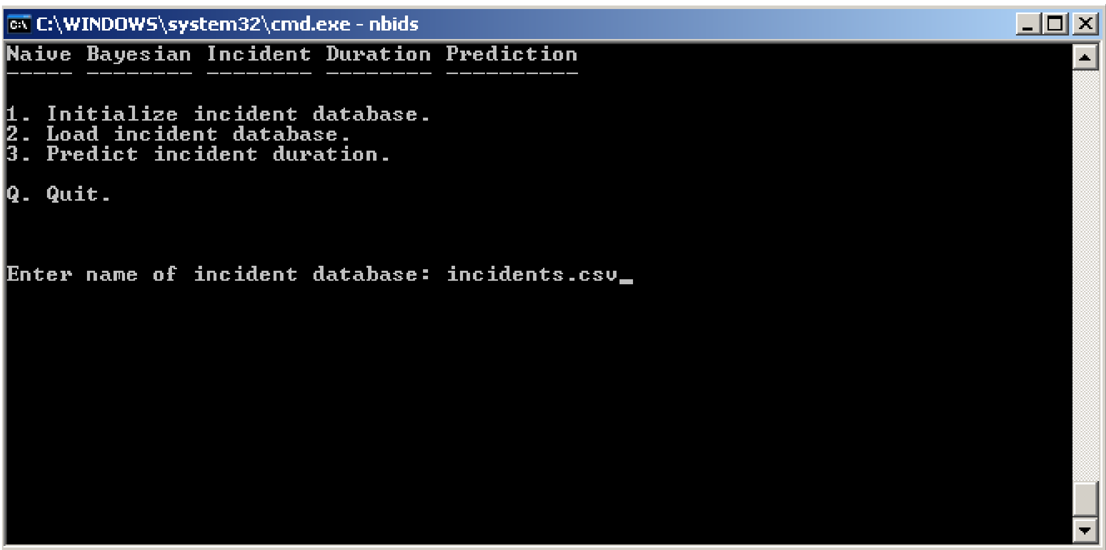


Figure 2.1: NBIDS title menu.

3. Incident Delay Prediction

3.1 Introduction

There are models that can predict incident delay when provided the duration of closure. Although described more fully in the next section, this suggests a natural incident management framework. Incident management personnel learn characteristics of an incident, such as the number of vehicles involved or the weather conditions, through a variety of means. This information is then used to estimate incident duration, which in turn is used to predict delay caused by this incident, using models described in Section 3.2.

However, incident duration can never be predicted with absolute certainty, and failure to account for this can lead to inefficient incident response. Particularly, using only the expected incident duration as input to the delay model will always give a wrong prediction and systematically underestimate expected incident delay. In essence, it is impossible to obtain an accurate prediction without considering the full range of possible incident durations.

The following sections give mathematical proof of this fact, but here we present a small motivating example which, while admittedly simplistic, illustrates this principle. A traffic management center learns of two incidents, A and B (see Table 3.1), but there is only one available incident response vehicle, so they decide to dispatch it to the incident which will cause the greater delay.

Table 3.1: Small example to demonstrate importance of modeling uncertainty.

	Incident A	Incident B
Duration	5 or 25	17
Expected duration	15	17
Delay using expected duration	225	289
True expected delay	325	289

Based on the information they have about these incidents, they predict that Incident A will either last for 5 minutes, or for 25 minutes (with equal probability); thus the incident duration model reports an expected duration of 15 minutes. Incident B, on the other hand, will last exactly 17 minutes with absolute certainty. Typically, total incident delay is proportional to the square of incident duration. So, if the expected duration is used, one predicts a delay of 225 and 289 units for Incidents A and B, respectively; thus, the vehicle is sent to assist with Incident B.

Instead of using the average duration, let’s calculate the delay for both scenarios. If Incident A only lasts 5 minutes, it will create a delay of 25 units, but if it lasts 25 minutes, it will create a delay of 625 units. Because these are equally likely, the true expected delay due to Incident A is the average of these, or 325 units. Because there is no uncertainty with Incident B, it will still have a delay of 289 units. Thus, when estimating delay in this manner, we conclude that the vehicle should be dispatched to Incident A instead.

This inconsistency is not the result of the simplifications made here; rather, in the next section, we show that this is due to fundamental principles of probability. Thus, it is incorrect to use the average duration to predict delay, and one must consider the underlying probability distribution. Surprisingly, to the authors' knowledge, this phenomenon has not been described in the incident management literature. Thus, this paper attempts to remedy this situation by proposing a *stochastic* incident delay prediction model.

In the next section, we describe past work on incident delay prediction. In section 2 we define the stochastic delay prediction problem more precisely, and derive exact formulae to account for incident duration probability distributions that have been encountered in the literature.

3.2 Background

A variety of models have been proposed in the literature to predict delay caused by an incident: Wirasinghe [20] and Morales [21] develop analytical formulae based on the classical shockwave theory of Lighthill and Whitham [22]; given a predicted incident duration, one can estimate the resulting delay. Al-Deek et al. [23] propose an online prediction model using loop detectors to calculate delay *a posteriori*; while practical and useful in the case of multiple incidents (a property few incident delay prediction models share), this method as presented is not suited for real-time prediction, but rather post-incident analysis. However, Garib et al. [7] combine this work with a linear regression model that can predict delay while an incident is in progress.

However, while linear regression models have the advantage of simplicity, they carry several major disadvantages. In the context of incident management, the most serious is that they require an input value for each independent variable, at least nominally. While techniques exist to interpolate missing data from other observations (see, for instance, Bhat [24]), this reduces accuracy and complicates the prediction process. Because information about incidents is often incomplete and obtained sequentially (rather than all at once), this suggests another method is more appropriate.

Generally speaking, the models based on traffic flow theory are more transferable and flexible as they are founded on fundamental principles. In fact, the models of Wirasinghe [20] and Morales [21] can be applied off-the-shelf when combined with an incident *duration* prediction model.

Many models exist to estimate incident duration, as seen in the previous chapter. However, as demonstrated in the above example, using a single, expected value of incident duration always underestimates delay in the presence of uncertainty. This effect can be traced to an inequality of Jensen [25] regarding convex functions: if f is convex, and X is a random variable, we must have $E[f(X)] \geq f(E[X])$. Specifically, let f represent incident delay, and X represent the uncertain incident duration. Because f is proportional to the square of X, $f(X)$ is strictly convex, and thus the expected value of f must be greater than the delay that would result from the expected value of X: $E[f(X)] \geq f(E[X])$. This result applies no matter what the distribution of incident duration is. Therefore, we prefer incident duration models that give information about the entire distribution of possible incident durations to those which only report single values. This is another reason why the naive Bayesian classifier presented in the previous chapter is useful. In the next section, we develop analytical formulae for total incident delay based on both the Bayesian models and the lognormal regression.

3.3 Analytical Stochastic Delay Formulae

The analysis in this section is based on Wirasinghe's [20] delay formula. The other commonly-used formula is that of Morales [21] which is based on the same principles, but involves considerably more parameters. Thus, the approach demonstrated here is equally applicable to this formula as well, although a parallel derivation is omitted for brevity. In particular, Wirasinghe's formula for total delay due to a stationary incident is

$$D_u = \frac{1}{2}\tau^2 \frac{(q_1 - q_4)(q_3 - q_4)}{q_3 - q_1}$$

where D_u is total delay, τ is the incident duration, and q_1, q_3, and q_4 are the initial, recovery, and metered vehicle flow rates (see Figure 3.1 for identification of these regions on a shockwave diagram). Real-time values of q_1, q_3, and q_4 can be estimated from loop detectors in the area.

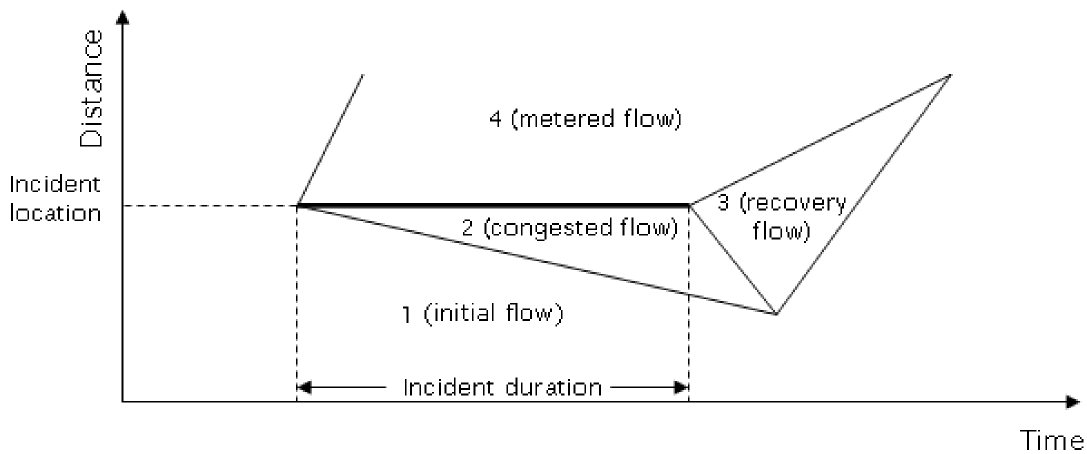


Figure 3.1: Idealized shockwave diagram for a stationary incident.

Assuming some probability distribution $f(\tau)$ for incident duration, we can express the expected delay as

$$E[D_u] = \frac{(q_1 - q_4)(q_3 - q_4)}{2(q_3 - q_1)} E[\tau^2] = \frac{(q_1 - q_4)(q_3 - q_4)}{2(q_3 - q_1)} \int_0^\infty \tau^2 f(\tau)\, d\tau$$

which, depending on f can be evaluated analytically or numerically. Therefore, the remainder of this section considers different probability distributions for incident duration which have been proposed in the literature, and derives corresponding formulae. In particular, we consider various probability distributions implied by Bayesian classification, and the lognormal distribution, both of which are repeatedly seen in the incident duration prediction literature.

Bayesian methods classify incidents into categories based on incident duration; for instance, reporting the probability that an incident will last between 0 and 15 minutes, between 15 and 25 minutes, between 25 and 35 minutes, and so on. (see Table 3.2 for an example); however, to calculate expected delay, we need to specify the distribution more precisely. The simplest approach is to assume a uniform distribution within each category, as in Figure 3.2, such that the area under the distribution in each category is equal to the probability that the incident lasts that duration.

Table 3.2: Example output from a Bayesian incident duration model.

Incident duration range	(0,15)	(15, 25)	(25,35)	(35, 50)	(50, 75)
Probability	0.05	0.13	0.37	0.34	0.11

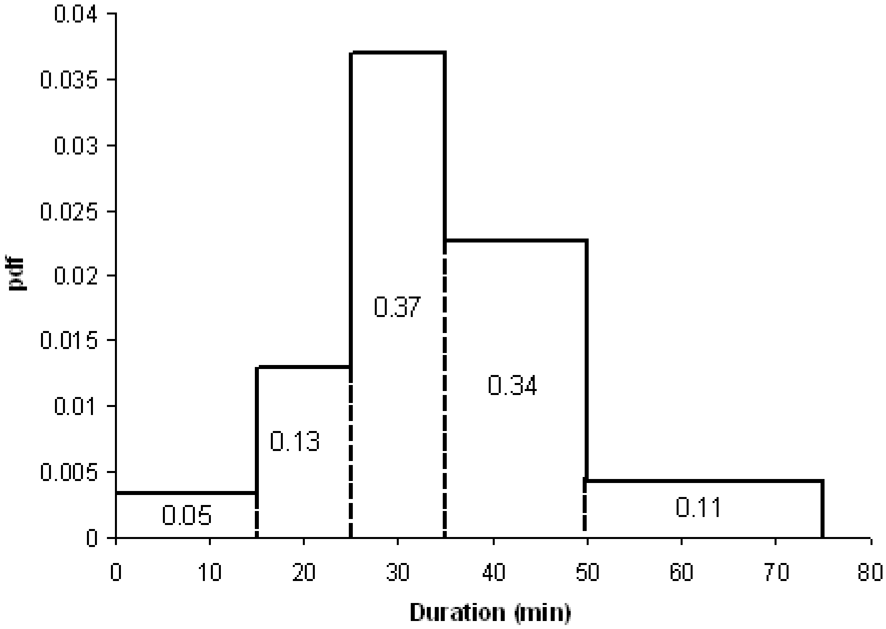


Figure 3.2: Probability density function corresponding to Bayesian example.

Mathematically, let there be n categories of incident duration where each category i contains incidents lasting longer than ξ_{i-1} but no longer than ξ_i units of time; the probability that an incident's duration is of category i is p_i. Thus these categories form the intervals $(\xi_0, \xi_1], (\xi_1, \xi_2], \ldots, (\xi_{n-1}, \xi_n]$ and the probability density function can be written

$$f(\tau) = \begin{cases} 0 & \tau \le \xi_0 \\ \dfrac{p_i}{\xi_i - \xi_{i-1}} & \xi_{i-1} < \tau \le \xi_i \\ 0 & \tau > \xi_n \end{cases}$$

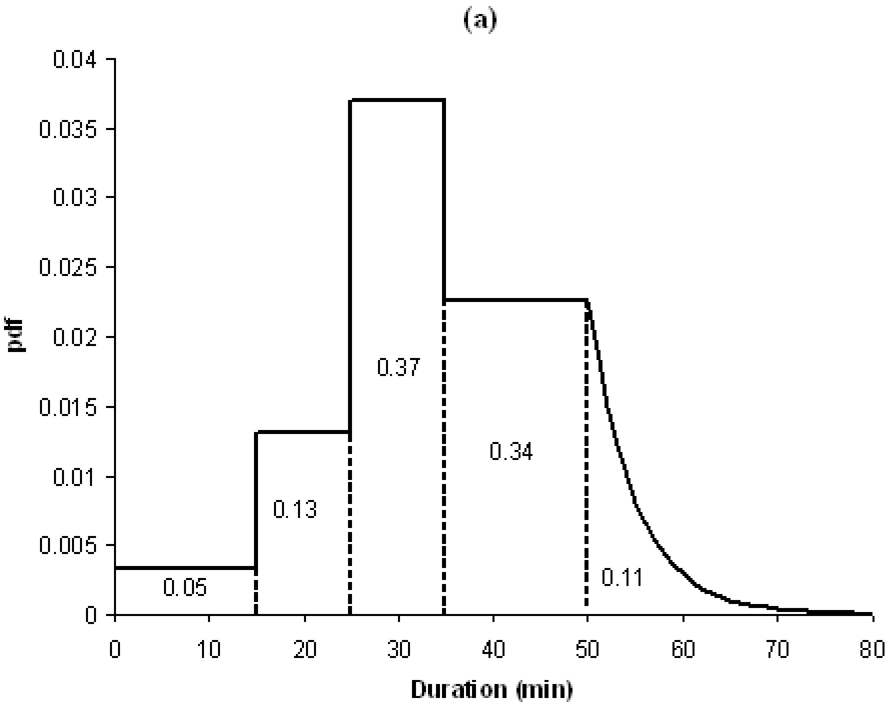
Then

$$
\begin{aligned}
E[\tau^2] &= \int_0^\infty \tau^2 f(\tau)\, d\tau \\
&= \sum_{i=1}^{n} \int_{\xi_{i-1}}^{\xi_i} \frac{p_i \tau^2}{\xi_i - \xi_{i-1}}\, d\tau \\
&= \sum_{i=1}^{n} \frac{p_i}{3} \frac{\xi_i^3 - \xi_{i-1}^3}{\xi_i - \xi_{i-1}} \\
&= \sum_{i=1}^{n} \frac{p_i}{3} \left(\xi_i^2 + \xi_i \xi_{i-1} + \xi_{i-1}^2 \right)
\end{aligned}
$$

and

$$
E[D_u] = \frac{(q_1 - q_4)(q_3 - q_4)}{2(q_3 - q_1)} \sum_{i=1}^{n} \frac{p_i}{3} \left(\xi_i^2 + \xi_i \xi_{i-1} + \xi_{i-1}^2 \right)
$$

Alternately, in some cases, the n-th interval may be unbounded; that is, it contains all incidents which have length greater than ξ_{n-1}, and has the form (ξ_{n-1}, ∞). In this case, a uniform distribution cannot be used for this last category. A suitable substitute is the exponential distribution, because (1) it has unbounded support, (2) the probability of longer and longer incidents approaches zero asymptotically, (3) long duration incidents may be characterized by a relatively constant hazard rate, and (4) it is tractable and amenable to analysis.



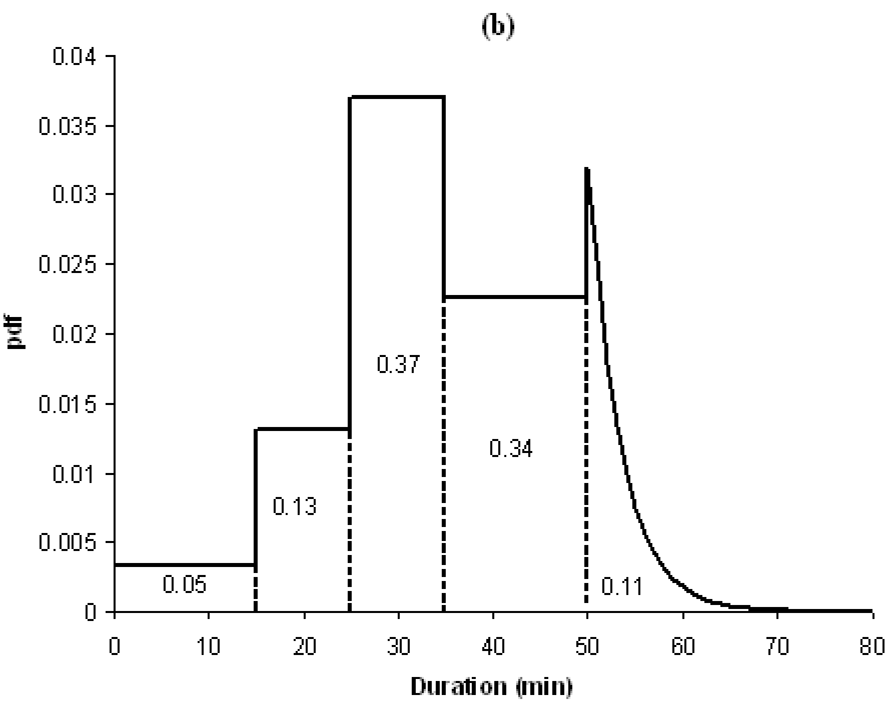


Figure 3.3: Choices of λ resulting in (a) continuous and (b) discontinuous densities.

We must choose an appropriate rate parameter λ for the exponential distribution; here we choose this such that the density function is continuous at ξ_{n-1} to avoid a sudden jump at this point (see Figure 3.3). Let $\bar{\tau} = \max\{\tau - \xi_{n-1}, 0\}$ represent the additional duration past ξ_{n-1}, so $\bar{\tau}$ has an exponential distribution, given $\tau > \xi_{n-1}$. To preserve continuity, we must choose $\lambda = f(\xi_{n-1}) = \dfrac{p_{n-1}}{\xi_{n-1} - \xi_{n-2}}$. Thus

$$E[\tau^2] = \sum_{i=1}^{n} \frac{p_i}{3}\left(\xi_i^2 + \xi_i \xi_{i-1} + \xi_{i-1}^2\right) + p_n E[\tau^2 \mid \tau > \xi_{n-1}]$$

To evaluate the latter term, realize that

$$\begin{aligned} E[\tau^2 \mid \tau > \xi_{n-1}] &= E\left[(\bar{\tau} + \xi_{n-1})^2 \mid \tau > \xi_{n-1}\right] \\ &= E\left[\overline{\tau^2} + 2\bar{\tau}\xi_{n-1} + \xi_{n-1}^2 \mid \tau > \xi_{n-1}\right] \\ &= \frac{2}{\lambda^2} + 2\frac{\xi_{n-1}}{\lambda} + \xi_{n-1}^2 \end{aligned}$$

using linearity of expected value, and well-known formulas for the first two moments of the exponential distribution. Substituting for λ and back into the previous equation, we obtain the final formula

$$E[\tau^2] = \sum_{i=1}^{n-1} \frac{p_i}{3}\left(\xi_i^2 + \xi_i \xi_{i-1} + \xi_{i-1}^2\right) + 2 p_n \left(\frac{(\xi_{n-1} - \xi_{n-2})^2}{p_{n-1}^2} + \frac{\xi_{n-1}(\xi_{n-1} - \xi_{n-2})}{p_{n-1}} + \frac{\xi_{n-1}^2}{2} \right)$$

Another common assumption is a lognormal distribution fitted using regression. Recall that the lognormal distribution is characterized by two parameters μ and σ^2, and its density function is

$$f(\tau) = \frac{1}{\sqrt{2\pi\sigma^2}\,\tau} e^{(\log\tau - \mu)^2 / 2\sigma^2}$$

for $\tau \geq 0$. Thus

$$\begin{aligned} E[D_u] &= E\left[\frac{1}{2}\tau^2 \frac{(q_1 - q_4)(q_3 - q_4)}{q_3 - q_1}\right] = \frac{(q_1 - q_4)(q_3 - q_4)}{2(q_3 - q_1)} E\left[\tau^2\right] \\ &= \frac{(q_1 - q_4)(q_3 - q_4)}{2(q_3 - q_1)} \left\{ Var[\tau] + (E[\tau])^2 \right\} \\ &= \frac{(q_1 - q_4)(q_3 - q_4)}{2(q_3 - q_1)} \left\{ e^{2\mu + \sigma^2}(e^{\sigma^2} - 1) + e^{2\mu + \sigma^2} \right\} \end{aligned}$$

using well-known values for the mean and variance of the lognormal distribution. Simplifying, we obtain

$$E[D_u] = \frac{(q_1 - q_4)(q_3 - q_4)}{2(q_3 - q_1)} e^{2(\mu + \sigma^2)}$$

4. Phase I: Which ramps should be closed?

Currently, decisions about ramp closure in the context of incident management are made on-site by emergency responders who are present, generally with safety as the primary goal. However, ramp closure can also be used to serve broader mobility improvement goals at the same time. A traffic management center (TMC) or similar facility is best suited to making decisions of the latter type due to their real-time knowledge of facility conditions across a larger area. This section describes the first part of a two-phase model for indicating how ramp closure should best be utilized in response to an incident (which has already been pre-screened using the methods in Chapters 2 and 3).

Using two distinct phases carries multiple advantages for incident response. First, closing ramps results in vehicles changing routes, which has impacts well beyond the immediate vicinity of the incident - the benefit to the freeway must be traded off against additional traffic on arterials and alternate routes. Each phase has a different geographic focus and uses different measures of effectiveness, ensuring that both freeway and arterial performance is accounted for. Second, time is of critical importance when managing incidents, and the sooner a response can be made, the more effective it will be. Therefore, the first phase, which selects the best ramps to close, is designed to run very quickly. The best times to reopen these ramps, which is given by the second phase, are not needed until somewhat later, so this phase chooses more detailed simulations than is possible within the strict time limit for the first.

The next section describes the assumptions we make about user behavior with ramp closure, followed by presentation of the first modeling phase.

4.1 Behavioral Assumptions

Currently, decisions about ramp closure in the context of incident management are made on-site by emergency responders who are present, generally with safety as the primary goal. However, ramp closure can also be used to serve broader mobility improvement goals at the same time. A traffic management center (TMC) or similar facility is best suited to making decisions of the latter type due to their real-time knowledge of facility conditions across a larger area. There are several ways to close access to ramps, including mechanized gates, VMSs, temporary barriers, and on-site personnel, but this is not the focus of this research. Rather, our intent here is to develop a methodological framework for quantifying the effects of ramp closure, to the end of deciding which ramps (if any) should be closed, and for how long. However, we do assume a "candidate list" of ramps where closure is feasible; the specific mechanism by which this closure occurs is not explicitly considered in this research. Still, one implementation-specific consideration is of prime relevance to this analysis, namely, the manner, locations, and time at which motorists are informed of the ramp closures, because this will determine where re-routing will occur. For instance, if motorists only learn of the ramp closure when approaching the onramp, such as from a VMS, they will follow their original route until reaching the ramp, at which point they must choose another, new route for the remainder of the journey. Alternately, if motorists learn of the ramp closure before departing on their trip, perhaps through an Internet site, entirely different routes may be chosen from the beginning of the trip. A still different route will be chosen if motorists learn of the closure after departing, but before reaching the onramp, as from a radio traffic report. For our analysis, we assume the first of these scenarios, that drivers are unaware of the ramp closure until arriving at the upstream end of the onramp. At this point, they learn of all of the ramps that are closed (so there is no chance of finding one's alternate route closed as well). One possible implementation that could convey such information is a VMS with a message such as that in Figure 4.1. Furthermore, all other drivers are unaware of the presence of the incident, so they will continue on their regular route unless they encounter one of these VMS signs when approaching an onramp (see Figure 4.2).

I-10 ACCIDENT ONRAMPS CLOSED	WYOMING AVE TO MISSOURI AVE

Figure 4.1: Example of two-phase VMS conveying the information assumed in this project.

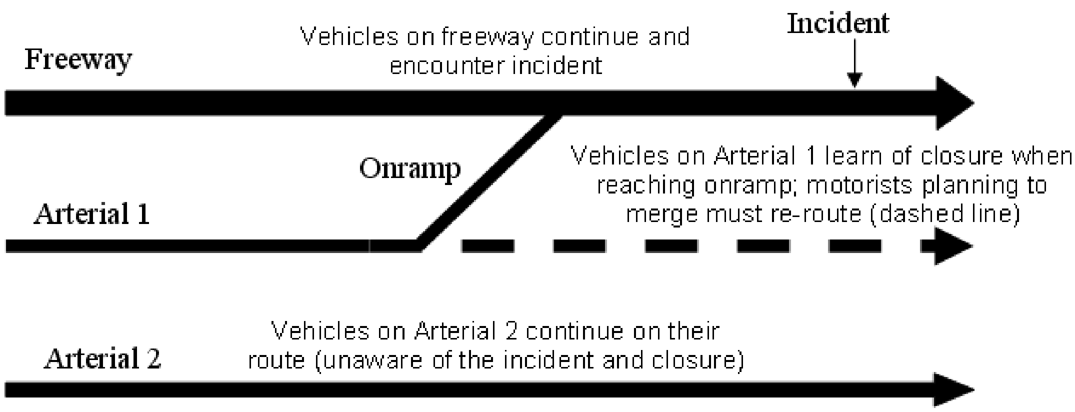


Figure 4.2: Diagram of rerouting behavior under information provision.

From a modeling standpoint, this information scenario was chosen because it minimizes the number of additional assumptions that need to be made. Allowing information to be learned at an earlier point in time requires describing driver behavior and individual preferences with far greater specificity, regarding response to information and online rerouting behavior. By restricting the information to a point at which there is no option - upon reaching the closed onramp, it is simply impossible to continue along the original route, and a new one must be chosen - we eliminate the need to introduce additional, poorly-understood behavioral parameters.

4.2 Macroscopic Model

In this section we describe the first phase, based on static traffic assignment methods used in transportation planning. This step is best directed towards the question of which ramps should be closed in response to an incident, because it can evaluate a large number of scenarios in a reasonable amount of time. As a result, this method proceeds by evaluating all possible combinations of open and closed ramps from a given candidate list (perhaps the four or five ramps upstream of the incident), allowing the best scenario to be chosen.

The general traffic assignment problem, on which this model is based, constitutes the final step of the canonical four-step planning process, and is concerned with identifying the routes chosen by all drivers in a large network, perhaps representing an entire metropolitan area. The underlying behavioral assumption, dating to Wardrop [26], is that all motorists are seeking the quickest route from the origin to the destination, accounting for congestion due to the anticipated behavior of others and for roadway properties such as capacity and speed limits. This mutually anticipatory behavior can be described as "user equilibrium" and typically assumes perfect information and deterministic conditions.

A solution to the static traffic assignment problem provides an "initial condition" describing equilibrium traffic flows, which will be modified by the presence of an incident and possible ramp closures. Specifically, an incident is modeled by adjusting the capacity parameter of the affected freeway segment (resulting in an increase in its travel time), and closing a ramp is modeled by deleting the ramp from the network.

As mentioned in Section 4.1, we assume that drivers are unaware of any ramp closures until they arrive at the upstream end of one of the closed ramps, at which point they learn of all of the closures. Note that the incident and the resulting ramp closure are unanticipated events, the equilibrium assumptions of determinism and complete information are violated: the resulting network state will not be an equilibrium.

With this information provision scenario, any driver whose regular route does not involve one of the closed ramps will keep the same route choice, as such drivers never even learn of the closure. On the other hand, any driver whose regular route includes one of the closed ramps must necessarily change routes. Because they learn of the closure only upon arriving at the ramp, the first part of their trip is the same. Once the closed ramps are learned, a new path must be chosen from their current location (the upstream end of a closed ramp) to the destination. In keeping with the assumptions that users are trying to minimize their travel times and have perfect information about the equilibrium (non-incident) conditions, we assume that they follow the shortest available path given the prevailing travel times from the equilibrium state. Finding these new paths can be performed efficiently using a one-to-all implementation of a shortest path algorithm, such as Dijkstra's [27]. Once this rerouting is accomplished, the resulting state can be analyzed to see the impact of closing a particular set of ramps.

To evaluate the performance of these scenarios, the most appropriate measure of effectiveness is total system travel time (TSTT), which is the total time spent by all motorists, regardless of origin or destination, traveling in the metropolitan area during the analysis period. Even though the ramp closure policy is enacted in a geographically small area, a global measure is the only way to fully account for all of the interdependencies and "ripple effects" generated by such a policy. TSTT accounts for any and all re-routing that occurs in response to ramp closure without having to specify the alternate routes a priori, and also accounts for the impact of these re-routed drivers on the alternate routes they traverse.

Thus, Phase I proceeds as follows: each possible ramp closure scenario (including closing no ramps) is considered; for each scenario, flows are rerouted as above, allowing that scenario's TSTT to be calculated. After evaluating all possible ramp closure scenarios, enact the one with the least TSTT, as this scenario leads to the lowest total travel time (or equivalently, the lowest average travel time). One possible implementation of this procedure is shown in Figure 4.3.

This method is efficient, because only one user equilibrium assignment needs to be performed to find a "base case." The rerouting procedure consumes comparatively little processor time, even when many scenarios are evaluated. Exact computation times can be found in Appendix C, where we apply this method to a sample network.

There are numerous static traffic assignment algorithms that have been developed. The first widespread solution method was based on the Frank-Wolfe convex programming algorithm [28]. Several more efficient algorithms have been created since then; those of

Jayakrishnan et al. [29], Bar-Gera [30], and Dial [31] are particularly noteworthy. For this application, the critical requirement is that the path flows are needed; that is, the number of motorists using every route in the network. Not all algorithms accomplish this directly: the Frank-Wolfe algorithm, for example, only returns the link flows (the number of motorists on each road segment), and the mapping between path and link flows is not one-to-one. Still, methods exist to infer path flows from link flows using the concept of entropy maximization (see, for instance, [32], [33], and [34]). Alternately, simple averaging of the path flows from each iteration of Frank-Wolfe can be used, although the resulting path flows do not maximize entropy in general.

I. **Initialize**. Find the user equilibrium link flows. Then, for each link L, each ramp r which is a candidate for closure, and for each destination d find the number of vehicles $x(L,d,r)$ *en route* to d which have previously used r.

II. **Iteration**. For each possible combination of opened and closed ramps:

1. **Find new paths for rerouted vehicles**. For each onramp r which is to be closed and for each destination d, find the shortest path from the upstream end of r to d.
2. **Remove rerouted vehicles from old paths**. For each arc L, if ramp r is closed, subtract $\sum_d x(L,r,d)$ vehicles from the flow on L.
3. **Add rerouted vehicles to new paths**. For every destination d, and for every arc on the shortest paths to d found in Step 1, add $\sum_r x(r,r,d)$ vehicles.
4. **Evaluation**. Calculate TSTT for new link flows.

III. **Output**. Report combination of closed ramps that minimizes TSTT.

Figure 4.3: Summary of Phase I.

4.3 Implementation Guidance

To summarize, the general framework for using a Phase I model is the following:

1. Perform a user equilibrium traffic assignment offline, noting the path flows and the usage of each candidate ramp by each destination.
2. When an incident occurs, perform the rerouting procedure described above for each combination of open and closed ramps.
3. Calculate total system travel time for each scenario, and choose the one minimizing total delay.

In implementing a Phase I model, the following should also be kept in mind:

- One needs the path flows from the traffic assignment results; either a path-based algorithm should be used for the assignment, or a most-likely path flows algorithm should be applied separately.
- Because accurate path flows are important, one should ensure that the traffic assignment is sufficiently converged (for further guidance on this, see [8]). The

time requirement for this should not be prohibitive because this assignment is done offline and the results stored for when an incident actually occurs.

- The number of combinations of open and closed ramps (and hence the number of scenarios that need to be tested) grows exponentially with the number of ramps that are candidates for closure. In our experience with a moderately large network (representing the city of El Paso), using four candidate ramps creates sixteen scenarios; evaluating these requires less than four seconds. However, using twice this many ramps would increase the computation time to approximately a minute, which is about as much as is practical (see Appendix C).

It may be that the model recommends a ramp closure which is counterintuitive (for instance, recommending closure of a ramp downstream of the incident). In this case, the model has simply identified an instance where ramp closure would be beneficial even in the absence of any incident, as a typical congestion relief measure. A specific policy to address such matters should be developed before implementation.

5. Phase II: How long should ramps be closed?

The first phase identifies the best combination of onramp to be closed in response to an incident on a freeway section, using TSTT as a measure of effectiveness. While this part runs quickly, it is unable to answer the question of how long ramps should be closed to optimize system performance.

Thus, we propose a second phase, using a dynamic model based on microsimulation. When beginning this phase, we take the set of ramps that should be closed as given, having been obtained from the first phase. Because Phase I is macroscopic and explicitly considers the impact of ramp closure on the entire network, and because Phase II takes the results of the Phase I as input, we restrict the scope of analysis to "zoom in" on the specific area of interest, as the broader impacts have already been accounted for.

Additionally, we assume that the incident duration is known, or can at least be estimated. Although there is a large amount of uncertainty in such predictions, a number of models exist to predict the clearance times of incidents (see Chapter 2). Still, these require some knowledge of incident characteristics (such as the number of vehicles involved, and whether emergency responders are present), which might be obtained from police dispatch reports, closed-circuit television cameras, or other sources. Because some time will be needed for a TMC to learn of the incident, and to enact the ramp closure strategy, we assume that it is desirable to close the onramps as soon as possible.

5.1 Duration Vector Search Procedure

With this information, the algorithm attempts to find the best length of time to close the ramps. Mathematically, if there are r ramps to be closed, we can express the closure durations as an r-dimensional vector **T**, and we seek the vector that optimizes facility performance. Note that the reduced geographic scope requires a different, less global measure of effectiveness than Phase I; several options are discussed below.

As there are infinitely many r-dimensional vectors, and as traffic systems lack provably "nice" mathematical properties such as linearity or continuity, we opt to discretize the possible ramp closure durations — for instance, by restricting our attention to ramp closure durations that are a multiple of 10 minutes — and search as many of these points as time allows. Graphically, we are reducing the search to a lattice of points, which we wish to traverse in an effective order (see Figure 5.1). We base this search on the following two principles:

- We anticipate that the ideal duration for all ramp closures is approximately the remaining incident duration.
- There is only a limited amount of time available for analysis before a decision must be made.

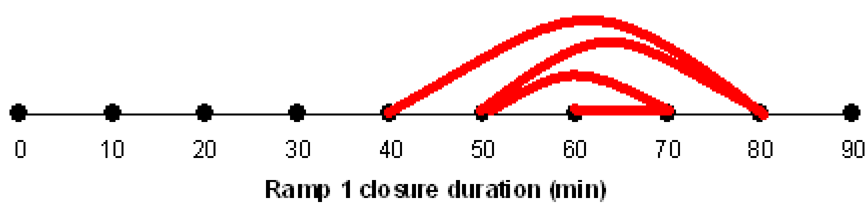
With these in mind, we begin the search by simulating the case where all ramps are closed for the entire duration of the incident; for instance, if the incident will be blocking lanes for 60 more minutes, we start by simulating all ramps being closed for 60 minutes. From here, the search "fans out" to nearby lattice points for additional simulations, as shown in Figure 5.1. This search continues for as long as is feasible, at which point the best set of durations is chosen and enacted. One way to implement this type of search is by using a FIFO queue data structure (see Figure 5.2 for pseudocode illustrating this procedure).

As mentioned above, TSTT cannot be used to evaluate a particular ramp closure scenario for Phase II, as it only models a small portion of the region. However, microsimulation allows more detailed measures of effectiveness based on fundamental traffic properties such as freeway speed and density. By using these, one can more fully predict the effect of ramp closure on the specific facility which is affected by the incident; the first phase can only do so approximately by reporting link volumes.

Microsimulation models typically require vehicle trajectories to be specified a priori; these should be obtained from Phase I above for consistency. If the choice of microsimulator makes this difficult, one may utilize a heuristic for rerouting vehicles, for instance, assuming all vehicles follow a frontage road or parallel facility and merge onto the freeway at the first open onramp, or by running one or two iterations of a built-in route finding feature of the simulator.

Additionally, because this phase is based on microsimulation, one can also evaluate a variety of other ramp treatment and incident management options, such as allowing HOV vehicle access to the closed ramps, or ramp metering.

(a) One ramp



(b) Two ramps

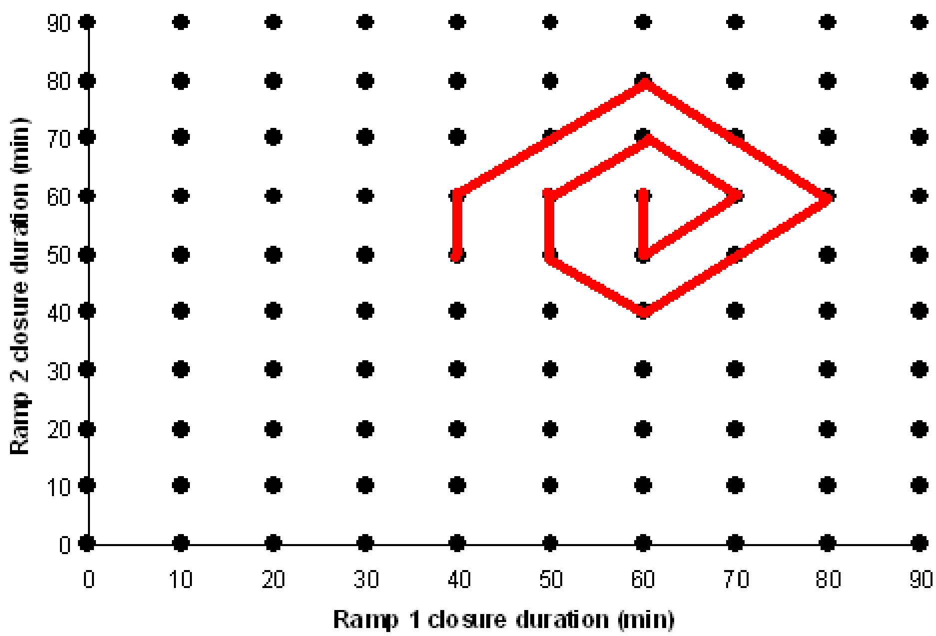


Figure 5.1: Possible duration vector search patterns

Input:

T : Maximum amount of time that can be used for analysis

D : Anticipated incident duration

Algorithm:

0. Create a new queue **[Trial mode]** and add the vector **[Trial mode]** to **[Trial mode]**

1. Remove the front element **v** from **[Trial mode]**.

2. Microsimulate the ramp closure for the durations **v**.

3. Consider every vector **w** which is adjacent to **v** (see note below). If **w** has never been in **[Trial mode]**, then add **w** to the end of **[Trial mode]**.

4. If elapsed time is less than **[Trial mode]**, return to step 1. Otherwise, terminate and output best duration vector.

Note: For example, if we are considering closure durations which are a multiple of 10 minutes, the vectors adjacent to [60 60] are [50 60], [70 60], [60 50], and [60 70].

Figure 5.2: Pseudocode for Phase II search procedure

5.2 VISSIM Implementation

The procedure described in the previous section is general in that any microsimulator can be used (assuming, of course, that it is capable of modeling incidents and temporary ramp closures). In our tests, we used the microsimulator VISSIM, developed by PTV. This choice was motivated by a desire to be consistent with the Phase I model by basing both on traffic assignment results. VISSIM readily interfaces with the planning software VISUM, which is capable of performing traffic assignment, allowing the incident corridor to be extracted easily and in a manner consistent with the macroscopic route flows.

Still, extracting a subnetwork requires more time than is available while an incident is in progress. Thus, in locations where ramp closure is being considered, appropriate subnetworks should be created and stored offline. The remainder of this section describes this network creation and simulation process in VISSIM.

5.2.1 Input Network

In order to dynamically model a ramp closure strategy we will use the VISSIM microsimulation software. In order to use the software we first must have a VISSIM network of the focus area. We can obtain this in a variety of ways:

From a VISUM network

One of the advantages of obtaining our VISSIM network from a VISUM network is that we can choose any subnetwork i.e. any possible incident could be modeled to any degree of focus. Another advantage of this method is that vehicles can be exported as both static routes or OD demands.

The disadvantage of this method, however, is that transforming a VISUM subnetwork to model the rerouting procedure (particularly the demands) is very time consuming, and very labor intensive task. As such, it becomes very difficult to do this on an incident by incident basis in real time. Furthermore, testing of every individual scenario, due to the amount of information included in each network, can take upwards of 15 minutes, which is unacceptable considering that we will usually have to test many different ramp closure scenarios.

Using VISSIM to create a network

The VISSIM package includes an easy to use graphic user interface (GUI), which can be used to create any particular network. Because creating a subnetwork for every possible focus area that one may need to analyze is infeasible, this method alone is also not appropriate. However, as it will be discussed in the implementation section, this method can be used for the creation of network "templates."

Using VISUM to create network and VISSIM to input routes

This is the option which we have determined to be the most efficient. In order to generate the full model, we use the VISUM network to generate the network, and using the VISSIM interface to generate the vehicle inputs and vehicle routes.

5.2.2 Generating Vehicle Inputs and Vehicle Routes

In order to appropriately load vehicles onto the network we need to define vehicle inputs and the routes that those vehicles will follow.

The reader is directed to the VISSIM manual for information on how to use the software package. The explanation that follows is meant for a person with VISSIM experience.

The objective of this step is to generate traffic on the network that matches that which you expect in real life. To achieve this, one can control the number of vehicles entering the network through each link (Vehicle Inputs), and the routing decisions that these vehicles make at different stages (Vehicle Routes). Both of these variables can be easily deduced from OD data. It is important to note that the information required should match the present state of the network to be useful. Generating the vehicle inputs in VISSIM is fairly straight forward. One can use the GUI to create different vehicle inputs, and assign any amount of flow to each one.

Generating the vehicle routes, on the other hand, is slightly more complicated. We will first need to define routes between every entry point and every exit point of the network (under the assumption that there are vehicles coming from every link going to every link). This is done in the form of a “routing decision” immediately following the vehicle input, and several routes that can be defined from the routing decision point. The vehicles will arrive at the decision point and will be divided according to the fractions assigned to each route, which are defined by the user. The fractions can be represented in any desired proportion, and need not add up to any particular number.

It is at this stage that we must choose the ratios appropriately in order to obtain as accurate of an approximation as possible to the real circumstances of the incident. Engineering judgment will probably have to be used to obtain appropriate results, as traffic conditions on a day to day, and more importantly, on an hour to hour basis can change radically.

Once the routing decision points and related routes have been defined, we will have to create a separate routing decision for each route that includes a ramp that will be closed. Because the fractions allocated to each route can be changed by time intervals, we can then change these fractions for the time interval in which the ramp will be closed by transferring flow from the route using the ramp to the route through which vehicles are expected to reroute themselves.

Changing the time intervals for which vehicles will be rerouted will represent the different durations for which the ramps will be closed.

5.2.3 Generating an Incident

Incidents in VISSIM are modeled as traffic signals, where the red cycle is the duration of the incident. We must make the overall cycle long enough so that the red cycle doesn't repeat.

An optional feature that can be included in order to make the model more realistic, are low speed areas. By defining low speed areas we can model the behavior of drivers approaching an incident area. If low speed areas are not defined, vehicles will be allowed to drive through the incident region at full speed, which is not desirable. Low speed areas can be created of any size and with any desired speed.

5.2.4 Evaluation

VISSIM has a wide variety of tools that can be used for obtaining information about any particular simulation. Three of these tools are of particular interest to planers:

- Travel Time Measurements Travel time segments can be defined in VISSIM between any two connected points in the network. The program will measure the travel time in every such segment during a defined time period, in regular intervals defined by the user. Both the compiled data (by time period) and the raw data (vehicle) can be obtained.
- Flow Measurements Using the sensors tool, the user can measure the number of vehicles that cross any such sensors. Like the travel time measurements, the flow measurements can be taken over a defined period of time in defined time intervals. This tool in combination with the travel time measurements allows us to estimate the overall effect of a ramp closure strategy, by estimating the total travel time of all vehicles of interest.
- Link Evaluation The link evaluation tool allows us to measure an extensive set of characteristics of a particular link. Of particular interest are density, average speed and flow. These measurements are very helpful in understanding the development of the incident.

What is particularly useful is that the link being evaluated can be subdivided into segments of equal length. By doing this we can track, for example, at what rate the queue is growing upstream of the incident, as well as quickly it is dissipating, and also in which way it is dissipating.

In order for VISSIM to calculate the measurements, they must be selected from the Evaluation/Files menu.

5.2.5 Practical Implementation

Because of the time and labor involved with creating the input network, generating the demands and routes, and testing the different ramp closure durations, it may be desired by the planner to have an easier way of determining the right strategy.

One possible way of simplifying the process is to create different network "templates." The idea is to have common highway setups stored as different VISSIM networks with routes and demands already set up. This eliminates the time involved in creating the network, which is significant, and provided that enough templates are created, a very wide array of incident locations can be modeled. One cannot expect, however, for a template to create as good of an approximation as the actual network, so there is an obvious tradeoff.

Another option for practical implementation is the use of a decision tree, in which different factors can be accounted for in order to determine the ramp closure strategy. Factors that will affect the decision include:

- Position of the incident with respect to adjacent on and off ramps
- LOS of the highway
- LOS of the adjacent arterials
- Capacity of adjacent arterials
- Expected duration of incident
- Severity of incident

As with the template approach, accuracy will be sacrificed (and thus VISUM-generated networks should be used for high-incident corridors), but increased simplicity makes it much easier to apply in any location.

6. Implementation Considerations

6.1 Introduction

While previous chapters focused on the modeling component of ramp closure for incident management, this section discusses several practical issues that will arise in any attempt to implement these models. In particular, we discuss specific means of closing a gate, and how this procedure might interface with TMC and on-site operations.

6.2 Access Control Options

Any sort of ramp closure policy, whether for incident management, congestion relief, reversing lanes, or inclement weather, requires some means to inform drivers of the closure and, ideally, to physically prevent entry onto the facility. A variety of means exist to accomplish this, but the most common is some type of gate that can be opened or closed as needed, although certainly other methods exist as well. Unsurprisingly, these methods vary in their cost, crashworthiness, effectiveness in preventing ramp entry, speed of response when it is decided that a ramp should be closed, and convenience and safety for the operations personnel who must open and close the ramps.

While a considerable amount of research has been performed in this area already, to our knowledge no study has considered what types of barrier is most effective for ramp closure as an incident management policy. Almost certainly, the best barriers for this purpose are different from those which best suit, say, closure due to poor weather conditions, for a variety of reasons. First, incident management is an inherently time-sensitive domain, and a speedy response can be critical for minimizing freeway delay. Second, some incident response personnel (either emergency responders or dedicated incident management teams) will probably be in the vicinity of the closed ramps. Third, this closure will generally be unexpected by the traveling public, no matter what efforts are expended to inform them in advance. Thus, this differs from regular ramp closure used for congestion management. Finally, it is not critical to prevent ramp entry for safety reasons, as it is in the case of reversible lanes where violators would drive into opposing traffic.

Thus, there is a need to consider the variety of options available specifically in the context of ramp closure due to incidents. This section first describes some of the many options available, along with cost estimates, and factors that either suggests that a particular ramp closure device is favorable or unfavorable for this project. Following this, a few case studies are described in detail, and a list of firms that manufacture barrier devices is given. Finally, we summarize the section by reiterating the key results.

6.2.1 Types of Barriers

Experience has shown that some type of physical barrier is necessary to prevent vehicles from entering a freeway; simply providing a sign is not enough to discourage violation. Physical barriers can be classified according to whether their operation is manual or automated; manual barriers can further be classified into portable and fixed categories. Finally, as a last resort, personnel can be used to block the ramp by performing on-site traffic control, although this is undesirable for a number of reasons described below.

Each of these types of barrier is discussed in turn, along with cost estimates; much of the information in this section is derived from the FHWA *Ramp Management and Control Handbook* [35], and two technical reports written for the South Dakota Department of Transportation (SDDOT) [36] and the Minnesota Department of Transportation (MnDOT) [37].

Portable Manual Barriers

Several types of portable barriers exist, including traffic cones, rods, barrels, or other movable gates, often made of plastic and anchored using some ballast material such as sand or gravel. Cones and rods are perhaps the easiest to move, although they are less visible and do not present as formidable a barrier as barrels or larger plastic gates. Common to all of these is the need to unload them from a truck or other vehicle and then install them, which may be time or labor-intensive. This is especially significant if one anticipates that a particular ramp will be closed frequently. Nevertheless, this is the least expensive method of closing a ramp. SDDOT estimates that the cost of the equipment necessary to close a ramp is roughly $2,500, and certainly this can be moved from ramp to ramp as necessary. However, this estimate excludes the labor and transportation costs needed to put these barriers in place. One type of portable manual barrier is depicted in Figure 6.1.

Figure 6.1: A portable manual barrier.

Fixed Manual Barriers

These barriers often take the form of gates which can swing either horizontally or vertically. This type of barrier is often used in cases of weather closures. As might be expected, this type of barrier is more practical to use if routine closures are expected for a particular ramp. Still, their operation requires personnel to drive to the ramp site and open or close the gate manually. This may be particularly difficult with horizontal swing gates during times of inclement weather, as snow may obstruct the path of the gate, or heavy wind may make it difficult to open or close. Also, there are generally greater safety risks involved, as these personnel may be exposed to high-speed traffic while operating these gates. An additional safety consideration is that this type of gate presents a permanent fixed-object safety hazard near this ramp, although this can be mitigated by proper placement and warning devices. At the same time, there are certain advantages to manual operation, as having a person on-site to open or close the gate provides greater assurance that the gate is properly opened or closed, and that malfunctions can be easier diagnosed on the spot.

A basic vertical swing gate costs approximately $15,000, although the total installation cost (including labor) is roughly twice this.

Figure 6.2: An automatic manual barrier.

Automatic Barriers

Automated barriers are similar to those described in the previous section, in that they usually take the form of horizontal or vertical swing gates which are permanently installed at a fixed location. The prime difference is seen in their ease of operation, as they can be controlled remotely. The most sophisticated (and most expensive) systems allow operation from a distant TMC using Internet-based communication protocols. Simpler installations allow operation from a remote-control device; with a good line of sight, some remote control devices can function up to distances of two or three miles. Although automatic gates are typically safer because gate operators are not as exposed to traffic, diagnosis and repair of a malfunctioning gate may take additional time as the appropriate personnel must be dispatched to that location. Alternately, successful operation of the gate can be confirmed by closed-circuit television (CCTV) cameras in the area.

There is a wide range in price estimates for automatic gates, depending on the desired degree of automation, type of control, and existing long-range communications infrastructure; SDDOT identified the cost of various installations as ranging from $34,000 to $200,000. MnDOT estimates the cost of the gate itself as between $8,000 and $16,000, with the supporting communication equipment costing an additional $6,000 to $45,000.

On-Site Personnel

As a stopgap measure, ramps can be closed by physically placing police or agency personnel in the ramp and guiding traffic to alternate routes. Although this method requires no fixed gate installation or prior investment in ramp closure mechanisms, using personnel in this manner should only be done as a last resort or as a stopgap measure, for several reasons.

First, it is unsafe and exposes personnel to great risks, especially at high-speed ramps. Second, it is an inefficient use of these personnel, whose specific training can almost certainly be put to better use, especially when there is a major incident in the vicinity. Third, dispatching personnel to each location that needs to be closed requires considerable time compared to automated gate operation.

Nevertheless, this option always remains, for instance, if a severe incident occurs in an area where no fixed gates are installed, and there are no available portable gates.

6.2.2 Case Studies

Multiple agencies have installed gates to close ramps in their jurisdictions. These installations represent a variety of gate types and a variety of purposes for ramp closure. Particular attention is paid to studies performed by MnDOT and SDDOT, as these agencies have performed detailed evaluations of the available options, and fully documented their decision-making process in several technical reports [36] [37] [38] [39]. Following these, a brief description of current ramp closure operations in the Seattle metropolitan area is discussed as a third case study.

Minnesota

In 2001, the Minnesota Department of Transportation investigated options to replace a fixed manual gate with some type of automated barrier at the I-90/US-71 interchange near Jackson, Minnesota, ideally controllable from an MnDOT office eighteen miles away. The primary function of this gate is to close this onramp in cases of inclement weather.

Several options were considered, including a conventional automated swing gate; a series of signs which informed drivers of the closure but provide no physical barrier to entry; a "virtual" gate using laser or holographic technology to project an image of a barrier without physically preventing entry; inflatable gates or nets that present a lesser safety risk should a driver collide with the barrier; and on-site personnel. Of these, the latter was immediately ruled out as infeasible. Virtual gates were also ruled out as cost-prohibitive and inflatable gates were found to be unreliable in the poor weather conditions when the ramp would be closed.

After further considerations, a conventional automated gate was selected as the preferred alternative, with communication provided by wireless technology. Several other communication options were discussed, including fiber optic cable, conventional telephone lines, and long-haul microwave transmission. A secondary recommendation was a "soft closure" strategy where variable message signs announce the ramp closure and violation penalties without presenting a physical obstruction to drivers.

South Dakota

Like Minnesota, South Dakota can experience winter weather severe enough to warrant freeway closure, and a system of freeway gates is in place to control access in these cases. On average, each gate is operated once every two to three years; thus they do not anticipate that these gates will be used frequently.

As SDDOT was considering the replacement of manual swing gates with automated ones, safety was a prime consideration. In particular, SDDOT was interested in the crashworthiness of different gate types, and specific recommendations were also made to improve safety for existing manual gate operation.

In the end, SDDOT decided to pursue a policy of replacing manual gates on a case-by-case basis, stating that the variety of gate types and the variability in the associated costs prohibits drawing any conclusions that can be applied statewide.

Seattle

The Washington State Department of Transportation operates gates on reversible lanes located on I-5 and I-90 in the Seattle metropolitan area. These gates control access to the reversible lanes from both the mainline and several on- and offramps depending on the direction of flow. Horizontal swing gates are used in all cases, augmented by variable message signs (VMS) announcing whether a particular ramp is open or closed.

These gates must be operated by an on-site employee, although limited automation eliminates the need for this employee to step into the roadway to close the gate. When an operator opens or closes a gate, the appropriate VMSs are contacted and switch messages to represent the new state. When possible, the Seattle traffic management center verifies that the VMSs have successfully changed using closed circuit television cameras.

In this case, verification is extremely important because a vehicle entering a closed ramp would enter a traffic stream moving in the opposite direction. Thus, multiple barriers are often employed in combination with a net at the highest-volume ramps, and a one-hour closure of the reversible lanes is provided whenever the direction changes. This hour allows time for the operator to change all of the gates and verify that the lanes are vacated before opening them in the opposite direction.

6.2.3 List of Firms Providing Gate Technology

This section provides a partial list of companies that provide barrier gate technologies to state agencies, or have done so in the past. Some agencies also choose to manufacture gates in-house. This list is not exclusive, and no recommendation is made to TxDOT as to the merits of any firm on this list.

- AMTECH
- B&B Gates & Barriers
- Eagle Traffic Control Systems
- Hy-Security Gate & Barriers
- Magnetic Automation Corp.
- New York Gate Manufacturing
- Roadway Manufacturing
- SAFETRAN Systems
- WAECorp
- Winter/Alpine Engineering Corp.

6.2.4 Summary of Gate Operation Requirements

The variety of methods for closing a gate indicates that any particular installation must be tailored to the specific application. In the context of ramp closure as an incident management strategy, several factors are apparent. First, for this purpose, the primary intent ramp closure is to improve operations, not to prevent entry due to safety reasons. Unlike ramp closure for inclement weather or for reversible lanes, an errant vehicle entering the freeway does not pose a significant safety risk. Second, incident management requires a swift response for maximum effectiveness. Third, it is impossible to predict when and where an incident is going to occur. Thus any ramp closure decision will likely be unexpected by the public. Successful implementation of ramp closure for incident management should take all of these into account when choosing the best method for closing the ramp.

6.3 Interaction with TMC and On-Site Operations

While the focus of this project has been on the methodological development needed to enact effective ramp closure policies, this can only be done by considering the context in which implementation may eventually occur. As with any incident management policy, this will require the cooperation of TMC staff and on-site responders. In particular, as these on-site responders are currently responsible for making all ramp closure decisions, it is important to incorporate them and their experience into any policies which are developed. This section describes one possible framework for putting these models into use.

Recall that the very first step is to prescreen incidents by predicting their severity to see if closure is warranted at all. This requires information about the incident; fortunately, this information is already collected by many TMCs as part of their routine operations (obtained from dispatchers or visually from CCTV cameras), and some TMCs train operators to enter this data into computer systems as it is received or updated. This information could then be input into NBIDS or a similar system very easily, possibly even completely transparently so that operators do not need to perform any additional tasks to do this. Once the incident duration has been predicted, it would again be a simple matter to combine this result with real-time traffic counts from loop detectors to predict incident delay.

If the incident delay estimate indicates that ramp closure should be considered, the Phase I/Phase II models should be run. (An alternate form of prescreening can occur if an on-site responder feels that ramp closure would be beneficial; such a call to the TMC could also trigger execution of these models.) At this point, an operator would run a computer program to conduct the Phase I analysis, selecting several ramps in the vicinity of the incident for possible closure. This phase should require less than ten seconds for networks of typical size, allowing a swift response. Typically, a response is needed within approximately sixty seconds for the result to be of much use, especially if this analysis is performed at the request of an on-site responder.

If closure is warranted, it should begin immediately. The specific details of closure depend on the method of access control (Section 1) — automated gates should be supervised by CCTV to ensure safety, and manual gates will temporarily require diversion of an on-site responder from the incident scene itself, or the dispatching of another responder to close the ramp. These considerations should be accounted for when enacting any ramp closures.

Simultaneously, operators will begin the Phase II analysis. Depending on the level of automation of this phase and the availability of a suitable subnetwork, this may be as simple as entering the incident's duration, and allowing an algorithm to perform the simulations automatically. Several simulations will be run (the exact number depending on the available time for this phase and the time required for the simulator to evaluate a given scenario), and the optimal closure durations relayed to the on-site responders. At the same time, this latter decision of when to reopen the ramps should ultimately be controlled by the on-site responders, who have the best indication of when the incident will actually clear.

One additional caveat bears mentioning, namely, that ramp closure may make it more difficult for emergency vehicles to access the incident site, especially if a large number of ramps are being closed. While this may be offset somewhat by improved freeway operations, this consideration should be kept in mind when choosing the optimal ramp closure policy and the type of gate or barriers that will actually control ramp access.

6.4 Combination with Other Ramp Treatment Strategies

Certainly complete ramp closure is not the only ramp treatment strategy that can be applied in incident management. In particular, ramp metering can be considered as a milder form of closure: instead of closing the ramp entirely, metering restricts the flow of vehicles merging onto the freeway through the use of a signalization device. Ramp metering has been applied in many areas throughout the country.

The interaction between ramp closure and ramp metering has not been explicitly considered in previous literature; however, many adaptive ramp metering algorithms (such as FUZY [40], used by the Washington State Department of Transportation in the Seattle area) are designed to respond to changing conditions in real time without requiring the constant oversight of an operator, and will adjust access rates based on freeway conditions and queue length.

Another option is to allow freeway access to certain vehicles, such as high occupancy vehicles (HOVs), transit, or emergency vehicles. The advantages of allowing transit vehicles to follow the predetermined route, or of allowing emergency access, should be clear. Allowing HOV access is less clear; in particular, because these drivers know that there is an accident downstream, many of these may choose a different route even if not denied access explicitly. Additionally, allowing some vehicles access may create enforcement difficulties with unauthorized vehicles bypassing the gate in the restricted lane.

Evaluating these options should be performed in Phase II in conjunction with the dynamic analysis, as microsimulators can typically model these features. For instance, in VISSIM, a traffic signal can be used to represent a ramp meter, and to control access to the freeway dynamically.

Several of these options were tested in combination, applied to a fictitious hour-long incident on a ten-lane freeway facility. A three-hour analysis period was used, with the first hour a “warm up” intended to allow the network to reach a steady-state condition. At this point, a severe incident occurs, blocking three of the five lanes in one direction, which lasts for an additional hour. Freeway density was compared for each of these scenarios, as shown in Figure 6.3.

From these simulations, it appears that ramp metering does not substantially affect freeway density while the incident is in progress, but leads to a significant reduction in congestion after the incident clears, effectively speeding the recovery process. This is logical, because ramp metering is most effective when merging traffic, and not a downstream bottleneck, is the primary cause of congestion.

Even allowing a generous proportion of HOVs (15%), allowing HOV access does not appear to affect freeway density; thus, based on this initial analysis, any benefit from allowing

HOV access would probably be outweighed by the additional enforcement difficulties such a policy would create.

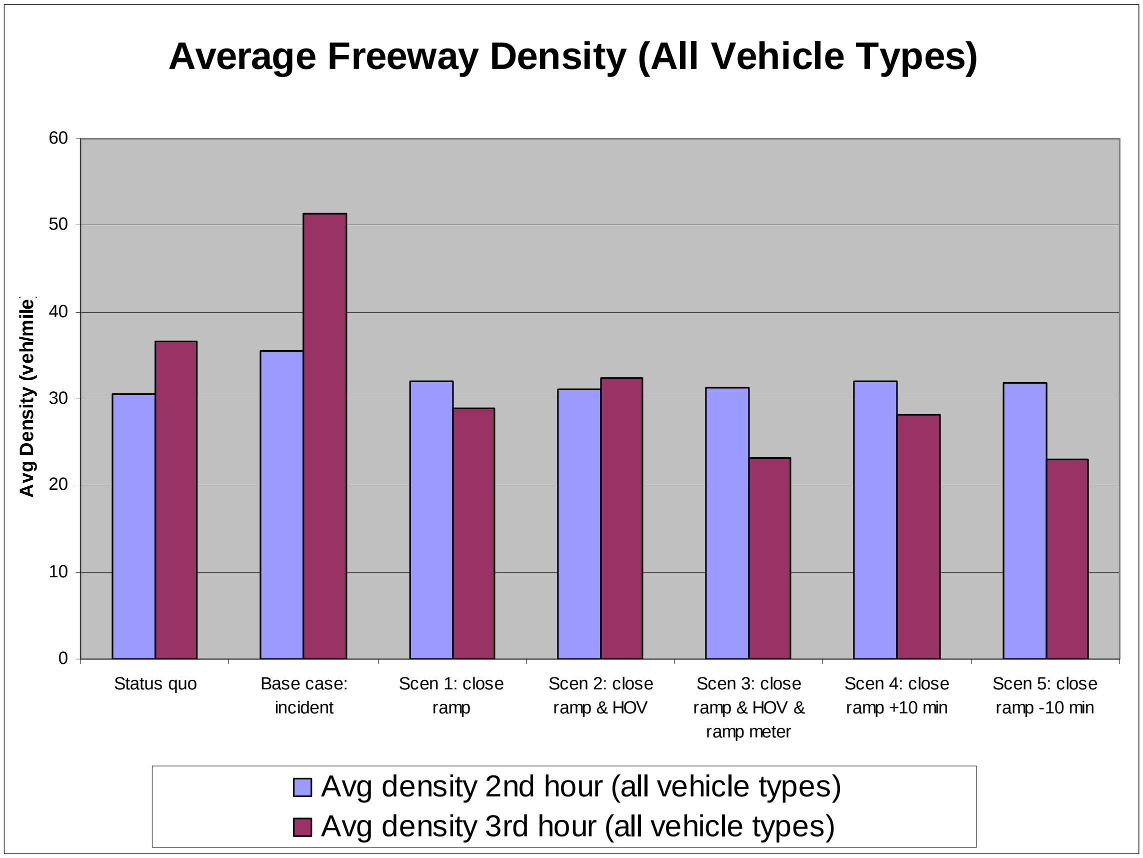


Figure 6.3: Comparing ramp metering and HOV ramp treatment strategies.

7. Summary and Conclusions

In this research, we have developed a three-step process for answering the basic questions related to ramp closure as an incident management policy:

1. Is the incident severe enough to consider ramp closure?
2. If so, *which* ramps (if any) should be closed?
3. If any ramps are closed, for *how long* should they remain closed?

Towards the first end, we developed novel incident duration and delay prediction models that can account for the inherent uncertainty and limited information in incident management; towards the second, we developed a Phase I model built on traffic equilibrium principles. Combined, these first two steps require less than thirty seconds, allowing a rapid response. The last question is addressed by a Phase II model employing microsimulation. We also discuss several options for actually closing the gates, and propose a method of integrating these models into an actual incident management plan.

Throughout this process, it is important to include the judgment and experience of trained TMC operators and on-site responders in this process, because no computer model or formula is sophisticated enough to replace this. Nevertheless, these models can supplement human knowledge and experience, leading to improved incident management and, in the end, to improved freeway performance.

References

[1] Lindley, J. (1987) Urban freeway congestion: quantification of the problem and effectiveness of potential solutions. *ITE Journal* 57, 27-32.

[2] Miesse, C. C. (1967) Optimization of single-lane freeway traffic flow by elective ramp closure. *Transportation Research* 1, 157-163.

[3] Gervais, E. F., and W. Roth. (1966) An Evaluation of Freeway Ramp Closure. Traffic Division, Michigan State Highway Department.

[4] Neudorff, L. G., J. E. Randall, R. Reiss, and R. Gordon. (1997) Freeway Management and Operations Handbook, Federal Highway Administration.

[5] Prevedorous, P. D. (2001) Freeway ramp closure: simulation, experimentation, evaluation and preparation for deployment. *ITE Journal* 71, 40-44.

[6] Khattak, A., J. Schofer, and M. Wang. (1995) A simple time-sequential procedure for predicting freeway incident duration. *IVHS Journal* 2, 113-138.

[7] Garib, A., A. E. Radwan, and H. Al-Deek. (1997) Estimating magnitude and duration of incident delays. *Journal of Transportation Engineering* 123, 459-466.

[8] Ozbay, K. and P. Kachroo. (1999) *Incident Management in Intelligent Transportation Systems*. Artech House, Boston.

[9] Jones, B., L. Janssen, and F. Mannering. (1991) Analysis of the frequency and duration of freeway accidents in Seattle. *Accident Analysis and Prevention* 23, 239-255.

[10] Giuliano, G. (1989) Incident characteristics, frequency, and duration on a high volume urban freeway. *Transportation Research A* 23, 387-396.

[11] Golob, T., W. Recker, and J. Leonard. (1987) An analysis of the severity and incident duration of truck-involved freeway accidents. *Accident Analysis and Prevention* 19, 375-395.

[12] Sullivan, E. (1997) New model for predicting freeway incidents and incident delays. *ASCE Journal of Transportation Engineering* 123, 267-275.

[13] Nam, D. and F. Mannering. (2000) An exploratory hazard-based analysis of highway incident duration. *Transportation Research A* 34, 85-102.

[14] Smith, K. and B. Smith. (2002) Forecasting the clearance time of freeway accidents. Publication STL-2001-012. Center for Transportation Studies, University of Virginia.

[15] Ozbay, K. and N. Noyan. (2006) Estimation of incident clearance times using Bayesian networks approach. *Accident Analysis and Prevention* 38, 542-555.

[16] Maron, M. (1961) Automatic indexing: an experimental inquiry. *Journal of the ACM* 8, 404-417.

[17] Kononenko, I. (1993) Inductive and Bayesian learning in medical diagnosis. *Applied Artificial Intelligence* 7, 317-337.

[18] Powers, R., M. Goldszmidt, and I. Cohen. (2005) Short term performance forecasting in enterprise systems. Publication HPL-2005-50, HP Labs.

[19] Domingos, P. and M. Pazzani. (1997) On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29, 103-130.

[20] Wirasinghe, S. C. (1978) Determination of traffic delays from shock-wave analysis. *Transportation Research* 12, 343-348.

[21] Morales, J. (1986) Analytical procedures for estimating freeway traffic congestion. *Public Roads* 50, 55-61.

[22] Lighthill, M. J., and G. B. Whitham. (1955) On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society A* 229, 314-345.

[23] Al-Deek, H., A. Garib, and A. E. Radwan. (1995) New method for estimating freeway incident congestion. *Transportation Research Record* 1494, 30-39.

[24] Bhat, C. R. (1994) Imputing a continuous income variable from grouped and missing income observations. *Economics Letters*, 46, 311-319.

[25] Jensen, J. L. W. V. (1906) Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica* 30, 175-193.

[26] Wardrop, J. (1952) Some theoretical aspects of road traffic research. *Proceedings of the Institute of Civil Engineers, Part II*, 325-378.

[27] Ahuja, R., T. Magnanti, and J. Orlin. (1993) *Network Flows*. Prentice-Hall, Englewood, NJ.

[28] Frank, M., and P. Wolfe. (1956) An algorithm for quadratic programming. *Naval Research Logistics Quarterly* 3, 95-110

[29] Jayakrishnan, R., W. Tsai, J. Prashker, and S. Rajadhyaksha. (1994) A faster path-based method for traffic assignment. *Transportation Research Record* 1443, 75-83.

[30] Bar-Gera, H. (2002) Origin-based algorithm for the transportation assignment problem. *Transportation Science* 36, 398-417.

[31] Dial, R. (2006) A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research B* 40, 917-936.

[32] Rossi, T. F., S. McNeil, and C. Hendrickson. (1989) Entropy model for consistent impact fee assessment. *Journal of Urban Planning and Development* 115, 51-63.

[33] Bell, M. G. H., and Y. Iida. (1997) *Transportation Network Analysis*. Wiley, New York.

[34] Bar-Gera, H. (2006) Primal method for determining the most likely route flows in large road networks. *Transportation Science* 40, 269-286.

[35] Federal Highway Administration. (2006) *Ramp Management and Control Handbook.*

[36] Russell, S. (2000) *Automated Road Closure Gate*. Technical report SD2000-11, prepared for the South Dakota Department of Transportation.

[37] ThomTech Design, Inc. (2002) *I-90 Gate Operations Project*, prepared for the Minnesota Department of Transportation.

[38] ThomTech Design, Inc. (2001) *I-90 Gate Operations System Research Report,* prepared for the Minnesota Department of Transportation.

[39] Johnson, D. (2003) *Automated Road Closure Gate Needs Assessment and I-90, Exit 67 Electric Actuator Project*. Technical report SD2001-08, prepared for the South Dakota Department of Transportation.

[40] Taylor, C. and D. Meldrum. (2000) *A Programmer's Guide to the Fuzzy Logic Ramp Metering Algorithm: Software Design, Integration, Testing, and Evaluation*. Technical report WA-RD 481.3, prepared for the Washington State Department of Transportation.

[41] Boyles, S., D. Fajardo, and S. T. Waller. (2007) A naïve Bayesian classifier for incident duration prediction. Presented at the 86th Annual Meeting of the Transportation Research Board, Washington, DC.

[42] Daganzo, C. (1994) The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research B* 28, 269-287.

[43] Box, G. E. P. and M. E. Muller. (1958) A note on the generation of random normal deviates. *Annals of Mathematical Statistics* 29, 610-611.

Appendix A: NBIDS Documentation

A.1 Introduction

Incidents cause tremendous delay in urban areas throughout the nation. In fact, some studies indicate that this type of non-recurring event creates the majority of congestion on freeways [1]. As such, effective incident management is crucial for mitigating the impact of these unanticipated events, and for reducing congestion and its associated personal, economic, and environmental costs.

Accordingly, effective incident management techniques have the potential to substantially reduce congestion. Some of these techniques, such as dedicated incident response vehicles to assist disabled vehicles, or the use of variable message signs and highway advisory radio to inform travelers of incidents, can be used in response to nearly all incidents, regardless of severity. However, other techniques, such as the temporary lifting of high-occupancy vehicle restrictions on lanes to increase capacity, or the closing of upstream freeway onramps, involve more operational overhead and are only appropriate for incidents which are expected to have a severe impact on the system.

The key to deciding when to enact this latter class of strategies is to predict how severe an incident is while it is still in progress. Ideally, this is done as soon as possible. However, two complicating factors make this somewhat more difficult: first, the incident management process is characterized by uncertainty, and may behave unexpectedly. Second, new or updated information is received at different points in time (such as by updated dispatch information when an emergency responder arrives). This latter fact is in direct conflict with the goal of making a decision as early as possible.

Existing methods for predicting incident severity in real time tend to either ignore both of these factors for the sake of modeling simplicity, or account for them at the cost of extensive calibration and additional assumptions (for a review of such methods, see Section A.3). By using a naive Bayesian classifier, described in Section A.2, NBIDS incorporates uncertainty and limited information into the prediction process while eschewing unnecessary complications sometimes seen in other classifiers.

NBIDS is designed to predict incident duration (one common measure of severity) in real-time, using historical incident data with a minimum of calibration effort. Section A.2 guides the reader through the process of creating and initializing an incident database and then using this to predict incident durations, using a small example data set. Section A.3 describes the naive Bayesian classifier, starting with a brief review of relevant concepts from probability. Sections A.4, A.5, and A.6 respectively describe the three steps of creating a database, initializing the database, and predicting incident duration in greater detail.

NBIDS was written using FreeBASIC, an open-source BASIC compiler. All source code for this software is provided in Section A.7.

A.2 Tutorial

This section guides the reader through the entire process of using this software. Predicting incident duration using NBIDS consists of three steps:

Step 1. Create comma-separated incident database.

Step 2. Use NBIDS to initialize the database.

Step 3. Predict incident duration.

Steps 1 and 2 only need to be performed when using a new incident database. Once the database has been created and initialized, incident duration prediction (Step 3) can be performed without repeating the first two steps. However, Steps 1 and 2 do need to be repeated whenever the database is updated or changes.

In this example, we consider a small database of only ten incidents. This is far too small to use for real prediction, but illustrates the operation of this software nevertheless. Specific instructions for the tutorial are printed in **boldface**, interspersed with brief descriptions of how the program works.

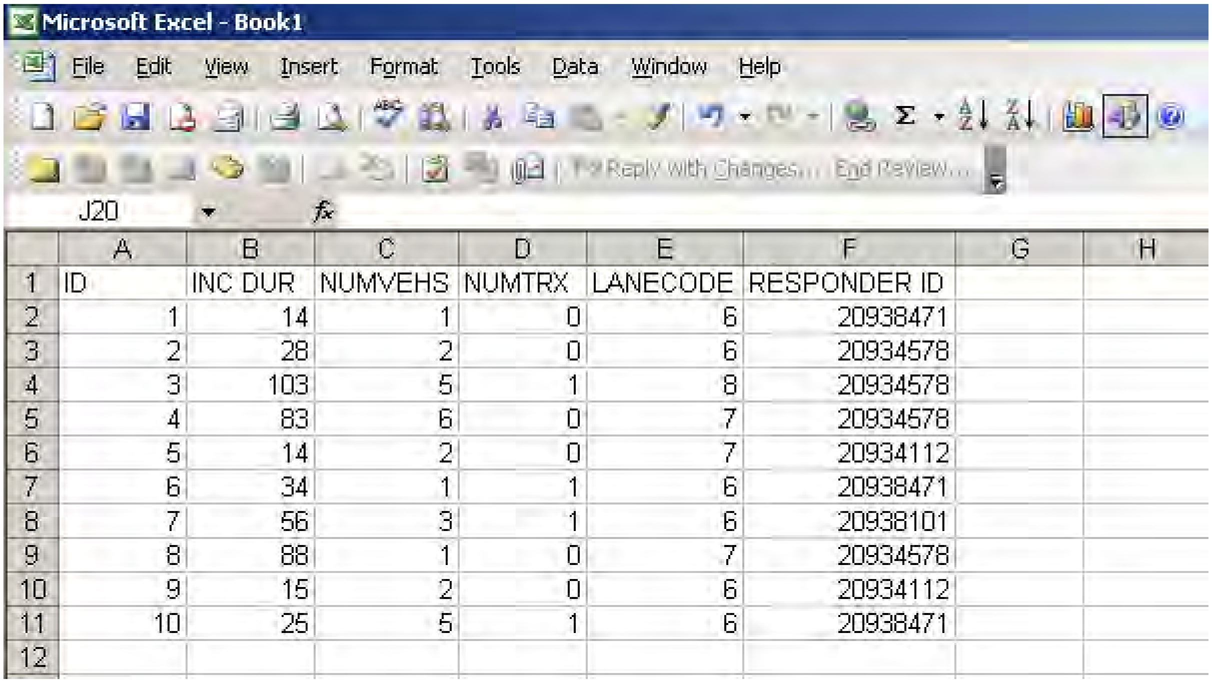
To begin, NBIDS must be installed on the computer. This process is very easy: simply create a folder for NBIDS, and copy the `nbids.exe` file from the CD-ROM. All incident databases and related files described in this manual (or a copy of them) should be stored in this folder for NBIDS to use.

Step 1: Create comma-separated incident database

The initial step involves creating the text file containing the historical incident data which will be used to calibrate the Bayesian classifier. For NBIDS, this data consists of a table of incidents, their durations, and their characteristics, along with a header row to identify each column. The easiest way to prepare this data is to use a spreadsheet or database program; in this example, we use Microsoft Excel®.

For actual applications, there should be a large number of incidents included in the database, but for this tutorial, we will only use ten. For each incident, the information we have is the incident's duration, the total number of vehicles involved, the number of heavy trucks involved, a numeric code representing the lane that was blocked, and an identification number of the emergency responder who arrived on scene.

Load Excel®, create a new spreadsheet, and enter the data shown in Figure A.1:



	A	B	C	D	E	F	G	H
1	ID	INC DUR	NUMVEHS	NUMTRX	LANECODE	RESPONDER ID		
2	1	14	1	0	6	20938471		
3	2	28	2	0	6	20934578		
4	3	103	5	1	8	20934578		
5	4	83	6	0	7	20934578		
6	5	14	2	0	7	20934112		
7	6	34	1	1	6	20938471		
8	7	56	3	1	6	20938101		
9	8	88	1	0	7	20934578		
10	9	15	2	0	6	20934112		
11	10	25	5	1	6	20938471		
12								

Figure A.1. Incident data for tutorial.

For NBIDS to read the data, it must be saved in a comma-separated text format. In this format, the above data looks like this:

```
ID,INC DUR,NUMVEHS,NUMTRX,LANECODE,RESPONDER ID
1,14,1,0,6,20938471
2,28,2,0,6,20934578
3,103,5,1,8,20934578
4,83,6,0,7,20934578
5,14,2,0,7,20934112
6,34,1,1,6,20938471
7,56,3,1,6,20938101
8,88,1,0,7,20934578
9,15,2,0,6,20934112
10,25,5,1,6,20938471
```

To convert the data into this form, choose ‘Save As...’ from the ‘File’ menu. In the ‘Save as type’ drop-down menu at the bottom of the dialog box, scroll down to ‘CSV (comma-delimited) (*.csv)’, about one third of the way down. In the ‘File name’ box, type ‘incidents.csv’ (Figure A.2). This saves the data in the proper format.

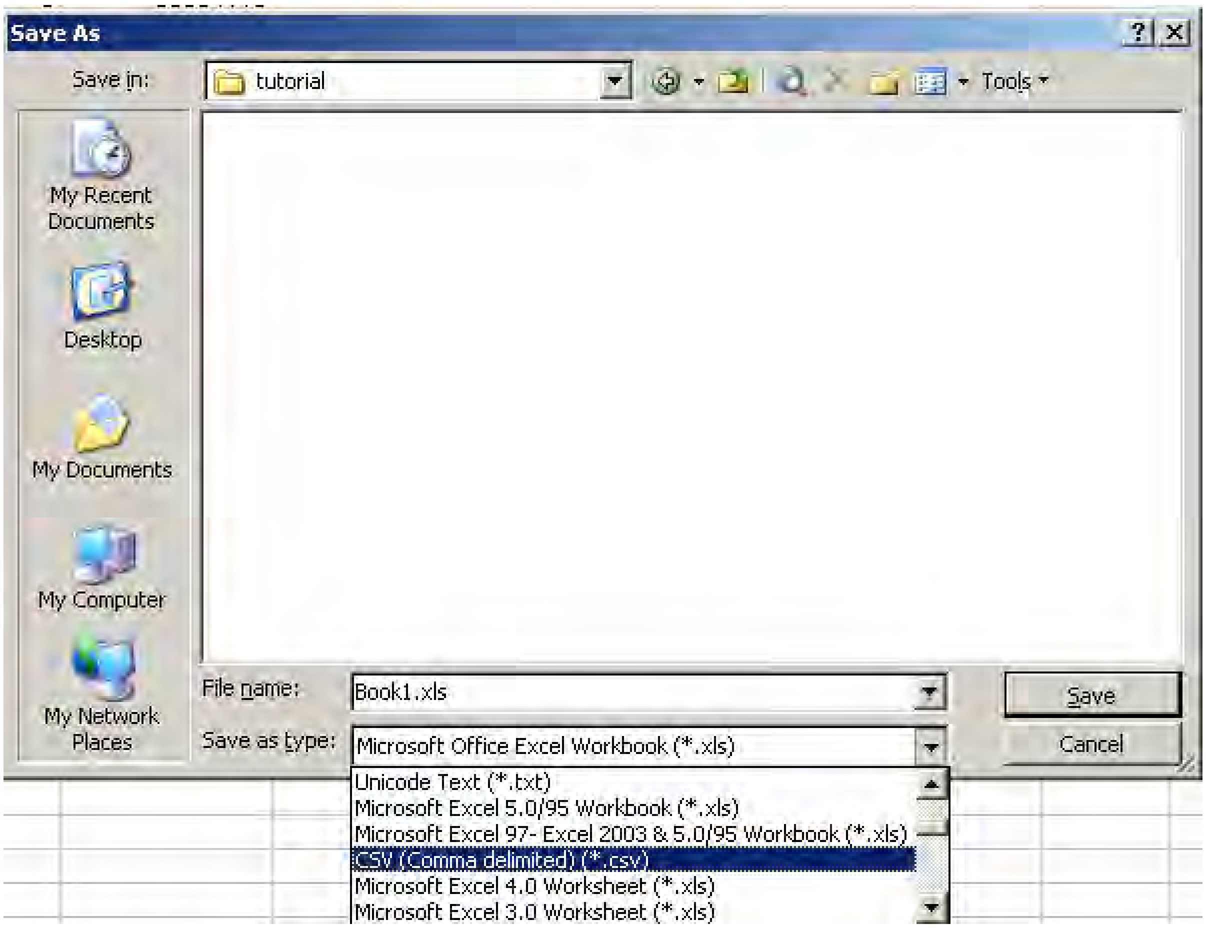


Figure A.2. Saving the data as a comma-separated text file.

Step 2: Use NBIDS to initialize the database

Once we've successfully created the database, the next step is to use NBIDS to generate the additional information needed to start predicting incident duration. First, we need to identify which column contains the incident durations (in this case, INC DUR). Second, not all of the information in the database is relevant; for instance, the incident ID does not affect how long a particular incident lasts. Likewise, the ID of the emergency responder probably has only slight relation to the incident's length. In this tutorial, we assume that we're only interested in using the number of vehicles (NUMVEHS) and the number of semitrailer trucks (NUMTRX) to predict incident duration. Selecting these fields is the first step in the initialization process.

Start NBIDS by double-clicking its icon. At the menu, press '1' and type the name of the incident database ('incidents.csv'). (Figure A.3). NBIDS will now display all of the column titles, according to the header row, and prompt you to select the column representing incident duration. **Enter 'INC DUR' when asked to enter the title of the duration field.** Now, you enter the names of the columns which will be used to predict incident duration. **Type 'NUMVEHS' and press [ENTER]; type 'NUMTRX' and press [ENTER]; and press [ENTER] again (a blank entry) to indicate that you're done.** (Figure A.4).

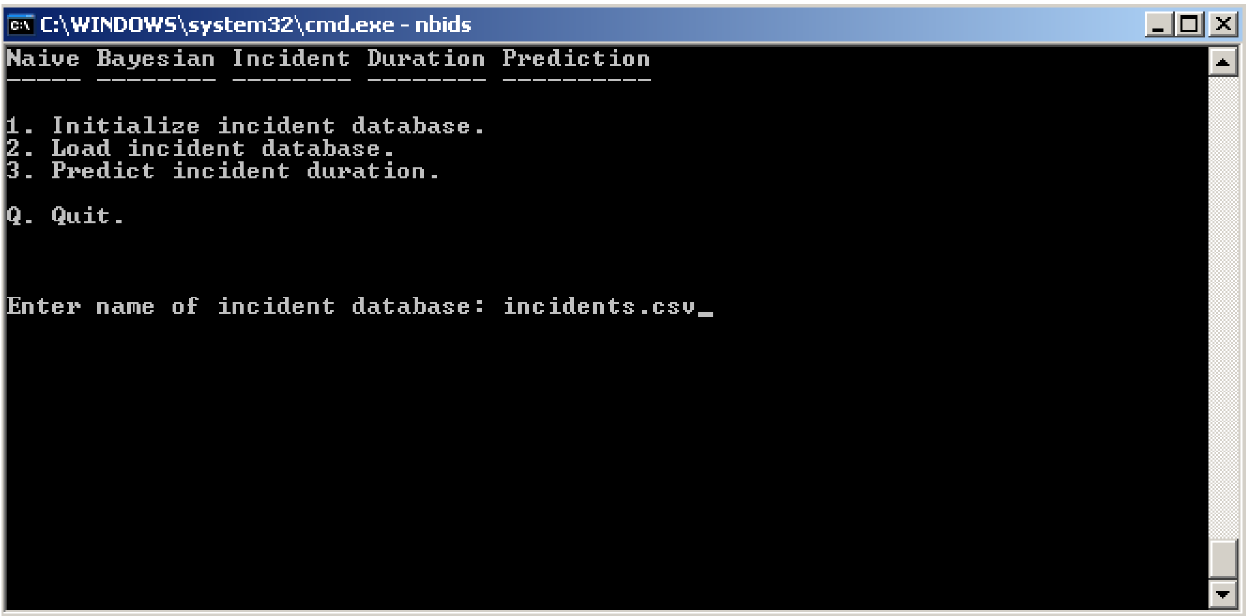


Figure A.3. Selecting the incident database for initialization.

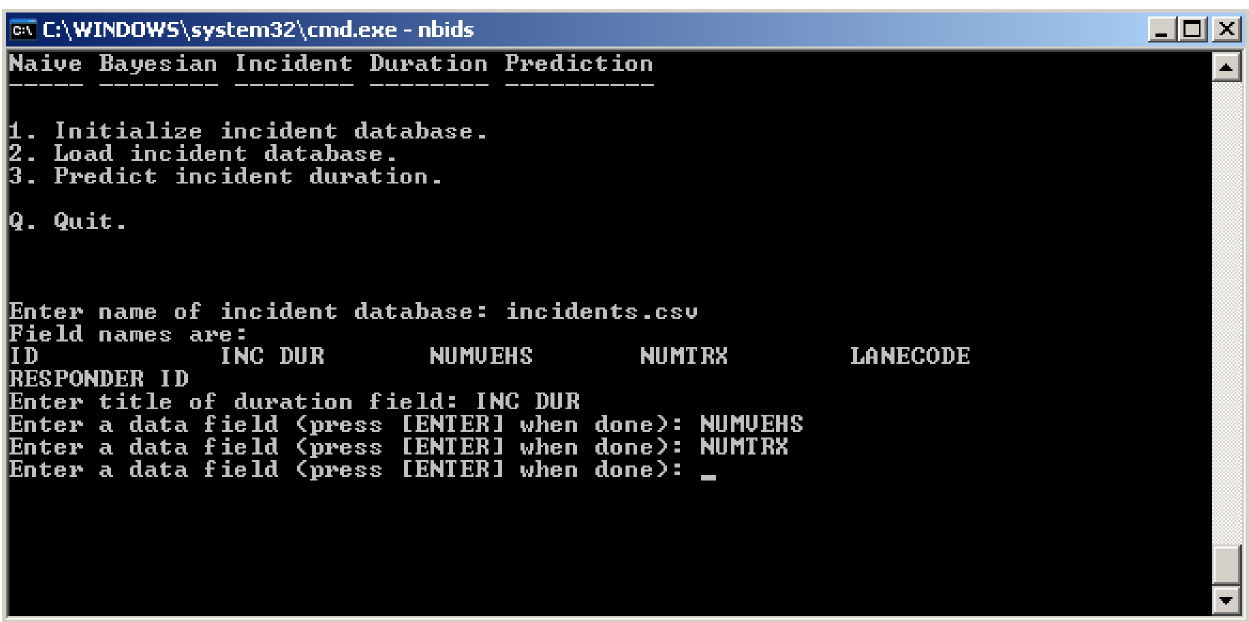


Figure A.4. Selecting the relevant columns in the database.

The second step is to identify the ranges of incident duration we're interested in. A Bayesian classifier works by estimating the probability that the duration of a given incident will fall into each of several categories; for instance, 0 to 30 minutes, 31 to 60 minutes, or more than 60 minutes. In NBIDS, this is done by entering the *highest* value in each category (called a *breakpoint*), in order: in this case, 30 and 60. The uppermost category is assumed to encompass everything greater than the last breakpoint, so even though there are three categories, we only need to enter two breakpoints.

Enter '30' and '60' when prompted for the duration breakpoints, followed by a blank line to indicate that you're done.

Finally, we need to specify the breakpoints for each of the data fields we selected — in this case, NUMVEHS and NUMTRX. These work in the same way as the duration breakpoints: the Bayesian classifier needs to know how values for these fields need to be "grouped." At one extreme, almost all of the values can be treated the same (e.g., for number of vehicles involved, one could choose only to differentiate between incidents involving more than one vehicle, and those only involving one). At the other extreme, one could choose to distinguish between all different numbers of vehicles. In practice, this decision will depend on how much data is available, how detailed it is, and on engineering judgment.

For this example, we will distinguish between incidents involving 1 vehicle, 2 vehicles, and more than 2 vehicles; and between those involving no trucks, and those involving at least one truck. As with duration, one enters this information into NBIDS by reporting the upper breakpoint for each category, and an additional category will be created for everything greater than the last entered breakpoint. **NBIDS will first prompt you for the breakpoints for NUMVEHS. Enter 1, 2, and a blank line. NBIDS will then prompt for the breakpoints for NUMTRX. Enter 0, and a blank line.**

At this point, the initialization process is complete (Figure A.5), and NBIDS creates an initialization file to save this information. Thus, this process only needs to be done when using a database for the first time, or when changing a database's configuration (for instance, if the names of the data fields change, or if you wish to change the breakpoints).

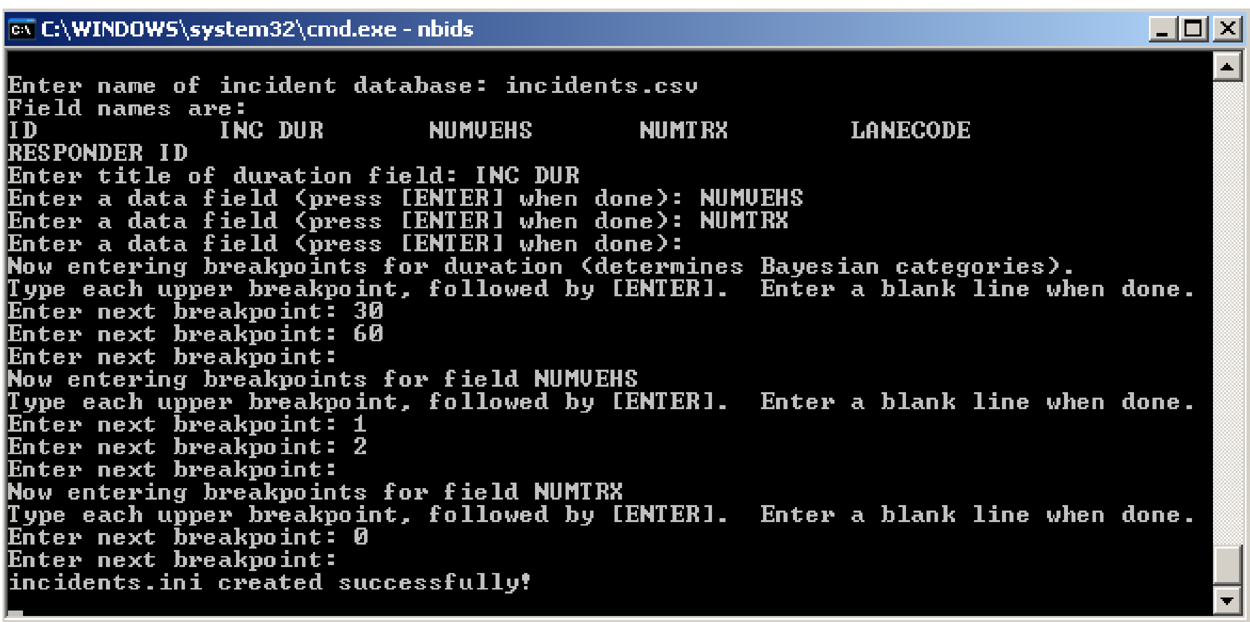


Figure A.5. Initialization is complete!

Step 3: Predict incident duration in real-time

After finishing the initialization process, NBIDS returns to its menu after waiting for a key press. The incident database must now be loaded before predicting incident durations. **At the**

menu, press ‘2’ and type ‘incidents.csv’ to load the database into memory, then press a key to return to the menu.

Now we are ready to predict the duration of an incident whenever one is detected — although the process of creating and initializing the database takes some time, this can be done offline and ahead of time. The actual prediction itself requires only seconds once the database is loaded. Let’s say that we learn of an incident involving one semitrailer truck, perhaps through a computer-aided dispatch system. Thus, we know that NUMVEHS and NUMTRX are both equal to one.

Press ‘3’ to predict the duration of this incident. NBIDS will prompt you for the values of NUMVEHS and NUMTRX; enter ‘1’ for both of these.

At this point, NBIDS will ask for the time since the incident occurred. This is an optional input that can be used to refine the prediction and account for the time at which our information about the incident was learned; however, for this tutorial, we choose not to use this option. **Press [ENTER] to generate the prediction.**

At this point, NBIDS returns the probability that the incident duration falls into the three categories defined in Step 2 (Figure A.6); we see that, based on the historical data, the most likely outcome is that the incident lasts between 30 and 60 minutes. Notice that the probabilities are positive for all three categories: this reflects the uncertainty involved in the prediction process, and the impossibility of calculating incident duration exactly.

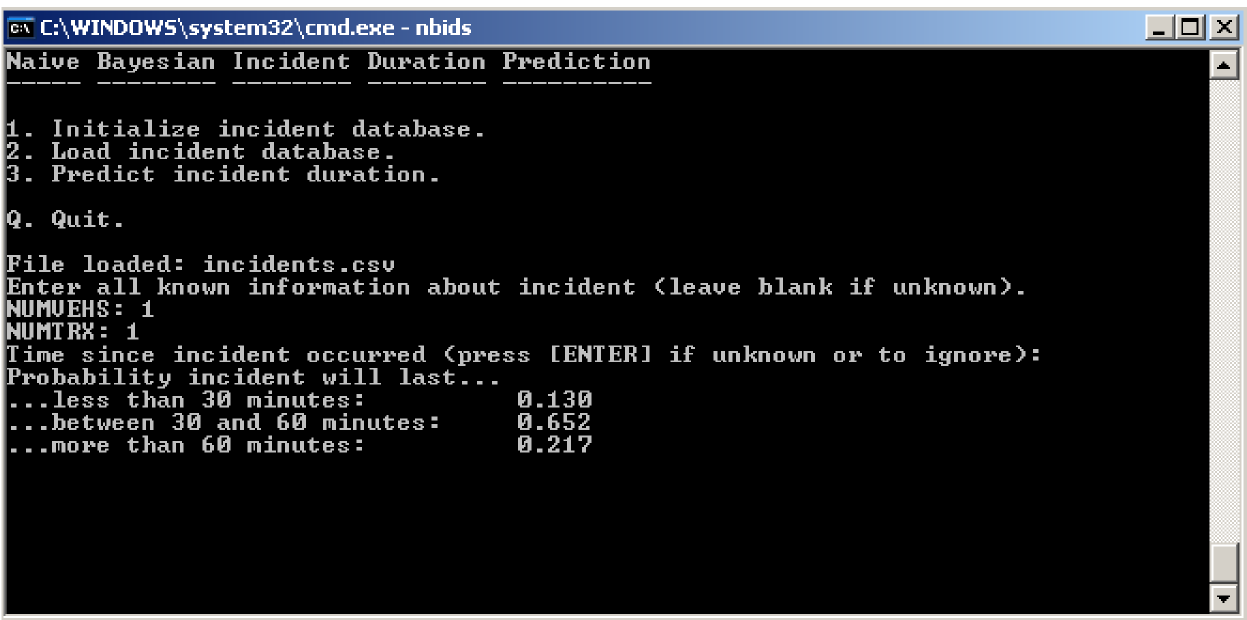


Figure A.6. Predicted incident duration

A.3 Prediction Methodology

This chapter explains the mathematics underlying the prediction model in NBIDS, first providing a brief review of the relevant concepts from probability, followed by descriptions of the naive Bayesian classifier and its application to incident duration prediction.

For predicting incident duration, the advantages of the naive Bayesian classifier are several; in particular,

1. Its operation is simple and intuitive, relying only on basic laws of probability.

2. It accommodates limited information, as seen in real-world incident management.

2a. As a corollary, this allows a broader set of model parameters to be used, as the model does not require observations for all independent variables.

3. Being explicitly probabilistic, it reports results in a form better suited for incident management policy.

4. It is robust to outliers.

5. It can account for information received at varying points in time.

Among the many models that have been proposed for modeling incident duration, the Bayesian models are the only ones that have all of these properties. In the research literature, two types of Bayesian models—Bayesian networks and naive Bayesian classifiers—have been formulated for this purpose. In developing NBIDS, the naive Bayesian classifier was selected because it is most straightforward to calibrate and use, and does not require sophisticated knowledge of probability on the part of the user.

Probability Review

Traditionally, incident duration prediction models attempt to predict the clearance time exactly, for instance, using a decision tree or regression analysis. The major disadvantage with this approach is that doing this is nearly impossible, and the model outputs are unnecessarily precise (a regression model, for example, might predict the clearance time for a particular incident to be 25.3 minutes). The reasons why this is difficult are obvious: incident management is a very unpredictable process, characterized by limited information, human behavior, and a host of other complications.

This is why probabilistic models are useful: instead of trying to explain away all of the uncertain factors to yield a single predicted duration, better results can be obtained by directly incorporating the uncertainty into the prediction. Bayesian models do this by estimating the probability that an incident's duration falls into one of several categories (such as "less than 30 minutes", "between 30 and 60 minutes", and "more than 60 minutes.")

A *probability space* is a mathematical construct which includes a set of *events*, and a probability measure indicating how likely each event is. For instance, an event might be "the incident lasts less than 30 minutes." Mathematically, if there are n different categories of incident duration, we can imagine a set of events $C_1, C_2, \ldots, C_n$ that, together, contain all possible durations for an incident. What we are hoping to find are the probabilities $\Pr(C_1), \Pr(C_2), \ldots Pr(C_n)$ which indicate the chance that each of these occurs.

There are other possible events, too, describing characteristics of the incident other than duration: "the incident involves two vehicles", "the incident blocks the leftmost lane", and so on. These types of events are unknown before the incident starts, but can be observed while the incident is still in progress. This is in contrast to duration, which is not observed until the incident clears. Thus, the goal of the naive Bayesian classifier is to use these observable events to infer the probabilities for the duration events, which cannot be known with certainty while the incident is still in progress.

Two other notions of probability that are needed for understanding naive Bayesian classifiers are *independence* and *conditional probabilities.* Independence has an intuitive interpretation: two events are independent if the outcome of one has no bearing on the likelihood of the other. As examples, two coin flips are independent, because the first flip does not influence the second; the high temperature on two successive days is not independent, as one day's weather certainly influences the next. From a mathematical standpoint, if A and B are independent events, it turns out that the probabilities "multiply": $\Pr(A \cap B) = \Pr(A)\Pr(B)$, where $A \cap B$ means that both A **and** B occur. In the context of incident management, imagine that A is the event "the incident involves two vehicles", and B is the event "the incident blocks the leftmost lane." If A and B are independent, the probability of $A \cap B$ ("the incident involves two vehicles **and** blocks the leftmost lane") is the product of the probability that the incident involves two vehicles, and the probability that the incident blocks the leftmost lane.

As mentioned above, some events can be observed while the incident is still in progress ("the incident blocks the leftmost lane"), while others cannot (the duration is unknown until the incident is over). However, the observable events still provide valuable information in helping to predict the duration — it is extremely unlikely that an incident involving over a dozen vehicles will last less than 15 minutes, for instance. Conditional probabilities are based on exactly this principle: *some knowledge can provide additional information on other, unknown knowledge*. This is written in expressions such as $\Pr(C_1 \mid A)$, which is read as "the probability of C_1 **given** B", and is interpreted as the likelihood that event C_1 happens, having the additional knowledge that B happens. Keeping with the same example, where A is "the incident involves two vehicles" and C_1 is "the incident lasts less than 30 minutes", $\Pr(C_1 \mid A)$ is the answer to the question "If that two vehicles are involved, what is the probability that the incident lasts less than 30 minutes." This is exactly the question that the naive Bayesian classifier seeks to answer.

A fundamental law of probability relates the conditional probability of two events to the probabilities of the events separately:

$$\Pr(X \mid Y) = \frac{\Pr(X \cap Y)}{\Pr(Y)}$$

Another basic relation for conditional probabilities is Bayes' Theorem

$$\Pr(X \mid Y) = \frac{\Pr(Y)\Pr(Y \mid X)}{\Pr(X)}$$

As described in the next section, this is the basic relationship that underlies the naive Bayesian classifier.

Introduction to Naive Bayesian Classifiers

In this section, we present the naive Bayesian classifier in its general form, that is, outside of the incident duration prediction context. The following section discusses how the classifier specifically functions in an incident management framework.

Like other classifiers, the naive Bayesian classifier attempts to determine which category (from a set $C_1, C_2, \ldots, C_n$ of n categories) best suits a particular object. Bayesian classifiers have been applied in many domains: one common use is as an e-mail spam filter (where the classifier tries to guess whether an e-mail is spam or not), or in other document classification problems. Still other uses include medical diagnosis and systems performance management.

In all of these cases, the categorization is based on observable *attributes* of the object in question (such as words in an email, or symptoms in a patient), which are denoted $X_1, X_2, \ldots, X_m$. Thus, we want to calculate the conditional probability that the object belongs to each of the different categories, given the attributes; that is, the probabilities $\Pr(C_i \mid X_1, X_2, \ldots, X_m)$ for each category i. Using Bayes' Theorem as presented in Section 1, this probability can be written as

$$\Pr(C_i \mid X_1, X_2, \ldots, X_m) = \frac{\Pr(C_i)\Pr(X_1, X_2, \ldots, X_m \mid C_i)}{\Pr(X_1, X_2, \ldots, X_m)}$$

This classifier is naive in the sense that it makes the strong assumption that the attributes are mutually conditionally independent; that is, the conditional probabilities of C_1, C_2, etc. given the value of some attribute is independent of the values of all other attributes. Under this assumption, the previous equation simplifies to

$$\Pr(C_i \mid X_1, X_2, \ldots, X_m) = \frac{\Pr(C_i)\prod_{j=1}^{m}\Pr(X_j \mid C_i)}{\Pr(X_1, X_2, \ldots, X_m)}$$

for each category C_i. Because the denominator will be the same for all categories, we only need to calculate the numerator for each category, choosing the optimal category $C^* \in \operatorname{argmax}\Pr(C_i)\prod_{j=1}^{m}\Pr(X_j \mid C_i)$.

Thus, the only information we need to calculate the necessary probabilities are (1) the *unconditional* or *a priori* probability that the object belongs to each category and (2) the *conditional* probability of observing the attributes we saw, given that the object belongs to each category. Notice how the conditional expression is reversed: we are interested in the probability of the object belonging to a particular category, given a set of observable attributes. Bayes' Theorem relates this to the probability of observing particular attributes, given that the object belongs to a certain category. Because of the mutual conditional independence assumption, gathering this data becomes an almost trivial problem, given enough historical data to estimate these probabilities. In fact, NBIDS calculates these probabilities automatically when loading a file, reducing the time needed to evaluate the Bayesian expressions to almost zero when an incident is in progress.

Application to Incident Duration Prediction

As might be inferred from the previous two sections, in incident duration, the categories C_i represent different ranges of incident duration, and the attributes $\mathbf{X}$ represent observable incident characteristics, such as the number of vehicles involved or the number of emergency responders present. It is important to distinguish that these X values are *events*, not just specific numbers. For instance, although the set of potential attributes could be "one vehicle is involved," "two vehicles are involved," and so on, it is also possible to aggregate some of these together: "one or two vehicles are involved," "more than two vehicles are involved."

In practice, some aggregation is desirable, because the accuracy of the classifier depends on the amount of data available to estimate the probabilities $\Pr(X_j \mid C_i)$. As an example, incidents involving thirty or thirty-one vehicles are quite rare; in fact, there may not even be any incidents involving exactly these numbers of incidents in the entire database. However, in practical terms, the difference between an incident involving thirty vehicles, and another involving thirty-one, is small, so it makes sense to combine them to increase the amount of data for calibration. In fact, using this logic it probably makes sense to group all incidents involving sufficiently many vehicles (perhaps seven or eight) into one attribute. Of course, taken to an extreme, aggregation renders the classifier useless, as it will be difficult to distinguish one incident from any other if the attributes are too broad. This issue is discussed in greater detail in Section A.6.

Thus, a reasonable middle ground must be found, with enough aggregation to ensure a useful amount of data for each attribute, but not enough to restrict the predictive power of the classifier. This is the main task in setting up a Bayesian classifier for incident duration prediction, and depends on the quantity of data available, engineering judgment, and many other factors.

Unlike some of the other applications of Bayesian classifiers, the incident management process is "dynamic" in that information is received at different points in time. Presumably, this should be accounted for in some way. For instance, if we know that the incident has already lasted 40 minutes, the probability that it will last, say, less than 30 minutes should be zero.

This can easily be done by adding an additional attribute T, indicating that the incident has lasted at least T minutes. T should be treated differently than the other attributes, because applying the mutual conditional independence assumption here is meaningless. Still, it is an easy matter to incorporate it into the classifier, by calculating the conditional probability of an incident's duration belonging to a certain category as

$$\Pr(C_i \mid X_1, X_2, \ldots, X_m, T) = \frac{\Pr(C_i) \prod_{j=1}^{m} \Pr(X_j \mid T, C_i)}{\Pr(X_1, X_2, \ldots, X_m)}$$

This formulation is flexible in that it can update the most probable category at any time (such as when new information is received); however, it is not without its drawbacks. First, the number of parameters needed grows substantially (multiplied by the number of possible values of T). Although this is not a major obstacle for NBIDS, the previous formulation can be completely and succinctly summarized in a single table, which is useful if one wishes to make qualitative statements about the impact of various attributes. Second, if the current time is large, there may be few incidents whose duration is greater than T, and the accuracy of the conditional probabilities may suffer due to a small sample size. Nevertheless, the ability to accommodate new information is of importance, and Boyles et al. [41] found that including the current time generally improves the predictive ability of the classifier.

As mentioned at the start of this chapter, one of the biggest advantages of the naive Bayesian classifier for incident duration prediction is the ability to account for incomplete information: if an attribute is not observed, it plays no role in the probability calculations. For example, if the number of vehicles is unknown, the attributes representing "X number of vehicles" will simply be ignored. Other types of models (such as those based on linear regression), require some input for every variable used for calibration, which is a severe restriction for this particular application.

Creating an Incident Database

This chapter describes the file format that the incident database needs to be stored in. For the example in the tutorial (see Section A.2), the data was prepared in Microsoft Excel®, although most spreadsheet software can also be used. The specific requirement is that the program be able to save a spreadsheet as a comma-separated value (CSV) text file, as described below. It is also possible to create this file directly, if the data is already in CSV format, or by typing the file directly using a text editor (although this latter option is usually too tedious for large data sets).

Incident Database File Format

In NBIDS, the incident database consists of a large table, with one row for each historical incident, and a header row at the top to indicate the title of each of the columns. Typically a spreadsheet program is the most efficient way to manage these files.

Often, the data set will be obtained from another source, and contain irrelevant columns. For example, the data may include columns indicating the geographic location of the incident, the operator responsible for logging the incident, a general comment field, and so on. It is perfectly fine to leave these in the database; during the initialization process (see Sections A.2 and A.6), you select which columns to use.

This table or spreadsheet must satisfy the following conditions:

- The spreadsheet must solely consist of the incident table
- Every row in the table must have the same number of columns
- The first row in the table must be the header, indicating the title of each column.
- No column name can be left blank or duplicated. (Blank columns should be deleted before converting to CSV format).
- No rows can be left blank.
- One column must indicate the duration of each incident.

Depending on the data set, some preprocessing may be necessary to satisfy the above conditions. Two additional qualifications apply to any columns which will be used to predict incident duration.

- Every data cell must consist of an integer (whole number) between 0 and 65,535.
- Any blank cell will be treated as if it were 0.

Non-integer numbers need to be rounded to an integer, and text entries are not allowable. For cases where the column represents an alphabetic code (e.g., “RS” for Right Shoulder, “RL” for Right Lane, and so on), one should replace this with numeric values (“0” for Right Shoulder, “1” for Right Lane, etc.). Most spreadsheet programs have find-and-replace functions that enable this to be done with relatively little effort. Note that this restriction only applies to columns which are to be used in the classifier. A comment field, for instance, which will not be used to predict incident duration is not subject to this requirement.

CSV Files

Any spreadsheet program can be used to enter or import the historical incident data, as long as it can then export the data in CSV format. A CSV file is essentially a table stored in text format, where the commas indicate the division between columns. For instance,

```
TITLE 1 TITLE 2 TITLE 3 Data 1 Data 2 Result 1 Data 3 Data 4 Result 2
```

is expressed in CSV format as

```
TITLE 1,TITLE 2,TITLE 3 Data 1,Data 2,Result 1 Data 3,Data 4,Result 2
```

Note that there is no space following the commas: the text in each cell consists of every character between successive commas. Because commas define the column breaks, one cannot include a comma in a cell without enclosing it in quotes `"like,this"`. Entering the text `like,this` will count `like` and `this` as separate entries. Most spreadsheet programs do this automatically, but you need to be aware of this if editing or entering data manually.

A.4 Initializing an Incident Database

This chapter describes the steps necessary to initialize the database in greater detail (item '1' on the main menu of NBIDS). This is the process in which you specify what information the classifier will use to predict incident duration. There are two main steps involved here:

1. Identify the fields you want NBIDS to use
2. Choose the appropriate breakpoints to define the attributes

After accomplishing these, NBIDS creates an initialization file with the same name as the database, with the `.ini` extension. This saves the configuration, so you do not need to repeat this process every time you load the database. You only need to re-initialize the database if there is a change in the fields you are using, or in your definitions of the attributes. You may update the data in the database itself (for instance, adding new incidents to the spreadsheet over time), while continuing to use the existing configuration.

NBIDS needs to know which column in the database represents incident duration, and which columns define the observable attributes (number of vehicles, location of incident, etc.) As mentioned in Section A.3, only numeric data can be used for this purpose; if you wish to include a text field, you must convert it to a numeric form.

The next, slightly more complicated, step is to define the attributes. To do this, you need to decide how the numeric fields should be "grouped." For instance, you might want to distinguish between incidents involving one vehicle, two vehicles, and three vehicles; but the difference between incidents involving four, five, or six vehicles is small enough that you want to treat them the same. The advantage of careful grouping is that the classifier works best when there are a substantial number of incidents showing each attribute. Even though there may be few incidents involving four, five, and six vehicles, by combining these into one category the sample size is increased.

Other possible groupings are shown in Figure A.7, where the parentheses indicate how the number of vehicles are grouped. The topmost row indicates that the exact number of vehicles involved is always important. This may be feasible if there's a large body of historical incidents to draw from, but also runs the risk of overcalibration — especially for severe incidents involving when more than ten or fifteen vehicles, the impact of one more vehicle is probably slight, and it might be better to combine them. The second row describes the scenario in the previous paragraph, in which four possible attributes are derived from the number of vehicles: "one vehicle involved," "two vehicles involved," "three vehicles involved," and "four or more vehicles involved." The third row simplifies still further: "one vehicle involved," "two or three vehicles involved", "four or more vehicles involved." The fourth row only distinguishes between two cases: "one or two vehicles involved," "three or more vehicles involved." The latter is probably too broad, and too much detail is lost in the aggregation. Other fields might suggest different aggregation. For instance, "number of semitrailer trucks involved" might be grouped (0) (12345...), distinguishing "no trucks involved" from "at least one truck involved."

(1) (2) (3) (4) (5) (6) (7) (8) ... (1) (2) (3) (45678 ... (1) (23) (45678 ... (12) (345678 ...

Figure A.7. Possible groupings for number of vehicles involved

NBIDS identifies these groupings by the *highest* number within each group. So, (1**2**) (345678...) is identified by **2**; (**1**) (2**3**) (45678...) is identified by **1**, **3**, and so on. The grouping in the to prow of Figure A.7 is identified by **1**, **2**, **3**, ... These numbers are called *breakpoints*. Note that no number is specified for the highest group; it is assumed that all values of the field above the largest breakpoint are grouped together.

This same system of identification works even when the values of the field do not have a direct physical interpretation. For instance, it is common to see numeric codes indicate the lanes blocked by the incident, where each number represents a different lane or combination of lanes. In such cases, every number indicates a different situation, so each group should consist of a single number, as in the top row of Figure A.7.

It is these breakpoints which are entered into NBIDS during the initialization process. For each field selected for the database (including the incident duration field), you will be prompted for the breakpoints. When finished entering the breakpoints for a field, enter a blank line to move to the next field.

A.5 Predicting Incident Duration

Once a database has been initialized, actually predicting the duration of incidents is extremely simple. Upon starting NBIDS, you must load an initialized database before predicting incident durations. (An error message is printed if no initialization file is found.) After loading the database, press '3' at the main menu to predict the duration of an incident.

You will be prompted for each of the fields in turn; any unknown information should be left blank. Finally, if you wish to use the current time to refine the prediction, you will be prompted to enter the time since the incident occurred after entering information for each field. You may choose not to use this feature by simply pressing `[ENTER]` when prompted for the time.

After obtaining this information, NBIDS will display the calculated probabilities for each of the different categories of incident duration.

A.6 Source Code

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'NBIDS: Naive Bayesian Incident Duration Prediction Software'
'                              '
'   Prepared for Texas Department of Transportation   '
'    Research Project 0-5422, Deliverabable P2      '
'                              '
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' ---------------------------
' Definitions and declarations
' ---------------------------
```

```
const ERRCODE_FILENOTFOUND = 2
const UINTEGERINFINITY = 4294967295

dim shared NumIncidents as uinteger
dim shared NumIncidentFields as ushort
dim shared NumFields as ushort
dim shared MaxBreaks as ushort

const Duration = 0  'field index for incident duration
const BLANK = 255  'value used for unknown data

dim shared DBFile as string
dim shared INIFile as string

declare sub ReadIncidentDatabase(DBFile as string, FieldName() as
  string, NumBreaks() as ushort, IncidentDB() as ushort, FieldBreaks()
  as uinteger, BayesianProb() as double, BaseProb() as double)
declare sub IncidentPredict(FieldName() as string, NumBreaks() as ushort,
  IncidentDB() as ushort, FieldBreaks() as uinteger, BayesianProb() as double,
  BaseProb() as double)
declare sub InitializeIncidentDatabase(DBFile as string)
declare sub FatalError(Message as string)



' ---------
' Main Loop
' ---------

redim FieldName(1) as string, NumBreaks(1) as ushort, IncidentDB(1, 1) as
ushort,
  FieldBreaks(1, 1) as uinteger, BayesianProb(1, 1, 1) as double, BaseProb(1)
  as double
dim as string a, InpStr
dim as string Loaded = ""
do
  cls
  a = inkey$ 'clear keyboard buffer
  do
   print "Naive Bayesian Incident Duration Prediction"
   print "----- -------- -------- -------- ----------"
   print
   print "1. Initialize incident database."
   print "2. Load incident database."
   print "3. Predict incident duration."
   print
   print "Q. Quit."
   print
   if len(Loaded) > 0 then print "File loaded: "; Loaded
   a = input$(1)
  loop until len(a) > 0
  select case a
  case "1"
   print
   print
   input "Enter name of incident database: ", InpStr
   InitializeIncidentDatabase InpStr
```

```
  case "2"
   print
   print
   input "Enter name of incident database: ", InpStr
   ReadIncidentDatabase InpStr, FieldName(), NumBreaks(), IncidentDB(),
     FieldBreaks(), BayesianProb(), BaseProb()
   Loaded = InpStr
  case "3"
   if len(Loaded) > 0 then
     IncidentPredict FieldName(), NumBreaks(), IncidentDB(), FieldBreaks(),
      BayesianProb(), BaseProb()
   else
     print
     print
     print "You must load a file first!"
   end if
  case "q", "Q"
   end
  end select
loop




'---------------------------------------------------------------
'ReadIncidentDatabase: Loads and processes an incident database
'---------------------------------------------------------------

sub ReadIncidentDatabase(DBFile as string, FieldName() as string, NumBreaks()
as
  ushort, IncidentDB() as ushort, FieldBreaks() as uinteger, BayesianProb()
  as double, BaseProb() as double)

  dim INIFile as string
  INIFile = left$(DBFile, instr(DBFile,".") - 1) + ".ini"

  '****** STEP 1. Open initialization file
  print "Opening initialization file...";
  if open(INIFile for input as #1) = ERRCODE_FILENOTFOUND then
   print "No initialization file exists for this database!"
   sleep
   exit sub
  end if
  dim i as uinteger, j as uinteger, k as uinteger

  input #1, NumIncidents, NumFields, NumIncidentFields, MaxBreaks

  'Read field names & numbers of breaks for processing incident database
  redim FieldName(0 to NumIncidentFields) as string
  redim FieldNumber(0 to NumIncidentFields) as uinteger
  redim IncidentDB(1 to NumIncidents, 0 to NumIncidentFields) as ushort
  redim NumBreaks(0 to NumIncidentFields) as ushort
  redim FieldBreaks(0 to NumIncidentFields, 1 to MaxBreaks) as uinteger
  for i = 0 to NumIncidentFields
   input #1, FieldName(i), NumBreaks(i)
  next
```

```
 'Finally read the breaks themselves
 for i = 0 to NumIncidentFields
  for j = 1 to NumBreaks(i)
    input #1, FieldBreaks(i, j)
  next
 next

 close #1
 print "Done."


 '***** STEP 2. Read incident database


 print "Processing incident database...";
 if open(DBFile for input as #1) = ERRCODE_FILENOTFOUND then
  print "Incident file not found!"
  sleep
  exit sub
 end if
 dim Dummy as string

 'First process header
 for i = 1 to NumFields
  input #1, Dummy
  for j = 0 to NumIncidentFields
    if FieldName(j) = Dummy then FieldNumber(j) = i
  next
 next
 'Now process file one field at a time
 for i = 1 to NumIncidents
  for j = 1 to NumFields
    input #1, Dummy
    for k = 0 to NumIncidentFields
     if FieldNumber(k) = j then IncidentDB(i, k) = val(Dummy)
    next
  next
 next
 close #1
 print "Done."


 '***** STEP 3. Calculate Bayesian probabilities
 print "Calculating Bayesian parameters...";
 redim  BayesianProb(1  to  NumBreaks(0),  1  to  NumIncidentFields,  1  to
MaxBreaks)
  as double
 redim BaseProb(1 to NumBreaks(0)) as double
 redim IncidentCount(1 to NumBreaks(0)) as uinteger
 dim DurationBin as ushort, FieldBin as ushort
 for i = 1 to NumIncidents
  'Find the right bin for incident duration
  DurationBin = 0
  do
    DurationBin += 1
  loop until FieldBreaks(Duration, DurationBin) >= IncidentDB(i, Duration)
  IncidentCount(DurationBin) += 1
```

```
    'Now go through each field, and increment the correct BayesianProb
    for j = 1 to NumIncidentFields
      FieldBin = 0
      do
       FieldBin += 1
      loop until FieldBreaks(j, FieldBin) >= IncidentDB(i, j)
      BayesianProb(DurationBin, j, FieldBin) += 1
    next
  next
  ' Normalize
  for i = 1 to NumBreaks(Duration)
    BaseProb(i) = IncidentCount(i) / NumIncidents
    for j = 1 to NumIncidentFields
      for k = 1 to MaxBreaks
       BayesianProb(i, j, k) /= IncidentCount(i)
      next
    next
  next
  print "Done."
  print "File loaded successfully! Press a key to return to menu."
  sleep

end sub




'-----------------------------------------------------------------------
'IncidentPredict: Prompts for incident information and predicts duration
'-----------------------------------------------------------------------

sub IncidentPredict(FieldName() as string, NumBreaks() as ushort,
  IncidentDB() as ushort, FieldBreaks() as uinteger, BayesianProb() as
  double, BaseProb() as double)

  '***** STEP 1. Input information about incident
  dim i as uinteger, Entry as string, CurTime as uinteger
  redim Incident(1 to NumIncidentFields) as ushort
  print "Enter all known information about incident (if any field is
   unknown, simply press [ENTER])."
  for i = 1 to NumIncidentFields
   print FieldName(i); ": ";
   input Entry
   if Entry = "" then Incident(i) = BLANK else Incident(i) = Val(Entry)
  next
  input "Time since incident occurred (press [ENTER] if unknown
   or to ignore): ",CurTime

  '***** STEP 2. Calculate probabilities
  redim DurationProb(1 to NumBreaks(Duration)) as double
  dim j as uinteger, FieldBin as ushort
  for i = 1 to NumBreaks(Duration)
   DurationProb(i) = BaseProb(i)
  next

  if CurTime = 0 then 'ignore "current time"
   for j = 1 to NumIncidentFields
```

```
    if Incident(j) <> BLANK then
     FieldBin = 0
     do
       FieldBin += 1
     loop until FieldBreaks(j, FieldBin) >= Incident(j)
     for i = 1 to NumBreaks(Duration)
       DurationProb(i) *= BayesianProb(i, j, FieldBin)
     next
    end if
  next
else 'recalculate Bayesian probabilities to account for "current time"
  redim CTBaseProb(1 to ubound(BaseProb)) as double
  redim CTBayesianProb(1 to ubound(BayesianProb, 1), 1 to
    ubound(BayesianProb, 2), 1 to ubound(BayesianProb, 3)) as double
  redim CTIncidentCount(1 to NumBreaks(0)) as uinteger
  dim CTDurationBin as ushort, CTNumIncidents as uinteger, k as uinteger
  for i = 1 to NumIncidents
    if IncidentDB(i, Duration) >= CurTime then
     CTNumIncidents += 1
     'Find the right bin for incident duration
     CTDurationBin = 0
     do
       CTDurationBin += 1
     loop until FieldBreaks(Duration, CTDurationBin) >= IncidentDB(i,
       Duration)
     CTIncidentCount(CTDurationBin) += 1
     'Now go through each field, and increment the correct BayesianProb
     for j = 1 to NumIncidentFields
       FieldBin = 0
       do
        FieldBin += 1
       loop until FieldBreaks(j, FieldBin) >= IncidentDB(i, j)
       CTBayesianProb(CTDurationBin, j, FieldBin) += 1
     next
    end if
  next
  ' Normalize
  for i = 1 to NumBreaks(Duration)
    CTBaseProb(i) = CTIncidentCount(i) / CTNumIncidents
    for j = 1 to NumIncidentFields
     for k = 1 to MaxBreaks
       CTBayesianProb(i, j, k) /= CTIncidentCount(i)
     next
    next
  next
  for j = 1 to NumIncidentFields
    if Incident(j) <> BLANK then
     FieldBin = 0
     do
       FieldBin += 1
     loop until FieldBreaks(j, FieldBin) >= Incident(j)
     for i = 1 to NumBreaks(Duration)
       DurationProb(i) *= CTBayesianProb(i, j, FieldBin)
     next
    end if
  next
end if
```

```
  'Normalize
  dim Total as double
  for i = 1 to NumBreaks(Duration)
   Total += DurationProb(i)
  next
  for i = 1 to NumBreaks(Duration)
   DurationProb(i) /= Total
  next

  '***** STEP 3. Output
  print "Probability incident will last..."
  print "...less than "; FieldBreaks(Duration, 1) ; " minutes: "; tab (35);
  print using "#.###"; DurationProb(1)
  for i = 2 to NumBreaks(Duration) - 1
   print "...between "; FieldBreaks(Duration, i - 1); " and ";
   print FieldBreaks(Duration, i); " minutes: "; tab(35);
   print using "#.###"; DurationProb(i)
  next
  print "...more than "; FieldBreaks(Duration, NumBreaks(Duration) - 1);
   " minutes: "; tab(35);
  print using "#.###"; DurationProb(NumBreaks(Duration))
  print
  print "Prediction complete; press a key to return to menu."
  sleep

end sub



'--------------------------------------------------------------------
'InitializeIncidentDatabase: Prompts user to set up incident database
'--------------------------------------------------------------------

sub InitializeIncidentDatabase(DBFile as string)
  if open (DBFile for input as #1) = ERRCODE_FILENOTFOUND then
   print "File not found!"
   sleep
   exit sub
  end if
  dim INIFile as string
  INIFile = left$(DBFile, instr(DBFile,".") - 1) + ".ini"

  dim Header as string, TempHeader as string
  dim InpStr as string
  line input #1, Header

  dim as uinteger NumIncidents = 0, i = 0, j = 0
  do
   line input #1, InpStr
   NumIncidents += 1
  loop until eof(1)
  close #1

  'parse for commas
  dim FieldEndPos as ushort
  dim as ushort NumFields = 0
```

```
TempHeader = Header
do
 'Found a new field...
 NumFields += 1

 'now trim the header
 select case left$(TempHeader, 1)
 case chr$(34) 'quotation mark

   FieldEndPos = instr(2, TempHeader, chr$(34))
   if FieldEndPos = 0 then
    FatalError "Quotes do not match in header row!"
   end if
   if FieldEndPos = len(TempHeader) then exit do
   TempHeader = mid$(TempHeader, FieldEndPos + 2)
 case else
   FieldEndPos = instr(TempHeader, ",")
   if FieldEndPos = 0 then exit do
   TempHeader = mid$(TempHeader, FieldEndPos + 1)
 end select

loop

redim FieldName(1 to NumFields) as string, DatabaseFields(1 to
 NumFields) as string

'now read in the field names
TempHeader = Header
do
 i += 1
 select case left$(TempHeader, 1)
 case chr$(34) 'quotation mark
   FieldEndPos = instr(2, TempHeader, chr$(34))
   if FieldEndPos = 0 then
    FatalError "Quotes do not match in header row!"
   end if
   FieldName(i) = ltrim$(rtrim$(mid$(left$(TempHeader,
    FieldEndPos - 1), 2)))
   if FieldEndPos = len(TempHeader) then exit do
   TempHeader = mid$(TempHeader, FieldEndPos + 2)
 case else
   FieldEndPos = instr(TempHeader, ",")
   FieldName(i) = ltrim$(rtrim$(left$(TempHeader,
    FieldEndPos - 1)))
   if FieldEndPos = 0 then exit do
   TempHeader = mid$(TempHeader, FieldEndPos + 1)
 end select
loop

print "Field names are: "
for i = 1 to NumFields
 print FieldName(i),
next
print

'Now ask user to identify fields
dim DurationField as ushort
```

```
input "Enter title of duration field: ", InpStr

do
 for i = 1 to NumFields
   if InpStr = FieldName(i) then
    if DurationField = 0 then
      DurationField = i
    else
      FatalError "More than one field with that name!"
    end if
   end if
 next
loop while DurationField = 0

dim NumDataFields as ushort = 0
redim DatabaseField(1 to NumFields) as string
do
 input "Enter a data field (press [ENTER] when done): ", InpStr
 if len(InpStr) = 0 then exit do
 for i = 1 to NumFields
   if InpStr = FieldName(i) then
    if DatabaseField(i) = "" then
      for j = 1 to NumDataFields
       if DatabaseField(j) = InpStr then
         print InpStr; " is already included in the database."
         exit for, for
       end if
      next
      if j > NumDataFields then
       NumDataFields += 1
       DatabaseField(NumDataFields) = FieldName(i)
       exit for
      end if
    else
      FatalError "More than one field with that name!"
    end if
   end if
 next
 if i = NumFields + 1 then print "Field not found!"
loop

for i = 1 to NumDataFields
 print i, DatabaseField(i)
next

dim BreakPointString as string, MaxBreakPoints as integer
redim NumBreakPoints(0 to NumDataFields) as integer
MaxBreakPoints = 0
for i = 0 to NumDataFields
 NumBreakPoints(i) = 0
 if i > 0 then
   print "Now entering breakpoints for field "; DatabaseField(i)
 else
   print "Now entering breakpoints for duration (determines Bayesian
    categories)."
 end if
 print "Type each upper breakpoint, followed by [ENTER]. Enter a blank
```

```
      line when done."
    do
      input "Enter next breakpoint: ", InpStr
      if len(InpStr) = 0 then exit do
      NumBreakPoints(i) += 1
      BreakPointString += str$(val(InpStr)) + chr$(255)
      print BreakPointString
    loop
    if NumBreakPoints(i) > MaxBreakPoints then MaxBreakPoints =
      NumBreakPoints(i)
  next
  'Account for upper buffer
  MaxBreakPoints += 1

  redim BreakPoints(0 to NumDataFields, 1 to MaxBreakPoints) as uinteger
  for i = 0 to NumDataFields
    for j = 1 to NumBreakPoints(i)
      print BreakPointString
      BreakPoints(i, j) = val(left$(BreakPointString,
       instr(BreakPointString, chr$(255)) - 1))
      print BreakPoints(i,j)
      BreakPointString = mid$(BreakPointString, instr(BreakPointString,
       chr$(255)) + 1)
    next
    BreakPoints(i, NumBreakPoints(i) + 1) = UINTEGERINFINITY
  next

  'Write to .INI file
  open INIFile for output as #1
  write #1, NumIncidents, NumFields, NumDataFields, MaxBreakPoints
  write #1, FieldName(DurationField), NumBreakPoints(0) + 1

  for i = 1 to NumDataFields
    write #1, DatabaseField(i), NumBreakPoints(i) + 1
  next

  for i = 0 to NumDataFields
    for j = 1 to NumBreakPoints(i) + 1
      write #1, BreakPoints(i, j)
    next
  next

  close #1
  print INIFile; " created successfully! Press a key to return to menu."
  sleep
end sub

'-------------------------------------------------------------
'FatalError: Displays an error message and terminates program.
'-------------------------------------------------------------

sub FatalError(Message as string)
  print Message
  end
end sub
```

Appendix B: Incident Delay Prediction Simulation Results

B.1 Introduction

While the previous section presents analytical extensions of traditional delay formulas to account for uncertain incident duration, these formulas are not necessarily applicable in practice without further refinement, due to several assumptions made in the derivation. In particular, having a uniform non-congested region in a space-time diagram (see Figure 3.1) requires the assumption of constant vehicle demand. While this may be applicable for incidents of short duration, or those that occur during off-peak periods, this assumption becomes problematic when considering extended incidents during periods of heavy congestion – which is precisely when these models are most useful.

Thus, we employ a simulation approach to both validate the theoretical results for the case of uniform demand, and investigate the impacts of varying demand. These simulations are based on the cell transmission model developed by Daganzo [42], a mesoscopic simulator which is consistent with basic shockwave theory, yet well suited to efficient computer implementation. We consider the impact of an incident on a generic freeway segment, whose duration is obtained from a truncated lognormal distribution with parameters $\mu = 3$ and $\sigma = 1.6$, with a maximum incident duration of approximately 50 minutes. This choice of distribution results in an average incident length of approximately 15 minutes, with a standard deviation of 9 minutes.

The incident is assumed to occur on an isolated three-lane freeway segment, with initial capacity of 6600 vehicles per hour. For the duration of the incident, freeway capacity is reduced to 3000 vehicles per hour, perhaps representing complete blockage of one lane, with additional capacity reduction due to the presence of emergency vehicles and merging requirements.

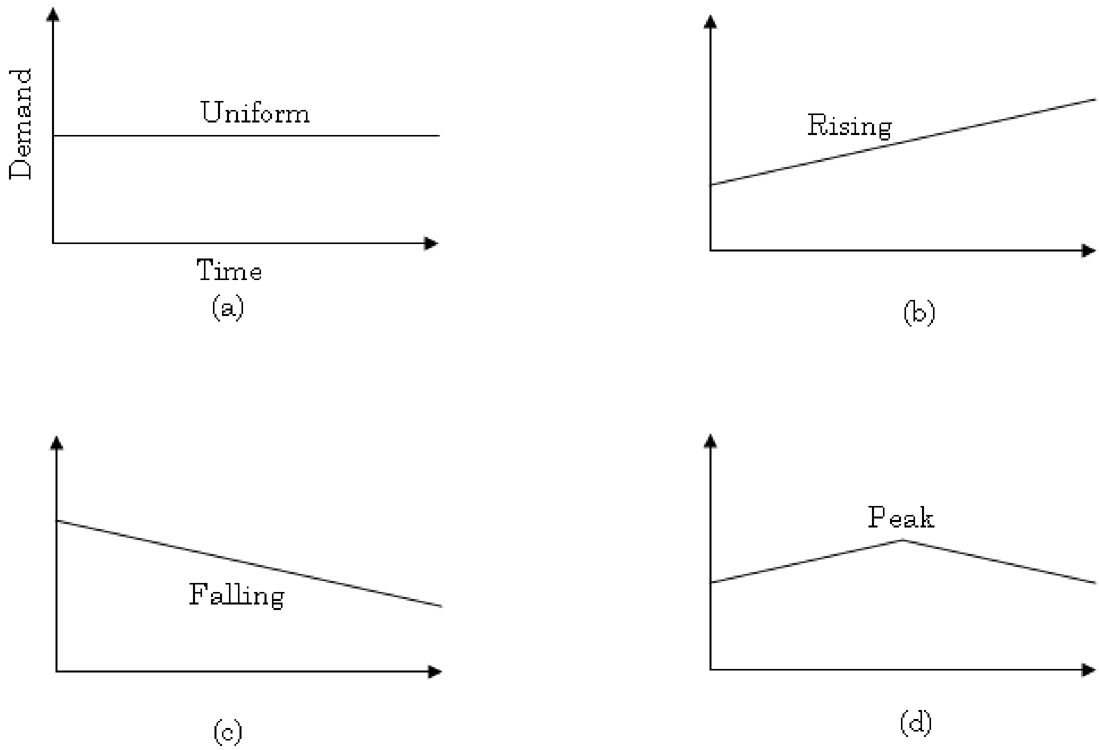


Figure B-1. Four demand profiles used for simulation.

Four different demand profiles are considered, four of which are shown in Figure B-1. Figure B-1(a) shows a uniform profile, where demand is constant over the analysis period. As mentioned above, this is the assumption in the Wirasinghe and Morales formulae. Figure B-1(b) indicates "rising" demand where the number of travelers is increasing (as at the beginning of a peak period), while Figure B-1(c) indicates "falling" demand, as would occur at the end of a peak period. Figure B-1(d) shows a "peaking" profile where demand first rises, then falls. Regardless of the profile, the average demand during the incident is 5000 vehicles per hour, indicating heavy (but uncongested) flow prior to the incident, but a congested state exceeding the freeway capacity while the incident is occurring.

For each of these demand profiles, we simulate 1000 incidents by applying Monte Carlo sampling to the lognormal distribution for incident duration; these are attained by applying the Box-Muller transform [43] to generate samples from a normal distribution, then exponentiating these to generate lognormal samples. To determine the delay caused by an incident, the travel time corresponding to each demand profile without any incidents was determined through simulation; this was subtracted from the total travel time when incidents occurred, leading to the additional delay due to the incident. Table B-1 summarizes these results.

Table B-1. Total delay results from Monte Carlo simulation (all values in vehicle-hours).

Profile	Average delay	Standard deviation	Skewness
Uniform	179.60	247.93	1.69
Rising	122.30	137.07	1.04
Falling	343.58	524.22	1.95
Peak	206.67	281.69	1.65

Clearly, the average delay differs substantially according to how demand is changing while the incident occurs. Compared to the uniform profile, delay was lower when demand was rising during the incident, and higher when demand was falling. This makes intuitive sense: when demand is falling, more vehicles encounter the incident while it is in progress, and thus must face a long queue. When demand is rising, the majority of vehicles encounter the incident after it has cleared, and encounter the queue in a recovery state, instead of a lengthening state. All of the profiles exhibit positive skewness, indicating asymmetry: when an incident is worse than average, its effects are likely to be much worse than average; when an incident is better than average, its effects are only slightly less than average.

As shown in the previous section, for the case of uniform demand, incident delay is directly proportional to the square of incident duration. For other demand profiles, the relationship will not be exact, but the ratio of delay to squared duration still gives an indication of the rate at which incident delay increases with duration. Table B-2 shows the average and standard deviations of this ratio for the different demand profiles. The zero value of standard deviation for the uniform profile indicates that the proportionality relationship is exact; for the other profiles, the larger the standard deviation, the more the delay-duration relationship differed from quadratic. Combining this result with the skewness results from Table B-1, it is evident that even when accounting for uncertainty, the uniform demand assumption may introduce additional underprediction in delay estimation, depending on the actual demand profile.

Table B-2. Proportionality results from simulation.

Profile	Average ratio	Standard deviation
Uniform	0.50	0.00
Rising	0.45	0.08
Falling	0.75	0.18
Peak	0.56	0.04

Finally, we seek to quantify the benefit of accounting for uncertainty, by comparing the prediction yielded by this model to what would have been predicted by simulating a single incident whose duration was equal to the average (15 minutes). These results are shown in Table B-3, and show that, depending on the profile, only using a single average duration value underestimates true average delay by 20-50%, certainly a significant amount. Using the average value with a falling demand profile results in particularly bad underestimation.

Table B-3. The degree of underestimation from ignoring stochasticity.

Profile	True average delay	Estimated delay	Underestimation
Uniform	179.60	106.22	41%
Rising	122.30	95.49	22%
Falling	343.58	159.04	54%
Peak	206.67	119.63	42%

Appendix C: Example of Prediction Method

In this section, we demonstrate the application of the two-phase ramp closure model to a hypothetical incident on a freeway in El Paso. As the intent here is simply to show how the procedure works, several practical issues (such as exact signal timing plans) are simplified or ignored, and as such these results should not be taken as engineering or policy recommendations.

In particular, we focus on a 2.6 mile segment of eastbound I-10 that passes through downtown, as shown in Figure C-1, during the evening peak period (3-6 PM). For Phase I, we use a planning network for the entire El Paso metropolitan area, comprising 681 zones, 2437 nodes, and 5233 links. Travel times on each of these links is assumed to follow the standard Bureau of Public Roads formula with $\alpha = 0.15$ and $\beta = 4$. The Frank-Wolfe algorithm was used to solve for the initial user equilibrium; path flows were calculated by simple averaging, and no entropy maximization method was applied.

For Phase II, the VISUM and VISSIM software packages were used. The microsimulator VISSIM was used to microsimulate and evaluate the ramp closure scenarios; for each of these, one iteration of VISSIM's dynamic equilibrium procedure was applied to reroute vehicles. VISUM was used to perform the traffic assignment needed to identify initial vehicle trajectories. Because both VISUM and VISSIM share the same developer (PTV), they interface readily, allowing the incident corridor to be directly extracted from the macroscopic model (see Figure C-2).

In our example, the incident occurs just upstream of the Cotton St onramp (see Figure C-1), and is severe enough to block three out of five travel lanes, reducing the freeway's capacity to 2000 vehicles per hour due to the merging that accompanies the lane blockage, and to the presence of emergency vehicles which may further disrupt flow. (When no incident is present, this facility has a capacity of 11,000 vehicles per hour.) We assume that this incident is predicted to last for one hour, and that there are four candidate onramps that can be closed: those from Porfirio Diaz St, Wyoming Ave, Franklin Ave, and Cotton St.

With this setup, we begin Phase I to determine which ramps should be closed. At the time the incident occurs, the initial equilibrium flows are already known and available, because these are found in the absence of an incident, and can be determined offline. Thus, the algorithm proceeds to test all sixteen possible combinations of open and closed ramps (including closing no ramps at all), evaluating TSTT for each of these, as shown in Table C-1. This entire analysis requires less than four seconds of computation time on a 3.40 GHz Pentium 4 machine, running Windows XP with 2 GB of RAM.

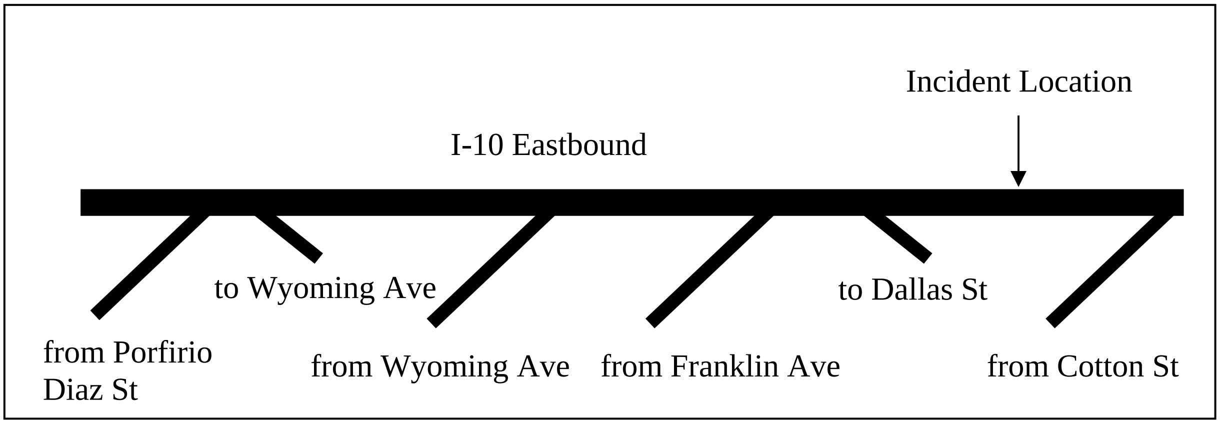


Figure C-1. Schematic of freeway used in example (not to scale).

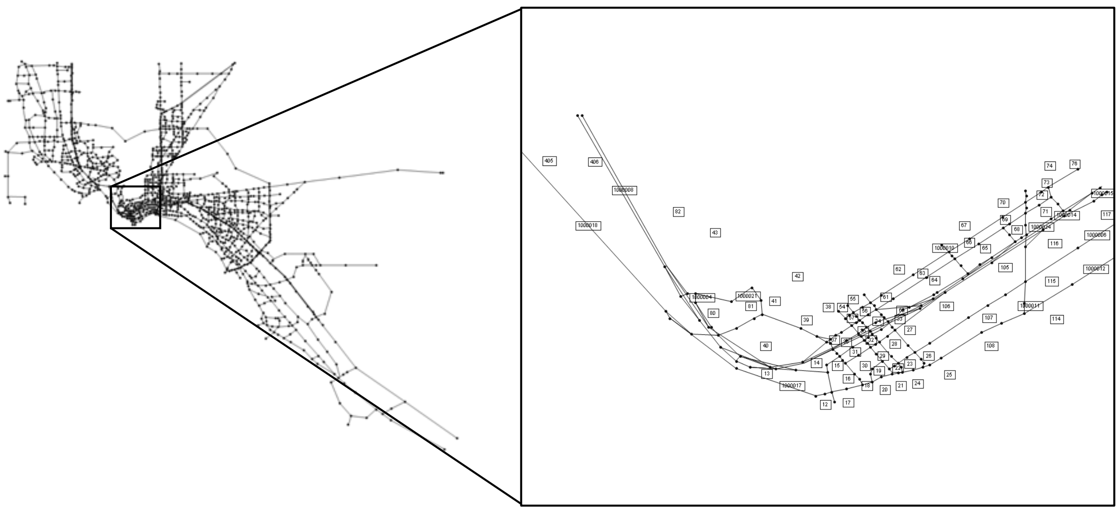


Figure C-2. El Paso network for Phase I, and zoom on incident corridor for Phase II

Of all the combinations, the best system performance occurs when only the Wyoming Ave onramp is closed. At first glance this may be surprising, as it is not the ramp closest to the incident, and it is possible for drivers to merge onto the freeway at the Franklin Ave ramp just downstream. However, closer examination of the before-and-after link flows show that most drivers choose to reroute on parallel arterials which are relatively uncongested, even after the entire ramp's volume is distributed among them. Further, closing the Franklin Ave ramp as well is not recommended because its volume is somewhat higher than that on the Wyoming Ave ramp – while closing this ramp would improve freeway performance more, in this case the negative impact of increased traffic on arterials outweighs the improvement seen on the freeway. Thus, while this policy seems counterintuitive when only considering the immediate vicinity of the incident, the macroscopic perspective afforded by Phase I shows that the best tradeoff between freeway and arterial performance is seen by closing only the Wyoming Ave ramp.

Table C-1. Phase I Results (O = Ramp Open, X = Ramp Closed)

					Time (s)	Cumulative (s)
Initialization and Shortest Paths					1.01	1.01
Porfirio Diaz St	Wyoming Ave	Franklin Ave	Cotton Ave	TSTT (veh-hrx10^6)		
O	O	O	O	4.46	0.02	1.02
X	O	O	O	4.48	0.11	1.13
O	**X**	**O**	**O**	**3.44**	**0.12**	**1.25**
X	X	O	O	3.45	0.16	1.42
O	O	X	O	3.63	0.13	1.55
X	O	X	O	3.63	0.26	1.81
O	X	X	O	3.78	0.20	2.01
X	X	X	O	3.79	0.24	2.25
O	O	O	X	9.44	0.12	2.36
X	O	O	X	9.48	0.16	2.52
O	X	O	X	5.15	0.17	2.69
X	X	O	X	5.16	0.24	2.93
O	O	X	X	3.97	0.17	3.09
X	O	X	X	3.97	0.24	3.33
O	X	X	X	4.14	0.25	3.59
X	X	X	X	4.15	0.32	3.90

Because this analysis only requires four seconds, the procedures to close the Wyoming Ave ramp can begin almost immediately after the incident is detected and deemed severe enough to warrant ramp closure. At this point, Phase II simulations should commence to determine the appropriate length of time that this ramp should remain closed. Because the anticipated incident length is 60 minutes, we begin by simulating a 60-minute ramp closure, followed by 50- and 70-minute closures, 40- and 80-minute closures, and so on, as time permits. For this case, assume that we wish to allow 15 minutes for Phase II. On the same machine, each microsimulation requires approximately five minutes, so we will simulate three scenarios: closing the ramp for 50, 60, and 70 minutes.

The results of these simulations is shown below, in Table C-2, compared using average freeway density as the measure of effectiveness (certainly, there are many other measures that could have been used as well). Considering density averaged over the two-hour time period from the beginning of the incident to the end of the peak period, the best freeway performance is seen when the ramp is closed for 50 minutes.

At this point, having completed the two-phase ramp closure model, the incident should continue to be monitored. Because incidents are constantly evolving, the remaining blockage time predictions need to be updated as more information is received, and additional Phase II simulations run if necessary.

Finally, although not specifically a step of Phase II, for the purpose of comparison we show the average freeway density if no ramp closure was to be enacted. Clearly, freeway operations are improved by closing the Wyoming Ave ramp during the incident.

Table C-2. Phase II Results

Closure Duration	Average freeway density (veh/mi) 4-5 PM	5-6 PM	4-6 PM
60 min	32.02	28.93	**30.48**
70 min	32.02	28.11	**30.06**
50 min	31.76	23.04	**27.40**
0 min (no closure)	35.43	51.27	**43.35**