Université de Montréal

A Discrete Flow Model For Dynamic Network Loading

par

Michael Mahut

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en informatique, option recherche opérationnelle

août, 2000

© Michael Mahut, 2000

Grade octroyé
à compter du
FEV 1 2001

QA
76
U54
2001
v.005

Université de Montréal

Faculté des études supérieures

Cette thèse intitulée

A Discrete Flow Model For Dynamic Network Loading

présentée par

Michael Mahut

a été évalué par un jury composé des personnes suivantes:

Sang Nguyen Professeur titulaire, Directeur Département d'informatique et de recherche opérationnelle Université de Montréal	président du jury
Michael Florian Professeur titulaire Département d'informatique et de recherche opérationnelle Université de Montréal	directeur de recherche
Bernard Gendron Professeur adjoint Département d'informatique et de recherche opérationnelle Université de Montréal	membre du jury
David Boyce Professor of Transportation and Regional Science Department of Civil and Materials Engineering University of Illinois at Chicago	examinateur externe

Thèse acceptée le: 19 décembre 2000

Abstract

A dynamic network loading (DNL) model is any model that defines a mapping from the time-dependent path demand flows to the time-dependent arc variables of flow, density and speed. A DNL model is an integral component of a dynamic traffic assignment (DTA) model, which is a mapping from the (time-dependent) origin-destination demand flows to the path demand flows. All DNL models have some underlying mechanism of flow propagation, which may be based on macroscopic traffic theory (fluid dynamics), microscopic traffic theory (car following models), or on approaches derived from static network equilibrium models (e.g., travel time functions).

This work presents an original discrete flow traffic model, derived from a simplified car-following relationship. When the sources of delay are located at fixed positions in a network, the model can be solved by only considering the traffic flow at these positions. Traffic networks satisfy this criterion, since the sources of delay – such as traffic signals and highway merges – are located at the nodes. The model can thus be solved directly for the traffic flow at these nodes, without the longitudinal discretization of arcs. This property is unique to this traffic model, and results in exceptionally fast computation times. The traffic model is extended to a multi-lane arc model, which considers the congestion effects due to the lateral motions of vehicles (i.e., changing lanes). The multi-lane model requires that the lane by which a vehicle will exit a given arc be chosen before it enters this arc. The model thus defines a mapping from the time-dependent lane-based demands (by arrival and departure lane) to the arc variables of flow, density, and speed, using a discrete-event procedure. Several supply/demand scenarios are tested for a small network, and the results are found to compare very well with those obtained using the AIMSUN2 traffic simulator. Execution times ranged from roughly one-tenth to one-fifteenth of the time required by the AIMSUN2 simulator when running in batch mode (without animation).

The node problem associated with a discrete-event discrete-flow arc model is defined, and a solution procedure is proposed. The solution procedure takes as an input the pair-wise precedence relationships obtained using a gap-acceptance model, and returns the next vehicle to enter the node. The algorithm is embedded in a discrete-event network loading procedure. Heuristics are proposed for the lane choice required by the arc model, and the complete DNL model is tested on a small network representing a section of highway connected to a service road with two ramps. The results compare quite well with those obtained using the INTEGRATION traffic simulator. Execution time was roughly one two-hundredth of that required by the INTEGRATION simulator, which does not have the option of running in a high-speed mode without animation. Noticeable discrepancies occur in flow and density on arcs where a high degree of lane changing is combined with delays propagating from downstream arcs. These discrepancies are restricted to relatively short arcs however, and thus have very little effect on overall path travel times, which are the critical measure for dynamic traffic assignment algorithms.

Compared to existing traffic models, the proposed model provides a unique combination of accuracy and computational speed. For this reason, it is particularly well suited for use with dynamic traffic assignment algorithms. Further testing of the model using empirical data from real networks is recommended.

Résumé

Un nouveau modèle dynamique de chargement de réseau ("dynamic network loading", égal à DNL) de flot discret fait l'objet de la présente thèse. Un modèle DNL est un modèle qui définit la projection du flot de demande sur les chemins (en fonction du temps) aux variables d'arc. (les variables étant le flot, la densité et la vitesse, en fonction du temps) Un modèle DNL fait partie intégrante d'un modèle d'affectation de trafic (DTA), qui est la projection des flots de demande des toutes les origines à toutes les destinations (en fonction du temps) aux flots de demande des chemins. Le modèle est basé sur un modèle de poursuite ("car following model") simplifiée. Une des propriétés importantes de tels modèles est le rapport empirique bien connu entre le flot et la densité (souvent appelé le diagramme fondamental du trafic), alors que plusieurs modèles proposés dans la littérature comme modèles DNL ne possèdent pas cette propriété.

Le modèle proposé ressemble à un modèle de Gipps, mais avec les contraintes sur l'accélération et la décélération relaxées. Ce modèle ressemble donc aux modèles de trafic basés sur les "Cellular Automata" (CA). Une différence entre les CA et le modèle proposé est que les CA sont basés sur une notion discrète de temps; par contre, le modèle proposé utilise une mesure de temps continue. Le modèle proposé a la propriété suivante: lorsque les trajectoires sont inversées — de $x(t)$ à $t(x)$ — il en résulte un modèle dont la solution provient des valeurs de $t(x)$ pour chaque véhicule aux positions où il y a des causes de retard sur le réseau. Puisque les retards sur le réseau — appelés également mécanismes de congestion — se rencontrent habituellement aux nœuds, en raison des feux de circulation ou bien des bretelles d'entrées sur les autoroutes, la solution proposée par le présent modèle est définie strictement en termes de temps de passage de chaque véhicule à chaque nœud sur son chemin. La solution donc n'implique pas la discrétisation de l'espace entre les nœuds; c'est à dire, n'exige pas la discrétisation des arcs. Il s'agit alors d'une solution *au niveau* des arcs. L'algorithme est une procédure à événements discrets, et non à temps discret. Cette approche est utilisée également pour les modèles de files d'attente. L'algorithme développé sert comme mécanisme de propagation du flot pour le modèle DNL proposé.

En général, un modèle DNL doit résoudre deux autres problèmes de trafic, autre que le problème de propagation du flot. Un de ces problèmes concerne le comportement du trafic aux nœuds, où le flot sur les arcs en amont peut définir une demande pour les arcs en aval qui dépasse leur capacité. Il faut donc définir un *modèle de nœud* qui détermine comment l'offre est distribuée parmi les demandes provenant des arcs en amont. Aux intersections dans les réseaux routiers, il faut souvent utiliser des voies spécifiques pour effectuer certains mouvements (d'une rue en amont à une rue en aval). Les véhicules donc doivent souvent effectuer des changements des voies pour rester sur un chemin particulier. Ces changements des voies peuvent générer des demandes pour une voie excédant sa capacité. Il faut donc définir un *modèle d'arc* qui distribue l'offre de chaque voie parmi les demandes de chaque voie.

Le modèle de flot de trafic proposé a été adapté aux arcs à voies multiples de façon à tenir compte de l'interaction entre les voies sans perdre la propriété de solution au niveau des arcs. Le modèle permet un seul mouvement latéral par véhicule par arc. Par ailleurs, la voie de sortie doit avoir été choisie avant l'entrée du véhicule dans l'arc. Le modèle à voies multiples définit donc la projection du flot de demande provenant des voies (voie d'arrivée et de sortie) aux variables d'arc. (flot, densité, vitesse) Le modèle a été implanté en C++, et a été testé sur un petit réseau qui présentait un grand nombre de changements de voie. Les résultats étaient comparables à ceux obtenus en utilisant le micro-simulateur de trafic AIMSUN2 pour plusieurs scénarios d'offre et de demande. Il a été possible de conclure que les contraintes du modèle, incluant la limite d'un seul mouvement latéral par véhicule, suffisent pour reproduire de façon satisfaisante les effets des changements de voie sur les statistiques d'arc. L'exécution du modèle proposé prend (au pire des cas) dix (10) fois moins de temps par rapport à l'AIMSUN2. Les postulats du modèle impliquent une description microscopique de la trajectoire de chaque véhicule. Les postulats ont donc pu être évalués à l'aide d'un petit exemple qui a été résolu à la main et animé seconde par seconde.

Le problème des nœuds correspondant à un modèle de flot discret et d'événements discrets a été défini, et un algorithme a été proposé pour le résoudre. Le modèle des

nœuds utilise le concept bien connu de l'acceptation de l'intervalle ("gap acceptance") entre les véhicules pour résoudre les conflits entre les paires de véhicules. L'algorithme proposé reconnaît le véhicule qui devrait précéder les autres dans le nœud compte tenu des résultats des comparaisons entre chaque paire. Ce problème est non trivial. L'algorithme utilisé pour résoudre le problème des nœuds est intégré à une procédure de chargement de réseau à événements discrets. Par conséquent, chaque fois que les données impliquées dans cet algorithme changent à un nœud quelconque, l'algorithme est exécuté de nouveau pour ce nœud. Des règles heuristiques pour le choix des voies, nécessaire au modèle d'arc, sont proposées. Le modèle complet a été testé sur un petit réseau contenant 12 arcs et 12 nœuds, représentant un tronçon d'autoroute entre une bretelle d'entrée et une bretelle de sortie, avec une voie de service parallèle. Il a été testé dans des conditions où l'offre et la demande variaient en fonction du temps. Les résultats étaient comparables à ceux obtenus avec le micro-simulateur du trafic. Des divergences non-négligeables ont été repérées dans le flot et la densité lorsqu'il y a, à la fois, un nombre élevé de changements de voie et des retards provenant des arcs en aval. Il est recommandé d'effectuer davantage de tests en utilisant des données concrètes. Le temps d'exécution d'INTEGRATION est deux cents (200) fois plus grand que celui du modèle DNL proposé.

Mots clés : chargement dynamique de réseau, théorie de flot de trafic, modèles de poursuite, modèles de files d'attente, simulation de trafic, affectation dynamique du trafic.

Extended Abstract

This work presents an original discrete-flow dynamic network-loading (DNL) model. A DNL model is any model that defines a mapping from the time-dependent path flows to the time-dependent arc variables of flow, density and speed. A DNL model is an integral component of a dynamic traffic assignment (DTA) model, which is a mapping from the (time-dependent) origin-destination flows to the path flows. In general, there is interdependence between a DNL and a DTA model, as the path flows are dependent on the arc speeds (or travel times), while arc speeds are dependent on the path flows. The proposed model is derived from a simplified car-following relationship. An important property of car following models is that they yield a well-known relationship between flow and density, often referred to as the fundamental diagram of traffic. Models that satisfy this relationship should generally obtain the correct arc densities, which implies that the total path delay experienced by a vehicle is correctly distributed in space. This is an important property for obtaining correct path travel times when certain arcs are shared by several paths. Many DNL models proposed in the literature do not yield the fundamental diagram of traffic.

The proposed car following relationship is similar to a model proposed by Gipps, but with the constraints on acceleration and deceleration removed. The proposed model is thus quite similar to Cellular Automata (CA) traffic models, and in particular, to a deterministic CA model known as CA-184. An important difference is that time is continuous in the proposed model, while it is discrete in CA traffic models, which makes the former considerably easier to calibrate against empirical data. The proposed model is found to have a very unique property. Upon inverting the trajectories – from $x(t)$ to $t(x)$ – it is found that the model can be solved by considering only the values of $t(x)$ for each vehicle at those positions in the network where sources of delay are found. Because the sources of delay – also called the mechanisms of congestion – in a network are generally found at the nodes, either due to traffic signals or highway merges, the solution of this model can be defined solely in terms of the time at which each vehicle crosses each node on its path. The solution thus does not require the discretization of space between nodes,

and for this reason is referred to as an *arc-based* solution. The algorithm is a discrete-event – as opposed to discrete-time – procedure. This type of procedure is typically used for simulating queueing models. This model is the underlying mechanism of flow propagation for a new DNL model.

In order to define a DNL model, two other questions must be addressed. One question is how to model the traffic behaviour at nodes. At a node, the demand flows on the upstream arcs may exceed the supply on the desired downstream arcs. A *node model* must be defined that determines how the supply is distributed to these demands. The second problem is analogous to the first, but pertains to multi-lane arcs. In real networks, specific lanes on a road must often be used in order to execute certain movements at intersections. As a result, vehicles may be obliged to change lanes in order to stay on a specific path. These lane changes may result in a demand flow for a lane that exceeds the supply it can offer. An *arc model* must be defined to determine how this supply is to be distributed. The arc model should also properly account for the reductions in arc capacity due to the interactions of lane-changing maneuvers of vehicles destined for different lanes.

The proposed traffic flow model was extended to the multi-lane case in such a way that the interactions between lanes could be addressed without losing the arc-based property of the solution. This model allows each vehicle only one lateral movement per arc, and requires the departure lane to be selected before a vehicle may enter an arc. The multi-lane model thus defines a mapping from the lane-based demand flows (by arrival and departure lane) to the arc variables (flow, density, and speed). The model was coded in C++ and tested on a small network where a high degree of lane changing was incurred on one arc. The results were found to compare very well with those obtained using the AIMSUN2 traffic simulation package, under a number of different supply/demand scenarios. The general conclusion was that the underlying constraints of the model – such as the restriction to a single lateral movement – were sufficient to capture the basic effects of lane changing on the arc variables. The execution time of the proposed model did not exceed one tenth of the execution time of AIMSUN2. The assumptions

underlying this model are consistent with a microscopic description of the trajectory of each vehicle. This allowed the model assumptions to be validated for a very small example that was solved by hand and animated on a second-by-second basis.

The node problem corresponding to a discrete-flow discrete-event arc model is defined, and a solution algorithm is proposed. The node model uses the well-known concept of gap acceptance to resolve the pair-wise conflicts that are identified. The proposed algorithm identifies the most logical vehicle to precede all others into the node given the results of the pair-wise comparisons. This problem is found to be non-trivial. The algorithm for resolving the node problem is embedded in a discrete-event network loading procedure, which ensures that the node problem is re-evaluated any time the inputs to this problem should change in the course of the simulation. Heuristics rules for choosing lanes in conjunction with the arc model are proposed, and the complete DNL model is tested on a small network consisting of 12 arcs and 12 nodes. The model represents a section of highway bounded by two ramps with a parallel service road, and is tested under time-varying demand and supply conditions. The results compare quite well with those obtained using the INTEGRATION traffic simulation package. Noticeable discrepancies occur in flow and density on arcs where a high degree of lane changing is combined with delays propagating from downstream arcs. Further testing of the model using real-world data is recommended. The execution time of the proposed DNL model is roughly one two-hundredth of that required by INTEGRATION.

Key words: dynamic network loading, traffic flow theory, car-following models, queueing theory, traffic simulation, dynamic traffic assignment.

Table of Contents

1 **Introduction**
3 Fluid Dynamic Models
7 Car Following Models
10 Cellular Automata
14 Queueing Models
15 Discussion
16 Outline and Contributions of the Thesis

20 **Behavioural Car Following Models**
20 Introduction
24 Steady State Properties
25 The Quadratic Deceleration Trajectory
28 The Logarithmic Deceleration Trajectory
31 Stability of the Behavioural Models
33 The Linear Safety Parameter
36 Discussion and Conclusions
37 References

39 **From Traffic Flow to Queueing Theory**
39 Introduction
41 Model Statement
45 The Cell-Based Solution
47 The Arc-Based Solution
48 The Equivalent Point Queue
52 The Fundamental Diagram
54 Conclusions
55 References
57 Figures

60 A Discrete Flow Arc Model of Traffic Dynamics Based on the Space-Time Queue

60 Introduction

64 Model Definition

74 A Numerical Example

78 Simulation Results

81 Conclusions

82 References

84 A Discrete Flow Model For Dynamic Network Loading

84 Introduction

85 The Node Model

92 A Discrete-Event Network-Loading Algorithm

95 A Simulation Example

102 Conclusions

103 References

105 Conclusions and Recommendations

110 Bibliography

List of Figures

Behavioural Car Following Models

22 Safe Deceleration To Stop Diagram
27 Steady State Curves For The Quadratic Deceleration Trajectory
28 Logarithmic Deceleration Trajectory
30 Steady State Curves For The Logarithmic Deceleration Trajectory
35 Steady State Curves With The Linear Safety Parameter Only

From Traffic Flow to Queueing Theory

57 Space-Time Trajectories
57 The "Point" Space-Time Queue
58 Space-Time Domain For Constrained Trajectories
58 Space-Time Areas For Constrained Trajectories
59 The Fundamental Diagram

A Discrete Flow Arc Model of Traffic Dynamics Based on the Space-Time Queue

62 Lane Changing
68 Delay Rules for Lane Changing
76 Animation of Vehicle Trajectories
77 Calculations for Animation Example
79 Test Network
80 Flow vs. Time (non-signalized)
80 Density vs. Time (non-signalized)
80 Flow vs. Time (signalized)
80 Density vs. Time (signalized)
81 Average Flow vs. Maximum Speed
81 Average Density vs. Maximum Speed

A Discrete Flow Model For Dynamic Network Loading

88 First Node Conflict Example

89 Second Node Conflict Example

89 Third Node Conflict Example

90 Fourth Node Conflict Exmple

97 Test Network

98 Three-Segment Flow-Density Relationship

100 Outflow vs. Time

101 Density vs. Time

Acknowledgements

I am greatly indebted to Professor Michael Florian for his guidance, support and for the unique opportunities he has given me to learn about the theory and practice of transportation network modeling. A research project with the Ministry of Transportation of Quebec involving the AIMSUN2 micro-simulator was a memorable learning opportunity, made all the more so by the involvement of Professor Florian, and Professor Jaime Barceló of the Universitat Politècnica de Catalunya. Professor Barceló was a pleasure to work with, and was very generous with his time on two visits I made to Barcelona. I was also very fortunate to have had the chance to collaborate with Vittorio Astarita, of the Universita degli Studi della Calabria, during his many visits to Montreal. Without doubt, I gained much more than I contributed to the papers that were written with him.

Nicolas Tremblay, Shane Velan, and Karim Er-Rafia made the effort devoted to several optimization courses considerably more profitable and enjoyable. They were the best of partners, and their solutions were often more enviable than my own. I have had the pleasure of working with – and learning from – them on various research problems as well. I would like to thank Professor Felisa Vásquez-Abad, who was kind enough to write me a letter of recommendation that contributed to a successful scholarship application. I would also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC), for awarding me a graduate scholarship that greatly facilitated the successful completion of my studies.

To my family – without whose support and guidance I would have never undertaken this degree – your influence is profoundly felt and appreciated. And to Anna Bragina – my companion, my tutor, my listener, my advisor, my best friend – words cannot adequately express my gratitude, any more than they can describe the patience, encouragement, and support you have offered me during these years. To repay this debt could take a lifetime, and I would be privileged to have the opportunity to do so.

Michael Mahut
Montreal
August, 2000

Introduction

This work presents a time-dependent model of traffic flow for general networks, or *dynamic network-loading* (DNL) model. A DNL model is a mapping from the time-dependent path demand flows to the time-dependent arc variables of flow, density, and speed. Models of this kind are an integral component of the *dynamic traffic assignment* (DTA) problem, which is the determination of the time-dependent path choices of vehicles in a network given the time-dependent demands from each origin node to each destination node. In general, these path choices are functions of the arc variables mentioned above. Many different approaches to the DNL problem have been developed in the past. These approaches are distinguished primarily by the way they represent traffic flow, which may be continuous or discrete. Solution techniques for these models are usually discrete-time methods, where the state of the network is determined over short time intervals of constant length. For most DNL models, the DTA problem is solved in an iterative fashion. An initial set of time-dependent paths is chosen, the DNL model is executed, and the path choices for the next iteration are adjusted as a function of the path times obtained (Smith and Wisten, 1996; Er-Rafia, 2000). Another approach to dynamic traffic assignment is based on a feedback mechanism within a single iteration of the DNL model. This approach – which is a feature of some commercial traffic simulation packages – uses current network conditions as an input to the path choices at the origins. Although the iterative approach tends to converge in empirical tests (*ibid.*), convergence is not easily obtained using the feedback method.

Discrete-flow DNL models can be further broken down by the representation of space, which may also be either discrete or continuous. Discrete-flow traffic models based on car-following (CF) logic are defined on continuous space (Rothery, 1999; Gabard 1991), while those based on the cellular-automaton (CA) paradigm use a discrete notion of space (Nagel, 1996A). In the latter case, the space discretization is relatively fine, equal to the length of a single vehicle. Both CF and CA-based models explicitly represent the individual lanes of a roadway. All solution methods for discrete-flow models in the literature are discrete-time approaches, with the exception of direct applications of queueing methodology to modeling traffic flow. It should be noted that traffic models defined in this way do not respect the well-known empirical relationship between traffic flow and density, often referred to as the *fundamental diagram* of traffic (May, 1990).

Continuous-flow DNL models are generally either defined over the entire length of an arc, or on arc segments sometimes referred to as cells. The family of *analytical* – or *link-based* – models belongs to the former case, while the *macroscopic* models belong to the latter. The analytical models are inspired from the solution to the static traffic assignment problem, and are usually based on travel time functions. These models are not based on – nor do they yield – the fundamental flow-density relationship. This is in contrast to the macroscopic models, which are closely related to the theory of fluid dynamics. Continuous-flow approaches do not explicitly model the lane-to-lane interactions that occur in multi-lane flows (lane changing), and are usually applied to the entire width of a roadway.

The different approaches have varying degrees of practical application, due to the varying limitations of each approach in modeling traffic flow realistically. It is generally believed that the most realistic models are those that use car-following logic, referred to as micro-simulation models. These models are based on a very detailed representation of the physical network, and employ a relatively large number of traffic parameters and heuristics. This renders these models difficult – if not impossible – to calibrate for large-scale networks. Moreover, these models are relatively slow in terms of computation time. This last point is significant in the context of real-time applications of dynamic network loading and dynamic traffic assignment. Applications of this kind include the determination of route-guidance information (e.g., to be posted on variable message signs on highways), and adaptive network control. Even when the DTA logic is based on a single run of the DNL model, micro-simulation models are currently not fast enough on the hardware typically available to be used for such applications on large-scale networks. Although computer memory and speed are incessantly increasing, this does not diminish the potential usefulness of a relatively simple DNL model with a degree of realism that approaches that of the micro-simulation models. In particular, a model with fewer parameters, and based on a less-detailed definition of the network, should be easier to execute and calibrate, especially for medium to large-scale networks.

In the following, a brief review of traffic flow models is presented. The term *traffic flow* is used to imply that a model yields the fundamental flow-density relationship of traffic. The models reviewed include the macroscopic (fluid dynamic), car-following, and cellular automata models. Although the analytical methods are not discussed here, a short section on the application of queueing theory to modelling traffic is included at the end of the chapter.

Fluid Dynamic Models

Theories relating to the description of vehicular traffic as a continuous medium are defined upon the variables of flow (q), density (k), and speed (v). Due to the relationship $q = kv$, only two of these variables are independent. The first fluid dynamic theory proposed for describing traffic flow was that of Lighthill and Whitham, and Richards (Lighthill and Whitham, 1955; Richards, 1956) – hereafter referred to as the LWR theory. The underlying hypothesis of this theory is that flow is strictly a function of density: $q = Q(k)$. This is equivalent to saying that speed is strictly a function of density: $v = V(k)$. One may thus write the fundamental conservation law of fluid dynamics as follows (*ibid*):

$$\text{(1)} \qquad \frac{\partial k}{\partial t} + \frac{\partial (kV(k))}{\partial x} = 0,$$

or alternatively as:

$$\text{(2)} \qquad \frac{\partial k}{\partial t} + C(k)\frac{\partial k}{\partial x} = 0$$

where: $C(k) = dQ(k)/dk$.

The theory states that slight changes in flow are propagated upstream along *kinematic waves*, whose velocity (relative to the road) is given by $c = C(k)$, i.e., the slope of the graph of flow vs. density. These waves represent regions of constant density, and move along straight lines, called *characteristics*, in space-time. The model is solved by determining the function $k(x,t)$, given the appropriate boundary conditions. This can either be done using graphical techniques (Lighthill and Whitham, 1955) or numerically (Daganzo, 1994; Lebacque, 1996).

Different variants of the model can be characterized by the relationship $v = V(k)$, or equivalently, $c = C(k)$. To be realistic, $C(k)$ must be monotonically decreasing, or at least non-increasing, with increasing density. The original proposition of the LWR model used a well-known relationship between speed and density, due to Greenshields (1935). This is a linear relationship, defined as:

$$(3) \qquad v = v^{\max}\left(1 - \frac{k}{k^{\max}}\right)$$

where: $v^{\max}$ = the maximum speed of the traffic

$k^{\max}$ = the maximum (jam) density of the traffic

In this case, the complete model – by substitution of (3) into (1) – is given by:

$$(4) \qquad \frac{\partial k}{\partial t} + \left(v^{\max}\right)\frac{\partial k}{\partial x} - \left(\frac{v^{\max}}{k^{\max}}\right)\frac{\partial k^2}{\partial x} = 0$$

It has been noted that with the addition of a term involving the appropriate diffusion constant, (4) becomes a nonlinear diffusion equation known as the Burgers equation (Nagel, 1996A). Another relationship that has been used in conjunction with the LWR theory is the following piecewise linear relationship between flow and density:

$$(5) \qquad q = \begin{cases} kv^{\max} & for \quad k \le k^* \\ \alpha\left(k^{\max} - k\right) & for \quad k \ge k^* \end{cases}$$

where: $\alpha = \dfrac{k^* v^{\max}}{k^{\max} - k^*}$

The density k^*, called the *critical density*, corresponds to the maximum flow. The following equivalent relationship for $C(k)$, in conjunction with (2), defines the LWR model based on (5):

$$(6) \qquad C(k)=\begin{cases} v^{\max} & for \quad k \le k^{*} \\ -\alpha & for \quad k \ge k^{*} \end{cases}$$

In this special case of the model, there are thus only two wave speeds. This version of the model is of some interest, as it has been used in the proposition of a simplified theory of kinematic waves (Newell, 1993), based on determining the cumulative number of vehicles that pass fixed positions of interest in linear networks. These networks permit inflow and outflow at intermediate nodes, thus producing the necessary congestion. A unique property of this method is that it does not require the discretization of space over the length of an arc. If no flow is entering or leaving the network, the cumulative count may be determined directly for whatever positions are of interest. The model must also be solved at all positions of inflow and outflow. Solving the model in this way is done using graphical techniques. It should be noted that these techniques are considerably simpler that those originally proposed for solving the LWR model with a non-linear flow density relationship. An approximate solution to Newell's model may be calculated using an automated (computer) algorithm defined on a space-time lattice. Again, the algorithm is remarkably simple. Newell also discusses at some length the suitability of this two-linear-segment flow-density model for describing real traffic (*ibid*).

At about the same time, another simplified model – known as the cell transmission model – was proposed, this time based on a three-linear-segment flow-density relationship Daganzo, 1994. The third segment defines a maximum flow constraint, somewhere below the maximum flow defined for the two-segment model. The linearization of the flow-density relationship simplifies the numerical solution algorithm, which is defined on a space-time lattice. This model thus requires the discretization of space into cells. In principle, the cell transmission model can be applied to general networks with the addition of appropriate node models. A piecewise-linear flow density model, used in conjunction with the conservation equations, is also used in another dynamic assignment model (Chang *et. al,* 1985; Mahmassani, Hu and Jayakrishnan, 1995). As with all continuous-flow approaches, lane-interactions are not modeled explicitly in either the simplified theory of Newell or the cell-transmission model of Daganzo.

By assuming that speed is only a function of density, the LWR theory predicts that vehicles react instantaneously to changes in density. This is equivalent to the assumption that acceleration is unbounded, i.e., that changes in velocity may be instantaneous. The theory is in fact a special case of a more complete fluid dynamical formulation, with the momentum equation left out. The momentum equation may be stated as follows (Prigonine and Herman, 1971):

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} = \frac{force}{mass} \tag{7}$$

Using this relationship, Payne proposed a model that overcame the basic problem of instantaneous changes in velocity of the LWR theory (Payne, 1971). The model contains a relaxation term – which limits how quickly the flow may adapt to changes in density – as well as an anticipation term, which "looks ahead" to the downstream density. An interesting mathematical motivation for the anticipation term, based on centering the local density for a vehicle halfway between itself and the vehicle in front of it, is due to Kerner (Nagel, 1996A). Payne's momentum equation thus reads as follows (*ibid.*):

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} = \frac{1}{\tau}\left[V(k) - v\right] - \left(\frac{c_0^2}{k}\right)\frac{\partial k}{\partial x} \tag{8}$$

where: c_0 = the wave speed of the linearized flow density relationship

This model is known to predict wholly unrealistic traffic conditions under certain circumstances (Ross, 1988). For example, when freely flowing traffic comes to a red light, or the back of a stationary queue, density increases without limit before relaxing down to its equilibrium value. Although it may be a straightforward matter to simply add a density constraint, other causes of unrealistic behaviour in the model are known that cannot be solved very easily, if at all (*ibid*). With the addition of a diffusion term, one obtains the momentum equation of Kuhne (1993), and Kerner and Konhauser (1993), which reads:

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} = \frac{1}{\tau}\left[V(k) - v\right] - \left(\frac{c_0^2}{k}\right)\frac{\partial k}{\partial x} + \upsilon\frac{\partial^2 v}{\partial x} \tag{9}$$

where: υ = a diffusion constant

The model of Kuhne *et al.* is the only fluid dynamical model known to produce spontaneous jam formation. Analysis of this model requires the methods of nonlinear dynamics (*ibid*).

Car Following Models

Car following models describe the way in which the space-time trajectory of one vehicle may depend on that of the vehicle in front of it. As such, these models apply to the study of vehicles traveling on a single lane of a road. These models are defined upon the notion of discrete flow, moving through continuous space and continuous time. The models are solved by discretizing time, leaving space as the only continuous variable. One of the first such models, proposed by Pipes, stated simply that the follower's speed is proportional to the spatial separation (spacing) between the two vehicles (Pipes, 1953):

$$(10) \qquad v_n(t) = \alpha\left(x_{n-1}(t) - x_n(t)\right)$$

where: n = the vehicle's identifier: vehicles are numbered, starting with 1, in the order in which they cross some fixed position in space

$v_n(t)$ = the speed of vehicle n at time t

This model implies an instantaneous reaction on the follower's part, to changes in the spacing between himself and the vehicle in front of him. To improve the model's predictions, a lag term, τ, was added in order to account for the follower's reaction time. The general form of the model became:

$$(11) \quad v_n(t+\tau) = \alpha\left(x_{n-1}(t) - x_n(t)\right)$$

Further development led to an entire family of models, sometimes referred to as the *stimulus-response* car following models (Gerlough and Huber, 1975). The stimulus is the right hand side of the equation, and the response is the left hand side of the equation. These models are of the following general form (*ibid.*):

$$(12)\quad a_n(t+\tau) = \alpha \frac{v(t)^m}{\left[x_{n-1}(t) - x_n(t)\right]^l}\left[v_{n-1}(t) - v_n(t)\right],$$

where m and l are constants. An important result concerning these models was the derivation of the associated steady state relationships between flow, density, and speed. The values of m and l could thus be determined by fitting these curves to empirical data. Many other car following models have been developed along similar lines. The underlying feature of all of these models is that they are based on a differential equation relating the trajectory of vehicle n (the *follower*), at time $t+\tau$, to some information regarding the trajectories of both vehicles n and n-1 at time t. Vehicle n-1 is referred to as the *leader*.

Another general type of car following model is derived from (11). Applying a first order Taylor series expansion to the left hand side, and rearranging terms, leads to the relationship (Whitham, 1990):

$$(13)\quad a_n(t) = \alpha\left(x_{n-1}(t) - x_n(t)\right) - v_n(t)$$

Other models of this general variety have been developed as well (Gerlough and Huber, 1975). These models are thus based on a differential equation involving both the leader's and follower's trajectories, at the same instant in time.

Gipps proposed a model that employs a second order approximation of the distance traveled, rather than a first order approximation of the change in speed (Gipps, 1981). Moreover, the model is based on taking the minimum of two inequality constraints, rather than using a single equality relationship. The model is of the general category of (14): the follower's velocity at time $t+\tau$ is calculated as a function of both its own, and its leaders, trajectories at time t. The general statement of the model is as follows:

$$(14)\quad v_n(t+\tau) = MIN\left\lfloor v_n^{dm}(t), v_n^{sp}(t)\right\rfloor$$

where: $v_n^{dm}(t)$ = the maximum speed allowed by the acceleration constraint, which may be thought of as the vehicle's *demand speed*

$v_n^{sp}(t)$ = the maximum speed allowed by the safety constraint, which may be thought of as the prevailing *supply speed*

The supply speed is derived from an equation that yields the maximum speed that vehicle n may have at time $t+\tau$, supposing that vehicle n-1 begins to decelerate at time t. This equation is derived from the following inequality constraint:

$$(15)\quad x_n(t) + \left[v_n(t) + \frac{v_n(t+\tau)\tau}{2} \right] + v_n(t+\tau)\theta - \frac{v_n(t+\tau)^2}{2b_n} \leq x_{n-1}(t) - \frac{v_{n-1}(t)^2}{2\hat{b}_{n-1}} - \lambda_{n-1}$$

where: τ = the response time of a driver/vehicle

$x_i(t)$ = the position of vehicle i at time t

$v_i(t)$ = the velocity of vehicle i at time t

λ_i = the minimum distance between the front bumpers of two vehicles at zero speed, called the *effective vehicle length*

b_i = the deceleration rate of vehicle i

$\hat{b}_i$ = the deceleration rate of vehicle i, as estimated by vehicle $i+1$

θ = a user defined parameter

The term in square brackets is a second order approximation for the distance traveled by the follower over the time lag, τ. This constraint is derived from the assumption that vehicles decelerate at a constant rate. The actual rate at which the leader will decelerate is not considered to be information that the follower could actually know, hence it is considered to be an estimated value, $\hat{b}_{n-1}$.

Setting (15) to equality, and solving for $v_n(t+\tau)$, yields the supply speed, $v_n^{sp}(t+\tau)$. Given that the temporal discretization step used to solve the model is τ, the acceleration constraint proposed by Gipps is of the following general form:

$$v_n^{dm}(t) = v_n(t-\tau) + a_n^{max}\left(v_n(t-\tau)\right)\tau \tag{16}$$

The acceleration constraint serves the purpose of maintaining realistic limits on the vehicle's acceleration. This constraint also implicitly contains the following constraints on speed:

$$\begin{cases} v_n(t) \leq v^{max} \\ v_n(t) \geq 0 \end{cases} \tag{17}$$

This is accomplished by having $a_n^{max}\left(v_n(t)\right) \to 0$ as $v_n(t) \to 0$, and also as $v_n(t) \to v^{\max}$. A negative lower bound on the vehicle's acceleration is also needed, but this bound is respected implicitly by (15).

Cellular Automata

Cellular automata (CA) are unlike fluid dynamical models and car following models, in that there is no distinction between the formal statement of the model and its solution. The models discussed thus far are defined on continuous space, and time – it is only in the solutions, or *simulations*, of these models, in which one or both of these dimensions are discretized. CA are simulation models, where both space and time, as well as flow, are discrete. For this reason, they are sometimes called "particle-hopping models", because they describe the behaviour of discrete particles hopping on a space-time grid. One- and two-dimensional-space CA models for vehicles, as well as pedestrian traffic, have been developed – see (Nagel, 1996A) and references therein. Due to the discrete nature of space, time, and flow, the notion of speed (or velocity) in these models is necessarily discrete as well. These models typically use between one and five discrete (positive) speeds. The maximum speed is denoted by $v^{\max}$.

A fairly primitive one-dimensional model of this types dates back to 1955 (Gerlough, 1956). The main feature of this work was a single-bit representation of each vehicle – the traffic dynamics, *per se*, left much to be desired. This approach remained undeveloped for some 30 years, when the same one-bit approach was extended with somewhat more sophisticated dynamics (Cremer and Ludwig, 1986). This work exploited byte and word-based Boolean operations to achieve high computational efficiency, but lacked proper analysis and testing of the traffic flow model. The one-bit representation has since been abandoned, due to the need to store more information – such as routes – with each vehicle.

More recently, a stochastic CA for one-dimensional vehicular traffic has been proposed (Nagel and Shreckenberg, 1992). The model is referred to as the Stochastic Traffic Cellular Automaton (STCA). The STCA uses a value of $v^{\max} = 5$, and is defined by algorithm *STCA*, below. Essentially, this model defines the maximum distance that vehicle i may travel in the next time step, as a function of the position of vehicle i-1 in the current time step. There is a lag effect in this model due to the simultaneous updating of the vehicles: the cause (or stimulus) is in one time step, while the effect (or response) is seen in the next time step. The randomness introduced by the variable p_{noise} is responsible for such model properties as the critical density, k^*, that separates the uncongested and congested flow regimes, as well as the maximum flow observed at bottleneck conditions. A substantial part of the discussion concerning the STCA has to do with the probability of spontaneous jam formation (congestion). The critical density is defined in this model as the density below which there is zero probability of such a jam lasting an infinitely long time. Another related property of the STCA is the phenomenon of jams breaking up into regions of laminar flow between high-density pockets. It should be noted that analytical results regarding these phenomena have not been found for the STCA – the model remains, above all, a *simulation* model. Due to the stochastic nature of jam formation in the STCA, this model shows the *bi-stability* property. This refers to the fact that laminar flow may exist well beyond the average value of k^*, but will not remain laminar at these densities for an infinite period of time. This property is somewhat complementary to the definition of k^* given above. Another property of the STCA is that it has unbounded deceleration.

The deterministic limit of the STCA, along with some related deterministic models, has also been investigated (Nagel and Herrmann, 1993). The deterministic limit of the STCA is defined by setting the randomization probability, p_{noise}, to zero. This model was found to produce "...the rather simple situation of a low density phase with particles having maximum velocity, a high density phase of low velocity waves, and a simple transition between the two" (*ibid*). In addition to the simultaneous updating approach, this model was also evaluated using sequential updating from left to right, and from right to left. Such models are no longer CA, but are still valid particle hopping models. The left-to-right update was found to yield the same result as the simultaneous update, while the right-to-left update produced markedly different behaviour. This is simply because the former maintains the lag effect, while the latter does not.

Algorithm *STCA*

```
1. Velocity update: for all particles i simultaneously, do the following:

IF $(v_i \geq gap_i)$                (deceleration due to close following)
    $v_i = \begin{cases} gap_i - 1 & \textit{with probability } p_{noise}, \textit{ if possible } (v < 0 \textit{ not allowed}) \\ gap_i & \textit{else} \end{cases}$
ELSE IF $(v_i < v_{\max})$           (acceleration)
    $v_i = \begin{cases} v_i & \textit{with probability } p_{noise} \\ v_i + 1 & \textit{else} \end{cases}$
ELSE                                 (free driving)
    $v_i = \begin{cases} v_{\max} - 1 & \textit{with probability } p_{noise} \\ v_{\max} & \textit{else} \end{cases}$
END IF

2. Movement step: move all particles i to $x_i(t+1) = x_i(t) + v_i$.
where:      $gap_i = x_{i-1}(t) + x_i(t)$
            $i - 1$ = the vehicle ahead of vehicle i
```

One can further simplify the deterministic version of the STCA by setting $v^{\max} = 1$. This model is known as CA-184, due to its equivalence with cellular automaton rule 184 in Wolfram's notation (Nagel, 1996B). It has been reported that the general behaviour of the model does not change due to this simplification (Nagel, 1996B). The only practical difference obtained by setting $v^{\max} = 1$ is that the acceleration is now unbounded as well. An important feature of CA-184 is that its fluid-dynamical limit is known. This limit is obtained by making the cells and time steps increasingly smaller while increasing the number of particles, such that the average density of particles remains constant (Nagel, 1996B). The fluid dynamical limit for CA-184 is defined by:

$$\begin{cases} q = kv^{\max} & if \quad k \le k^* \\ q = \dfrac{1-\lambda k}{\tau} & if \quad k \ge k^* \end{cases} \tag{18}$$

where: $$k^* = \frac{1}{\lambda + v^{\max}\tau}$$

All of the variables are as defined in the previous section on car following models. This is identical to (5), the two-linear-segment flow-density relationship. It may be conjectured that CA-184 is essentially equivalent to the LWR model proposed by Newell (Newell, 1993). The existence of only two distinct wave speeds in Newell's model – one for each of the two flow regimes – is consistent with the behaviour of CA-184. Moreover, both models implicitly allow instantaneous acceleration and deceleration. Like the LWR theory, CA-184 does not produce spontaneous traffic jams.

The equivalence of these models may also be considered as follows. LWR theory is based on the conservation of mass, and the fundamental assumption that flow (or speed, or wave speed), is strictly a function of density. In the special case of Newell's simplified model, it is sufficient to say that mass is conserved, and that there are exactly two wave speeds, as given by (6). It is obvious that CA models conserve mass. Thus, the only remaining condition to verify is that CA-184 implies, in a strict way, the same wave speeds as (6). Although this is not demonstrated here, it is a known property of CA-184 (Nagel, 1996B).

Queueing Models

A single-server FIFO (first-in-first-out) point queue model has been used successfully in estimating waiting times and queue lengths at isolated bottlenecks, such as vehicles waiting at a red light. Difficulties commonly arise in such applications, however, when the arrival process is non-stationary, and in particular when the demand exceeds the capacity of the system during some portion of the period in question. Although one may simulate such a queue quite easily, predictions such as waiting time and queue length are only applicable to the isolated intersection case. Even in the simplest possible network consisting of two sequential servers defining the entrance and exit of a single arc, it has been observed that point queues cannot be adequately applied without careful incorporation of wave phenomena (Daganzo, 1995A). Queueing theory has also been applied to modelling the process of pedestrians or vehicles crossing a road (Edie, 1973). The service process in these cases consists of the gaps between vehicles on the road to be crossed. Modelling this process also suffers from the inability to consider wave phenomena, i.e., when the distribution of the gaps depends on downstream conditions.

It should be noted that some additional constraints have been added to queueing models in order to satisfy some basic physical rules that apply to traffic flow. Rather than moving vehicles (or *customers*), instantaneously from one server to the next queue, a lag time may be used to offset these departure and arrival processes (Heydecker and Addison, 1996). This time would be equivalent to an estimate of the minimum time required to traverse a section of roadway between two intersections, where the intersections are of course the servers in the model. In this way, a maximum speed constraint is respected. Moreover, one may assign a physical length to each vehicle, which, along with the physical length of a section of road, yields an obvious (storage) capacity constraint for each queue. The addition of such constraints does not however incorporate the wave phenomena. The difficulty is that queueing models are incapable of describing how vehicles are spread out in space. This information is necessary in order to determine the spatial extension of a queue, which may often extend beyond a single arc. Spatial queue extension is thus required to determine when a vehicle enters and exits a given arc along its route. Accurate arc travel times, in turn, are necessary for applications such as dynamic traffic assignment (DTA). To spread vehicles out in space, however, requires the concept of density, which is not a feature of queueing theory.

Discussion

Many traffic models have been proposed in the literature. In the context of dynamic traffic assignment, where highly detailed dynamics and stochastic behaviour may be less important than the computational speed of the model, it is the simpler models that are of particular interest. In this regard, it is significant that two of the three major approaches to traffic theory – namely, macroscopic theory and cellular automata – yield a two-segment flow density relationship in a limiting, simplified case. Although this relationship is an *input* (rather than an output) in the LWR theory, it is known to yield less sophisticated dynamics than non-linear relationships (Velan, 2000), and thus truly represents a *simplified* theory as suggested by Newell (1993). It may be noted that in both the LWR theory and CA-184, positive and negative acceleration are unbounded. It may be conjectured that for the purposes of dynamic traffic assignment, it is sufficient for a DNL model to satisfy the two-linear-segment flow-density relationship.

Another question of importance concerns the features of a complete DNL model that lie beyond the theory of traffic flow. Such features include the modelling of nodes, as well as the lane interactions that are features of multi-lane CF and CA-based simulation models (Gipps, 1986; Rickert *et al*, 1997). Concerning node models, it should be noted that many applications of continuous flow models apply constraints at the node that are based only on the demand for, and the capacity of, each of the arcs downstream of the node. Without considering the geometry of the turnings at the node, these models capture only the congestion effects due to the merging of flows. Another important situation involves flows that *cross*, i.e., that share neither an upstream nor downstream arc at the node. This type of constraint is generally included in discrete-flow approaches. As mentioned above, continuous-flow models are not well suited to considering lane interactions within arcs, which are known in the traffic engineering literature to be nonnegligible as mechanisms of congestion (Transportation Research Board, 1998). It would seem that any improvements in the area of simplified traffic modelling – that might be useful in the context of DTA – should be focused on the discrete-flow approach in order to be well suited to the consideration of node and arc-based constraints that go beyond the basic traffic dynamics.

Outline and Contributions of the Thesis

The main objective of the thesis is to develop a traffic model that is well suited for dynamic traffic assignment (DTA) applications for large-scale networks. To be of practical use for large scale networks, it is important that the amount of input data required by a traffic model, and in particular the number of parameters needed to describe the traffic behaviour on each arc of the network, be kept to a minimum. To be of practical use within the context of an iterative DTA algorithm, it is important that a traffic model have relatively short execution times, even on large networks. For these two reasons this thesis is concerned with *simplified* traffic models, and with the trade-off between accuracy and computational effort in this type of model.

The body of the thesis is composed of four articles. The first article, entitled "Behavioural Car Following Models", was prepared during the course of a preliminary literature review on the topic of car following models. One contribution of the paper is a unique formulation of the well-known relationship between two sequential vehicles based on maintaining a safe distance between them, in which the variables describing the trajectory of *each* vehicle are referenced to a single point in time, but no two vehicles are referenced to the *same* point in time. It is found that this formulation more readily permits the derivation of some basic properties associated with car following models, such as the equilibrium relationship between density and flow and the property of asymptotic stability. Another contribution is the proposal of an alternative function for the deceleration trajectory of a vehicle. The equilibrium relationship resulting from the use of the proposed function is found to resemble very strongly the general shape suggested by empirical data, and hence is very well suited to calibration against such data. At the same time, the parameters of the equilibrium relationship represent physically meaningful quantities, such as reaction time, maximum speed and vehicle length, which allows such a relationship to be constructed for a road section when empirical flow-density data are not available. An earlier version of this paper was presented at the 78th Meeting of the Transportation Research Board (United States). The paper in its current form is under review for publication in *Transportation Science*, a journal that remains uncontested as the most prestigious in the field.

In the second article, "From Traffic Flow to Queueing Theory", an important contribution is made to the area of simplified traffic models through the discovery of a discrete-flow model that is essentially a special type of point queue whose resource is one-dimensional space-time, rather than one-dimensional time. The result is a queueing model that respects the fundamental diagram of traffic, which is the equilibrium relationship between the traffic variables of flow and density. This simple, one-lane traffic model can be thought of as the continuous-time counterpart of a deterministic cellular automaton traffic model. With respect to continuous-flow models, its closest counterpart is the simplified kinematic wave model proposed by Newell. The paper was recently presented at the 8th Annual Meeting of the EURO Working Group on Transportation – EURO is the Organization of European Operational Research Societies. Papers presented at the meeting are currently under consideration for a special issue of the European Journal of Operations Research (EJOR).

This new model provides the underlying structure for a more general model presented in the third paper, "A Discrete Flow Arc Model of Traffic Dynamics Based on the Space-Time Queue". The generalized model is designed to capture the effects of interactions between parallel lanes where lateral movements (lane changing) may occur. Designing such a model is a problem that has not been attempted before, as there are no other "one-lane" discrete-flow models – i.e., discrete-flow models of longitudinal dynamics – in the literature that do not require the longitudinal discretization of space. The proposed multi-lane model successfully captures the necessary interactions in a physically meaningful way without resorting to such a discretization, and hence requires remarkably little computational effort. Properly capturing the effects of lane-to-lane interactions, particularly on highways, is critical to the potential of such a model to be used for real-world applications. Hence, the high quality of the results produced by the model for the low computational cost required represents an important contribution to field of traffic simulation, and more generally to the domain of dynamic network loading. Dissemination of this paper will be pursued after the review has been received for the second paper. Publication will be sought in *Transportation Research B: Methodology*, a very well respected journal that has published much of the relevant literature on traffic modelling.

The application of the multi-lane arc model for general networks is the topic of the fourth article, "A Discrete Flow Model for Dynamic Network Loading". The first problem to be addressed in this paper is that of resolving conflicts between competing traffic streams at network nodes (intersections), referred to as the *node problem*. Because virtually all existing traffic models are solved using discrete-time procedures, while the proposed arc model is most efficiently solved using a continuous-time (or *discrete-event*) procedure, the design of an appropriate node model represented yet another problem that had not been addressed in the literature. Although the proposed node model uses the basic concept of gap-acceptance that has been developed and studied in the literature, more attention is paid here to how this concept is applied in the context of a model for general traffic nodes. In particular, it is found that this concept alone cannot prevent "dead-lock" situations and that additional constraints or heuristics are required in order to obtain realistic results. The proposed solution algorithm for general networks is based directly on the node model, and thus provides the modeler with a very clean separation between the arc model, node model, and solution algorithm modules. The proposed algorithm may thus be used for solving other continuous-time discrete-flow models, such as those based on the direct application of traditional queueing methodology. Moreover, the overall structure permits other types of node models to be tested very easily in conjunction with the proposed arc model. A very cursory description of the model, along with a discussion of the results presented in this paper, can be found in "A Comparison of Three Methods for Dynamic Network Loading", which has been accepted for presentation at the 80th Annual Meeting of the Transportation Research Board, and for publication in

Transportation Research Record. The meetings of Transportation Research Board are undoubtedly the largest annual meetings of the transportation research community, and the quality of the work accepted for publication in the associated journal (Transportation Research Record) is well recognized. Due to its focus on the procedural aspects of the model and the evaluation of its overall performance, the complete paper will be sent to *Transportation Research C: Emerging Technologies*. Although this journal is relatively new, it is well recognized in the field for publishing high quality work.

Behavioural Car Following Models

Michael Mahut, University of Montreal, Centre for Research on Transportation
Montreal, Canada

Abstract

This paper presents and analyzes a generalization of the car following model originally proposed by Gipps (1981). The generalization is with respect to the *safe-stopping*, or *safety* constraint. In addition to the usual quadratic expression for the deceleration trajectory, a second deceleration function is proposed, which yields a logarithmic relationship between speed and stopping distance. For each of these two deceleration trajectories, two models are defined : one for unconstrained vehicles, and another for constrained, or interacting, vehicles. Although the unconstrained models are non-linear inequalities – involving roots or logarithms – the constrained models are essentially linear equations, involving only powers of ± 1. This implies a significant increase in the efficiency of the model as a network becomes congested. The macroscopic (steady state) curves corresponding to these two deceleration trajectories are derived under the assumption of homogeneity of the traffic stream, and are found to be functions of four parameters: maximum speed, maximum density, driver response time, and a safety parameter. It is found that the response time and safety parameter can be derived from the capacity (maximum flow) and speed at capacity of the traffic. Sample calculations are found to yield very realistic response times. The macroscopic curve derived from the logarithmic function is a "single regime" curve, and is very well suited for calibration to empirical data. A necessary condition for the asymptotic stability of these models is defined, which yields a theoretical envelope on the fundamental traffic flow diagram, under the assumptions of asymptotic stability and homogeneity of the traffic. An extension of the quadratic model to include a linear safety parameter is also presented.

Acknowledgements

This work was supported in part by an individual operating grant from the Natural Sciences and Engineering Research Council of Canada (NSERC). The author would like to thank Professor Michael Florian, of the Center for Research on Transportation at the Université de Montréal, for his many useful comments on earlier drafts of this article.

1 Introduction

The car following paradigm investigated here is based on a very simple rule, which is already well known in the literature (Gerlough and Huber, 1975; Gipps, 1981; Krauss, 1998). The follower attempts to maximize his instantaneous speed subject to two constraints: an acceleration constraint, and a safety constraint. This may be stated as follows:

$$v_f(t) = MIN\,[v_f^{dem}(t), v_f^{sup}(t)] \tag{1.1}$$

where:
$v_f(t)$ = the follower's speed at time t
$v_f^{dem}(t)$ = the maximum speed allowed by the acceleration constraint, which may be thought of as a "demand speed"
$v_f^{sup}(t)$ = the maximum speed allowed by the safety constraint, which may be thought of as a "supply speed"

The acceleration constraint is a statement of a vehicle's physical limitations for speed and acceleration, as well as of a driver's desire for comfort. Essentially, it describes the trajectory of a vehicle that is free to accelerate to its maximum desired speed in the absence of downstream vehicles. It is not necessarily a constant value: it may be a function of the vehicle's speed, for example (see Gipps 1981). The safety constraint is a statement of how a vehicle's trajectory is affected by the next downstream vehicle. The remainder of this article will be concerned only with the safety constraint, and with the properties of the traffic flow that can be determined from it. In particular, it will be shown that both the steady state properties, as well as a necessary condition for asymptotic stability, can be determined directly from this constraint. A diagram illustrating the derivation of this constraint is shown in Figure 1, where the vehicles interact as follows: if the leader (indexed by *l*) should begin to decelerate to a stop at some time t, then the follower (indexed by *f*), starting at time $t+\tau$, must be able to decelerate to a stop in a safe way. The lag time τ is the follower's response time, while the time T is the moment the follower arrives to a stop. Hence, to avoid a collision, the follower's trajectory must satisfy the following inequality constraint, hereafter referred to as the *safety constraint*:

$$x_f(t+\tau) + \delta(v_f(t+\tau), \beta_f) + \lambda \le x_l(t) + \delta(v_l(t), \beta_l) \tag{1.2}$$

where:
τ = the response time of a driver/vehicle
λ = the minimum distance between the front bumpers of two vehicles at zero speed, referred to as the *effective vehicle length*
x_f = the follower's position
x_l = the leader's position

v_f = the follower's speed

v_l = the leader's speed

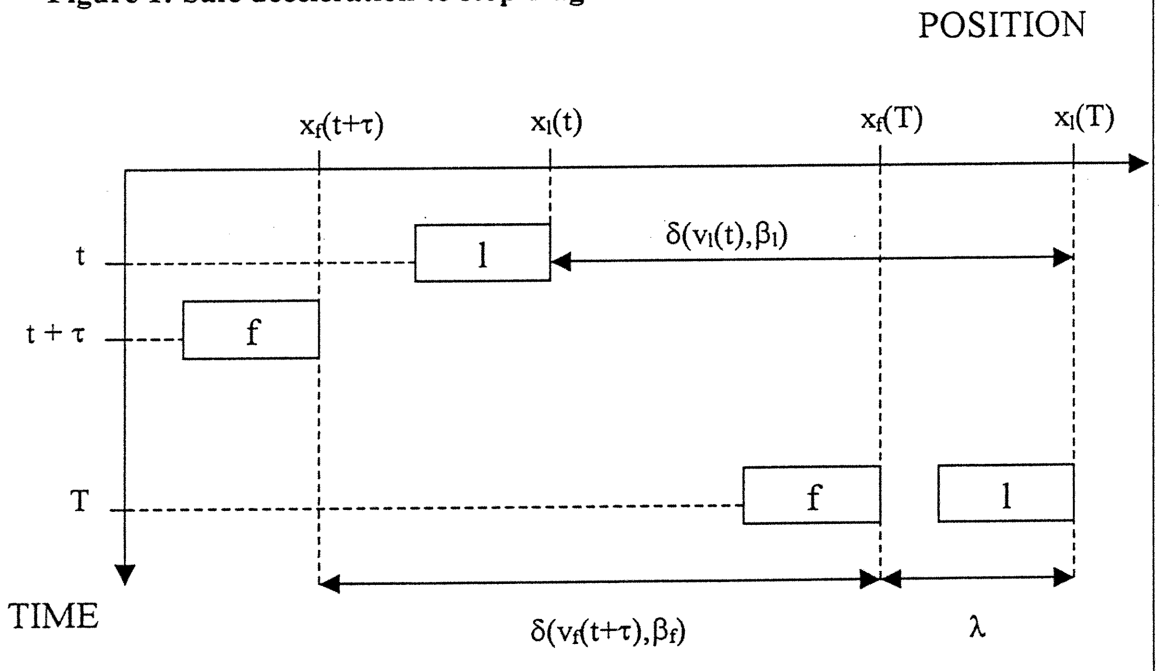
$\delta(v,\beta)$ = the deceleration distance function: the distance required to decelerate from speed v to a stop, with deceleration parameter β

β_f = the *desired* deceleration parameter: a parameter that reflects the way in which the follower would like to decelerate to a stop

β_l = the *anticipated* deceleration parameter: a parameter that reflects the way in which the follower anticipates the leader will decelerate to a stop

It should be noted that the traffic stream is assumed to be *homogeneous*. That is to say that the vehicle parameters, such as τ, λ, β_l, β_f, etc... will be assumed to be constant for all vehicles, as is typically done in order to simplify the analysis of the steady state and stability properties implied by a model. In particular, the speed at which vehicles attempt to travel, often referred to as the *desired speed,* or *free-flow speed,* is considered to be constant for all vehicles. This parameter is referred to here simply as the maximum speed, and is denoted by *V*. The state vector of a vehicle, at any moment in time, consists of its instantaneous position, speed, and acceleration: the latter is denoted by *a*(*t*).

Figure 1: Safe deceleration to stop diagram



A useful property of such a model is that all of the parameters involved correspond directly to physical or behavioural properties of the traffic stream: vehicle length, response time, maximum speed, and desired and anticipated deceleration. For this reason, Gipps (1981) referred to his model as a *behavioural* car following model – and hence the usage of the term here -- contrasting its parameters with those required by a different car-following paradigm, based on the notion of stimulus and response (Herman et al, 1959; Gazis et al, 1959), which do not necessarily correspond to physical or behavioural quantities. The advantage of such an approach is that most of these quantities are measurable, at least in principle. The more of these parameters that are directly measurable, the fewer degrees of freedom there are in the calibration process. Moreover, if one has even an intuitive idea of what are reasonable values for these parameters, due to their physical interpretation, one can say something about the plausibility of such a model.

Typically, in the formulation of a safety constraint equivalent to (1.2), all variables are referenced to the same time *t*, by estimating the follower's state vector at time $t+\tau$ as a function of his state vector at time *t* (Gerlough and Huber, 1975; Gipps, 1981; Krauss, 1998). The approach taken here is to allow the state vectors of *l* and *f* to be referenced to different points in time, i.e. to *t* and $t+\tau$, respectively. Although this will have no effect on the discretized solution formulas for the problem, it will permit the identification of an analytical solution for a special case. If the function $\delta(v,\beta)$ is invertible in v for a given β, one may solve (1.2) for the supply speed as follows:

$$v_f(t+\tau) \le \delta_v^{-1}\left(x_l(t) - x_f(t+\tau) + \delta(v_l(t),\beta_l) - \lambda, \beta_f\right) = v_f^{\text{sup}}(t+\tau). \tag{1.3}$$

This inequality will be referred to as the *unconstrained car following model*. It can be applied when the safety constraint is *inactive* (not binding), and can be used to determine the point at which the safety constraint becomes *active*, i.e. the point at which the follower's trajectory becomes *constrained* by its leader's. This is done by discretizing time into fixed increments, and finding a numerical solution to the right hand side of (1.3). Typically, one would estimate $x_f(t+\tau)$ by assuming constant acceleration from the last known state vector of vehicle *f*. When the safety constraint eventually becomes *active*, i.e. when:

$$x_f(t+\tau) + \delta(v_f(t+\tau),\beta_f) + \lambda = x_l(t) + \delta(v_l(t),\beta_l), \tag{1.4}$$

it is likely to stay that way for some time, under realistic traffic conditions. When this happens, the derivatives of the right and left-hand sides of (1.4) must also be equal over this interval. Assuming now that the function $\delta(v,\beta)$ is differentiable with respect to v, one may take the derivative of (1.4) with respect to time, which yields:

$$v_f(t+\tau)+\delta'_v(v_f(t+\tau),\beta_f)a_f(t+\tau)=v_l(t)+\delta'_v(v_l(t),\beta_l)a_l(t), \tag{1.5}$$

where the function $\delta'_v(v,\beta)$ is the derivative $d\delta(v,\beta)/dv$. What makes this relationship very convenient is that it can be solved easily for $a_f(t+\tau)$:

$$a_f(t+\tau)=\frac{v_l(t)-v_f(t+\tau)+\delta'_v(v_l(t),\beta_l)a_l(t)}{\delta'_v(v_f(t+\tau),\beta_f)}, \tag{1.6}$$

which can be interpreted as the maximum safe (or "supply") acceleration on the interval $(t, t+\tau)$. Thus, once the safety constraint becomes active, one may use (1.6) to calculate the appropriate values of $v_f^{sup}(t+\tau)$. This equation will hereafter be referred to as the *constrained car following model.* To solve (1.6) numerically, one would again have to use the last known state vector of f, this time to approximate $v_f(t+\tau)$. Although an analytical solution to (1.4) and (1.5) does not exist in general, one such solution does exist for a special case which is discussed and interpreted below.

2 Steady state properties

The objective now is to determine the steady state behaviour toward which the traffic will tend. The term steady state here is used in reference to a vehicle's state vector, and corresponds to the condition that its acceleration equals zero. Since the follower always attempts to maximize his instantaneous speed, he will achieve a constant speed only upon reaching the maximum speed of the traffic stream (V), or upon reaching a minimum safe distance behind a leader who is also moving at constant speed. In the case of the latter, the leader's speed must be less than the maximum speed (V). Clearly then, the safety constraint (1.2) is active for all steady state speeds less than the maximum speed of the traffic stream, where the follower's trajectory is given by the differential equation (1.5). Substituting $a_f(t+\tau)=a_l(t)=0$ into (1.5) yields:

$$v_f(t+\tau)=v_l(t). \tag{2.1}$$

Letting $v=v_f(t+\tau)=v_l(t)$, and noting that for $a_f(t)=0$,

$$(2.2)\qquad x_{\mathrm{f}}(t+\tau)=x_{\mathrm{f}}(t)+v\,\tau,$$

one may find the steady state spacing by substituting (2.1) and (2.2) into the active safety constraint (1.4), which yields:

$$(2.3)\qquad x_{\mathrm{l}}(t)-x_{\mathrm{f}}(t)=\delta(v,\beta_{\mathrm{f}})-\delta(v,\beta_{\mathrm{l}})+v\,\tau+\lambda,\quad \forall\, v<V.$$

Defining $s_0(v)$ as the steady state spacing as a function of speed, (2.3) becomes:

$$(2.4)\qquad s_0(v)=\delta(v,\beta_{\mathrm{f}})-\delta(v,\beta_{\mathrm{l}})+v\,\tau+\lambda,\qquad \forall\, v<V.$$

The steady state flow and density, as functions of speed, can be derived from (2.4) in the usual way, using the relationships $q = kv$, and $k = 1/s$, where q and k are the flow and density of the traffic stream, respectively. In the case where $v = V$, the only necessary condition is that the safety constraint (1.2) be satisfied, and thus all possible values of spacing that satisfy this inequality are possible steady state values for $v = V$. It can thus be seen that $\delta(v,\beta_f)$ - $\delta(v,\beta_l)$ represents a measure of the level of safety of the traffic stream, since the greater the distance between two vehicles, the less likely it is that a collision will occur.

3 The Quadratic Deceleration model

The question now turns to the nature of the deceleration distance function, $\delta(v,\beta)$. This is often posed as a constant deceleration (with time) trajectory (Gerlough and Huber, 1975; Gipps, 1981), representing a *worst case* stopping scenario. Under this assumption, the deceleration distance function, its inverse, and its derivative are defined as:

$$(3.1)\qquad \delta(v,\beta)=\frac{v^2}{2\beta},\qquad \delta_v^{-1}(x,\beta)=\sqrt{2\beta x},\qquad \delta_v'(v,\beta)=\frac{v}{\beta}.$$

The unconstrained car following model thus becomes:

$$(3.2)\qquad v_{\mathrm{f}}(t+\tau)\le\sqrt{2\beta_{\mathrm{f}}^{\mathrm{quad}}\left(x_{\mathrm{l}}(\mathrm{t})+\frac{v_{\mathrm{l}}(t)^2}{2\beta_{\mathrm{l}}^{\mathrm{quad}}}-x_{\mathrm{f}}(t+\tau)-\lambda\right)}.$$

where: β_f^{quad} = the absolute value of the follower's desired deceleration rate (m/s^2), and

β_l^{quad} = the absolute value of the leader's deceleration rate as anticipated by the follower (m/s^2).

Substituting (3.1) into (1.6), one obtains the following constrained car following model for the quadratic deceleration function:

$$a_f(t+\tau) = \beta_f^{quad}\left\{\frac{v_l(t)}{v_f(t+\tau)}\left(1+\frac{a_l(t)}{\beta_l^{quad}}\right)-1\right\} \tag{3.3}$$

An important characteristic of (3.3) is that it is essentially a linear equation, i.e., it involves only powers of ± 1, and is thus very inexpensive from a computational point of view. The steady state spacing is found by substituting (3.1) into (2.4), which yields:

$$s_0(v) = \frac{v^2}{2\beta_f^{quad}} - \frac{v^2}{2\beta_l^{quad}} + v\tau + \lambda, \qquad \forall\, v < V. \tag{3.4}$$

Introducing the notion of a *quadratic safety parameter*, ξ^{quad} defined as

$$\xi^{quad} = \frac{1}{2\beta_f^{quad}} - \frac{1}{2\beta_l^{quad}}, \tag{3.5}$$

the steady state spacing (3.4) becomes:

$$s_0(v, \xi^{quad}) = v^2\xi^{quad} + v\tau + \lambda, \qquad \forall\, v < V. \tag{3.6}$$

A useful property of this model is that appropriate values of τ and ξ^{quad} can be determined uniquely as functions of the capacity (Q) and speed at capacity (V_Q) of the traffic stream. One begins by defining λ as the inverse of the jam density (K):

$$\lambda = 1/K \tag{3.7}$$

Defining the steady state headway as $h_0(v) = s_0(v)/v$, one can then define:

$$V_Q = \operatorname{argmin}(h_0(v)). \tag{3.8}$$

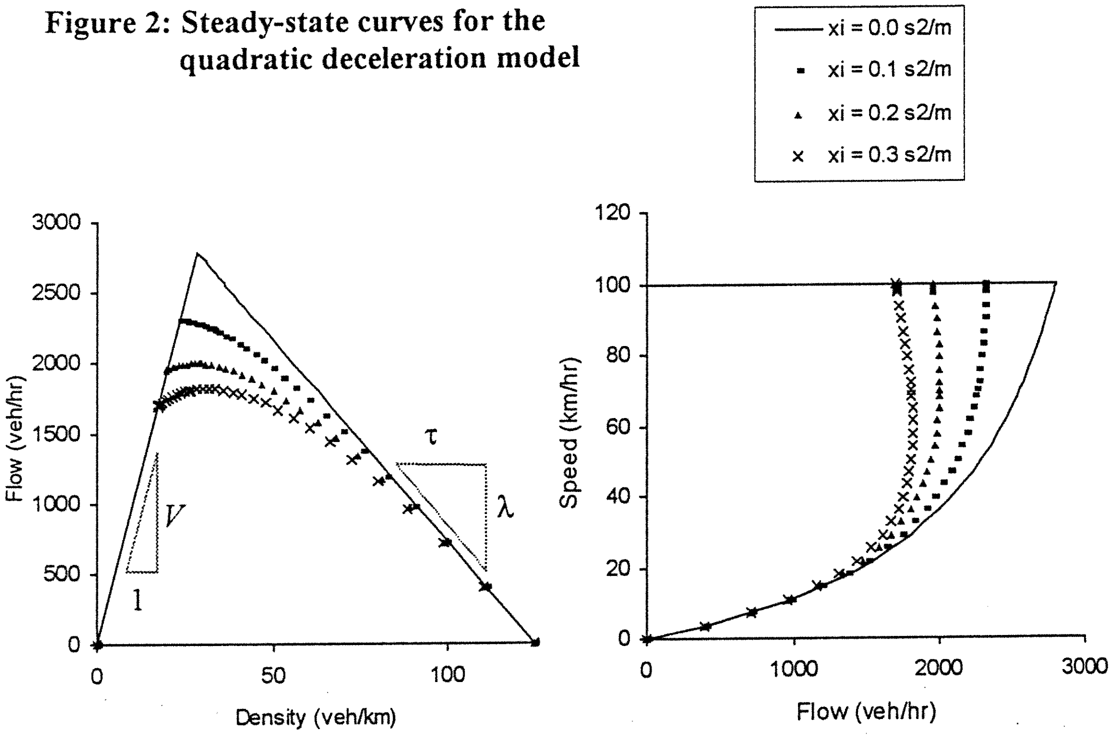
Setting the derivative $h_0'(v)$ equal to zero, one obtains:

$$(3.9) \qquad \xi^{\text{quad}} = \frac{\lambda}{V_Q^2}.$$

The relationship $h_0(V_Q) = 1/Q$ yields the following relationship for τ:

$$(3.10) \qquad \tau = \frac{1}{Q} - \frac{2\lambda}{V_Q}.$$

Figure 2: Steady-state curves for the quadratic deceleration model



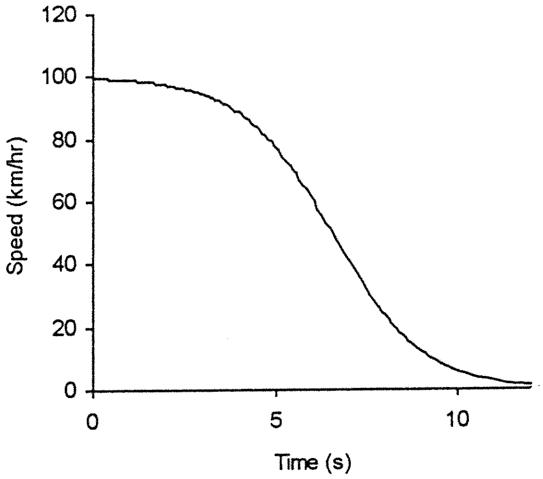
Although the traffic stream parameters allow one to uniquely define the safety parameter, ξ^{quad}, one can make no inference about the car following parameters β_f^{quad} and β_l^{quad} from this information. To calibrate the car following models (3.2, 3.3), one would have to estimate β_f^{quad} from observation (since one cannot directly measure β_l^{quad}), and then calculate β_l^{quad} as a function of β_f^{quad} and ξ^{quad}.

For the traffic flow parameters $V = 100$ km/hr, $K = 125$ veh/km, $Q = 2000$ veh/hr, and $V_Q = 70$ km/hr, the corresponding model parameters are $\lambda = 8.00$ m, $\tau = 0.977$ s, and $\xi^{\text{quad}} = 0.0212$ s^2/m. It is encouraging to note that for a realistic set of traffic stream parameters, the resulting value of τ is remarkably close to empirical measurements of response time. Figure 2 shows the macroscopic curves for this model when the parameters V, λ, and τ are held constant at 100 km/hr, 8.0 m, and 1.0 s, respectively, while the value of ξ^{quad} is varied. In the case of $v = V$, the steady state spacing may be any value that satisfies the safety constraint (1.2), and thus the steady state curve for this model is represented by a two-part function.

4 The Logarithmic deceleration model

By assuming different forms for the deceleration distance function, one can develop alternative behavioural models. A second model is developed here in this way, assuming that the follower is anticipating an *expected* deceleration trajectory on the leader's part, rather than a *worst case* trajectory. The form of the expected trajectory is posed here as:

Figure 3: Logarithmic deceleration trajectory



$$v(x) = V\left(1 - exp\left(-\beta x\right)\right) \tag{4.1}$$

where:
exp(y) = e^y
V = the maximum speed
β = a trajectory parameter

This function in fact describes an acceleration trajectory in x, but if one measures x backwards from the zero velocity position, it can equally well describe a deceleration trajectory to zero speed. Upon inverting (4.1);

$$x(v) = -\ln\left(1 - \frac{v}{V}\right)\frac{1}{\beta}, \tag{4.2}$$

one can describe the deceleration distance from a given speed (v) to zero speed, as a function of the maximum speed V and the parameter β. The corresponding formula for $v(t)$ is given by:

$$v(t) = \left(\frac{1}{V} + \exp\left(-\beta Vt\right)\right)^{-1} \tag{4.3}$$

Figure 3 shows a plot of equation (4.3). This curve provides a realistic form for the decelerating trajectory: deceleration is gradual at first, becomes more severe, and then gradual again as speed approaches zero. Defining the deceleration distance function and its derivative as:

$$\delta(v,\beta) = -\ln\left(1-\frac{v}{V}\right)\frac{1}{\beta}, \qquad \delta'_v(v,\beta) = \frac{1}{\beta(V-v)} \tag{4.4}$$

The unconstrained car following model is given by:

$$v_f(t+\tau) \le V\left[1-\exp\left\{-\beta_f^{\log}\left(x_l(t)-x_f(t+\tau)-\lambda-\ln\left(1-\frac{v_l(t)}{V}\right)\frac{1}{\beta_l^{\log}}\right)\right\}\right], \tag{4.5}$$

where: $\beta_f^{\log}$ = the follower's desired deceleration parameter (1/m), and
$\beta_l^{\log}$ = the leader's anticipated deceleration parameter (1/m),

The inverse function $\delta_v^{-1}(x,\beta)$ is given in (4.1), above. The constrained car following model for the logarithmic function becomes:

$$a_f(t+\tau) = \beta_f^{\log}\left(V-v_f(t+\tau)\right)\left(v_l(t)-v_f(t+\tau)+\frac{a_l(t)}{\beta_l^{\log}\left(V-v_l(t)\right)}\right). \tag{4.6}$$

One may note that the constrained model (4.6) is once again a linear equation. Introducing a *logarithmic safety parameter*, $\xi^{\log}$, defined as:

$$\xi^{\log} = \frac{1}{\beta_f^{\log}} - \frac{1}{\beta_l^{\log}} \tag{4.7}$$

The steady state spacing (2.4) corresponding to this trajectory becomes:

$$s_0(v,\xi^{\log}) = -\xi^{\log}\ln\left(1-\frac{v}{V}\right)+v\tau+\lambda, \qquad \forall\, v<V, \tag{4.8}$$

where $\lambda = 1/K$, as before. Setting once again the derivative of the steady state headway equal to zero, one obtains:

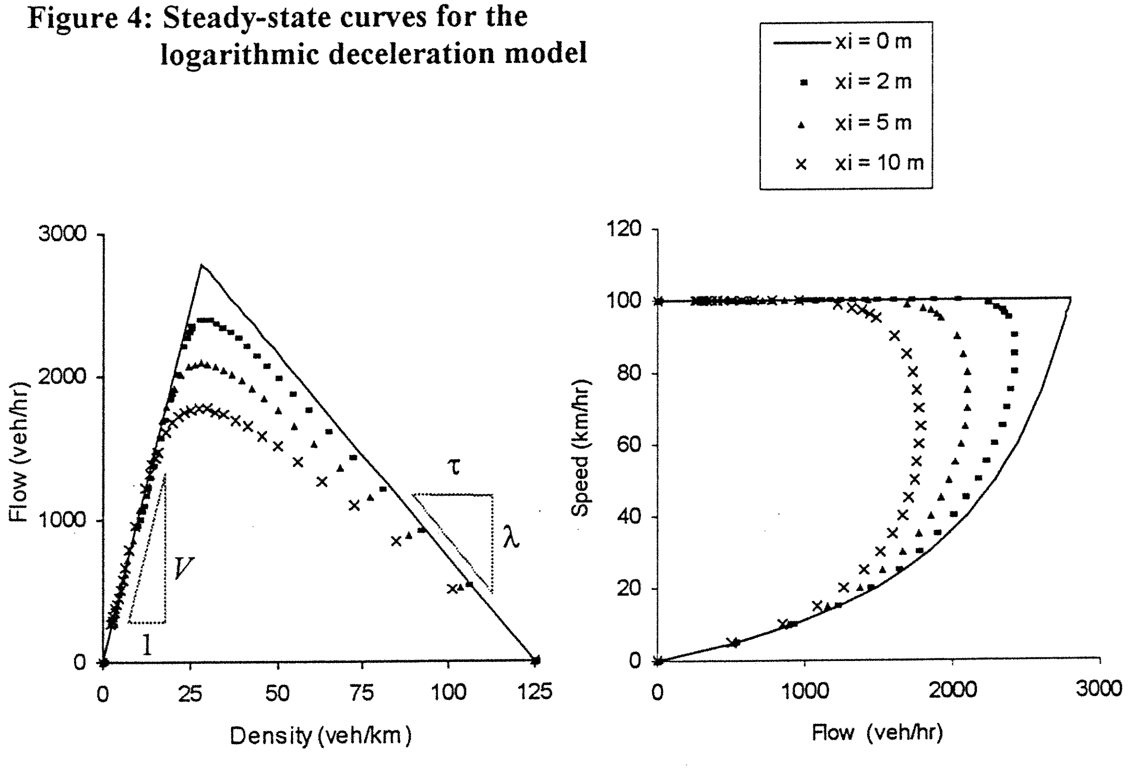
$$\xi^{\log} = \lambda\left(\ln\left(1-\frac{V_Q}{V}\right)+\left(\frac{V_Q}{V-V_Q}\right)\right)^{-1}. \tag{4.9}$$

The value of τ for this model is given by:

$$\tau = \frac{1}{Q} + \frac{1}{V_Q}\left(\xi^{\log} \ln\left(1 - \frac{V_Q}{V}\right) - \lambda\right), \tag{4.10}$$

which is again derived from the relationship $h_0\,(V_Q) = 1/Q$. For the traffic stream parameters $V = 100$ km/hr, $K = 125$ veh/km, $Q = 2000$ veh/hr, and $V_Q = 70$ km/hr, the resulting model parameters are $\lambda = 8.0$ m, $\xi^{\log} = 7.08$ m, and $\tau = 0.95$ s. Once again, the realistic value obtained for τ is very encouraging. Figure 4 shows the macroscopic curves for this model when the parameters V, λ, and τ are held constant at 100 km/hr, 8.0 m, and 1.0 s, respectively, while the value of $\xi^{\log}$ is varied. As with the quadratic model, one cannot fully calibrate the car following models (4.5, 4.6) from the traffic stream parameters. In this case, one would again have to measure the desired deceleration parameter, $\beta_f^{\log}$.

Figure 4: Steady-state curves for the logarithmic deceleration model



A major difference between the macroscopic curves for the quadratic and logarithmic models is that the latter is a "single regime" curve. This occurs even though vehicles can, in general, be in either constrained or unconstrained states, and is due to the fact that the deceleration distance goes to infinity as speed approaches the maximum speed. Hence, all steady state vehicles are in a constrained state, and the result is a single regime macroscopic curve.

5 Stability of the Behavioural Models

The notions of local and asymptotic stabilities of car following models have been examined in some depth for a family of models known as the "stimulus-response" models (Herman et al, 1959; Gazis et al, 1959). In the case of asymptotic stability, which is concerned with the way a perturbation moves through a platoon of vehicles, one analyzes a steady state platoon. In the case of a platoon at zero speed, this corresponds to an analysis of queue dispersion. For the steady state condition defined by (2.1), i.e., $v_f(t+\tau) = v_l(t)$, the constrained car following model (1.6) becomes:

$$a_f(t+\tau) = \frac{\delta'_v(v,\beta_l)}{\delta'_v(v,\beta_f)} a_l(t) \tag{5.1}$$

If one now refers to a platoon leader as '0', his follower as '1', and to the subsequent followers as 2, 3, ... n, one obtains:

$$a_n(t+n\tau) = \left\{\frac{\delta'_v(v,\beta_l)}{\delta'_v(v,\beta_f)}\right\}^n a_0(t), \tag{5.2}$$

where t is the moment in time the leader's trajectory is perturbed. It can be seen from (5.2) that the question of asymptotic stability is related to the value of $\delta'_v(v,\beta_l)/\delta'_v(v,\beta_f)$, since asymptotic stability implies that a perturbation to the flow diminishes in magnitude as it propagates upstream from one vehicle to the next. One can thus make the following statements regarding the value of $\delta'_v(v,\beta_l)/\delta'_v(v,\beta_f)$ as a necessary condition for asymptotic stability, instability, and the stability boundary:

$$\textit{asymptotically stable flow} \Rightarrow [a_n(t+n\tau) < a_0(t)] \Rightarrow \left[\frac{\delta'_v(v,\beta_l)}{\delta'_v(v,\beta_f)} < 1\right] , \tag{5.3}$$

$$\textit{asymptotically unstable flow} \Rightarrow [a_n(t+n\tau) > a_0(t)] \Rightarrow \left[\frac{\delta'_v(v,\beta_l)}{\delta'_v(v,\beta_f)} > 1\right] , \tag{5.4}$$

$$\textit{stability boundary} \Rightarrow [a_n(t+n\tau) = a_0(t)] \Rightarrow \left[\frac{\delta'_v(v,\beta_l)}{\delta'_v(v,\beta_f)} = 1\right] . \tag{5.5}$$

It can be further stated, from (5.2), that the more stable the flow, the smaller the value of $\delta'_v(v,\beta_l)/\delta'_v(v,\beta_f)$. Moreover, if one keeps β_f fixed, then the following is implied if $\beta1_l$ corresponds to more stable flow than $\beta2_l$:

$$(5.6)\quad \left[\frac{\delta'_v(v,\beta1_l)}{\delta'_v(v,\beta_f)} < \frac{\delta'_v(v,\beta2_l)}{\delta'_v(v,\beta_f)}\right] \Rightarrow \left[\left(\delta'_v(v,\beta_f)-\delta'_v(v,\beta1_l)\right) > \left(\delta'_v(v,\beta_f)-\delta'_v(v,\beta2_l)\right)\right]$$

In addition, if $d(\delta'_v(v,\beta))/d(\delta(v,\beta))$ is strictly positive, as is the case with the quadratic and logarithmic models presented above, it can be seen that:

$$(5.7)\quad \left[\left(\delta'_v(v,\beta_f)-\delta'_v(v,\beta1_l)\right) > \left(\delta'_v(v,\beta_f)-\delta'_v(v,\beta2_l)\right)\right]$$
$$\Rightarrow \left[\left(\delta(v,\beta_f)-\delta(v,\beta1_l)\right) > \left(\delta(v,\beta_f)-\delta(v,\beta2_l)\right)\right].$$

And furthermore;

$$(5.8)\quad \left[\left(\delta(v,\beta_f)-\delta(v,\beta1_l)\right) > \left(\delta(v,\beta_f)-\delta(v,\beta2_l)\right)\right] \Rightarrow \left[s_0(v,\xi1) > s_0(v,\xi2)\right],$$

where $\xi1$ and $\xi2$ are functions of $(\beta_f,\beta1_l)$ and $(\beta_f,\beta2_l)$, respectively. The right hand side of (5.8) implies that $\beta1_l$ yields a higher level of safety than $\beta2_l$, since a higher spacing implies higher safety. Hence, by (5.6-5.8), an increase in the stability is equivalent to an increase in the level of safety, for a given value of β_f.

For both the quadratic and logarithmic deceleration models, the value of $\delta'_v(v,\beta_l)/\delta'_v(v,\beta_f)$ is in fact *constant over all values* of v, and is equal to the ratio $\beta_l^{quad}/\beta_f^{quad}$, and $\beta_l^{log}/\beta_f^{log}$, respectively. As a result, the measure of stability suggested here is constant over all v given the choice of these parameters. It may be noted that the stability boundary case, which implies $\beta_l = \beta_f$, yields a value of zero for the quadratic and logarithmic safety parameters, while asymptotic stability and instability yield positive and negative safety parameters, respectively. It may also be noted that empirical results reported by Gipps (1981) suggest that behavioural models are non-oscillating, and hence *locally stable*, regardless of the values of β_l and β_f. Similar tests carried out by the author support this possibility.

A physical interpretation of the stability boundary case

In the case of the stability boundary, one may substitute a common deceleration parameter, β, for β_l and β_f. The constrained car following model (1.5) thus becomes:

$$v_f(t+\tau)+\delta_v'(v_f(t+\tau),\beta)a_f(t+\tau)=v_l(t)+\delta_v'(v_l(t),\beta)a_l(t) \tag{5.9}$$

Clearly, a solution to this differential equation is simply $v_f(t+\tau) = v_l(t)$, which is the steady state condition as defined in (2.1). Substituting this result, along with $\beta_l = \beta_f$, into the active safety constraint (1.4), yields;

$$x_f(t+\tau)+\lambda = x_l(t) \tag{5.10}$$

What this implies is that if the initial condition satisfies the steady state condition, the follower's trajectory will be an exact copy of the leader's trajectory, displaced τ units in time and λ units in space. Practically speaking however, one may consider the follower's trajectory to be sufficiently close to (5.10) after a relatively short time from when the safety constraint becomes active, particularly when the leader's speed is close to zero due to heavy congestion or traffic signals. To interpret the stability boundary in a physical way, one might consider the case of queue dispersion at a traffic signal. All vehicles stopped in queue at the moment the signal turns green, as well as all those that arrive to the end of the queue behind a near-stationary leader ($v_l \approx 0$), approximately satisfy the steady state condition ($v_f(t+\tau) \approx v_l(t)$). By (5.10), the last vehicle in queue would accelerate with virtually the same trajectory as the first vehicle in queue, offset in time and space. This does not seem to be very realistic: common experience would indicate that acceleration is typically less severe, the further back one is in the queue. It is not possible to say, however, whether this phenomenon is due to the existence of a "positive safety factor", or simply due to the non-homogeneity of the traffic.

6 The Linear Safety Parameter

Another behavioural hypothesis is that the follower maintains a distance behind the leader that allows him a response time that is longer than his actual response time (Gipps, 1981). Since the follower's actual response time no longer corresponds to the response time implicit in his behaviour, the follower's trajectory at time t is determined as a function of where he *might* be some short time later. For simplicity, a linear prediction will be used here. Introducing a *linear safety parameter*, θ, one assumes that:

$$v_f(t+\theta)=v_f(t), \tag{6.1}$$

$$x_f(t+\theta)=x_f(t)+\theta v_f(t). \tag{6.2}$$

The safety constraint (1.2) thus becomes:

$$(6.3)\qquad x_{\mathrm{f}}(t+\tau)+\theta v_{\mathrm{f}}(t+\tau)+\delta(v_{\mathrm{f}}(t+\tau),\beta_{\mathrm{f}})+\lambda \le x_{\mathrm{l}}(t)+\delta(v_{\mathrm{l}}(t),\beta_{\mathrm{l}})$$

This hypothesis thus yields a variant of the original safety constraint (1.2), by adding the term $\theta v_{\mathrm{f}}(t+\tau)$ to the left hand side. This variant cannot be used with the logarithmic model, since it would not be possible to solve the inequality (6.3) for $v_{\mathrm{f}}(t+\tau)$, in order to define the unconstrained car following model. One may apply this extension to the quadratic model, however, the result being essentially equivalent to Gipps' (1981) car following model. In this case, (6.3) becomes quadratic in velocity, and the approach taken by Gipps was to solve for v_{f} using the quadratic formula. When the follower's trajectory is constrained, however, one may solve the constrained car following model (analogous to (1.6)) corresponding to the quadratic version of (6.3);

$$(6.4)\qquad a_{\mathrm{f}}(t+\tau)=\beta_{\mathrm{f}}^{\mathrm{quad}}\left\{\frac{v_{\mathrm{l}}(t)}{v_{\mathrm{f}}(t+\tau)}\left(1+\frac{a_{\mathrm{l}}(t)}{\beta_{\mathrm{l}}^{\mathrm{quad}}}\right)-(1+\theta)\right\},$$

which is again a linear equation, and thus easier to solve than the quadratic formula. Furthermore, the steady state spacing in this case is given by:

$$(6.5)\qquad s_0(v,\xi^{\mathrm{quad}},\theta)=v^2\xi^{\mathrm{quad}}+(\theta+\tau)v+\lambda, \qquad \forall\, v<V.$$

It may be noted that (6.5) is of the same form as the steady state spacing (2.4) developed for the original safety constraint (1.2): the only difference between the two is in the interpretation of the linear coefficient. Whereas originally, this term was assumed to yield the response time, τ, it is now assumed to yield the response time plus a linear safety parameter. To be calibrated, such a model (6.4) requires that both the parameters β_{f} and τ be determined empirically. However, since the linear coefficients determined in the above examples are virtually equivalent to typical values of driver response time, it does not appear that the linear safety parameter can be used in *conjunction* with a quadratic safety parameter. That would imply response times significantly less than 1 second, which may be unrealistic. The linear safety parameter will thus be considered here as an *alternative* to the quadratic safety parameter, and will thus be analyzed only for the case where $\beta_{\mathrm{f}} = \beta_{\mathrm{l}}$. Defining a common deceleration parameter, β^{quad}, the unconstrained car following model, constrained car following model (6.4), and steady state spacing (6.5) become, respectively:

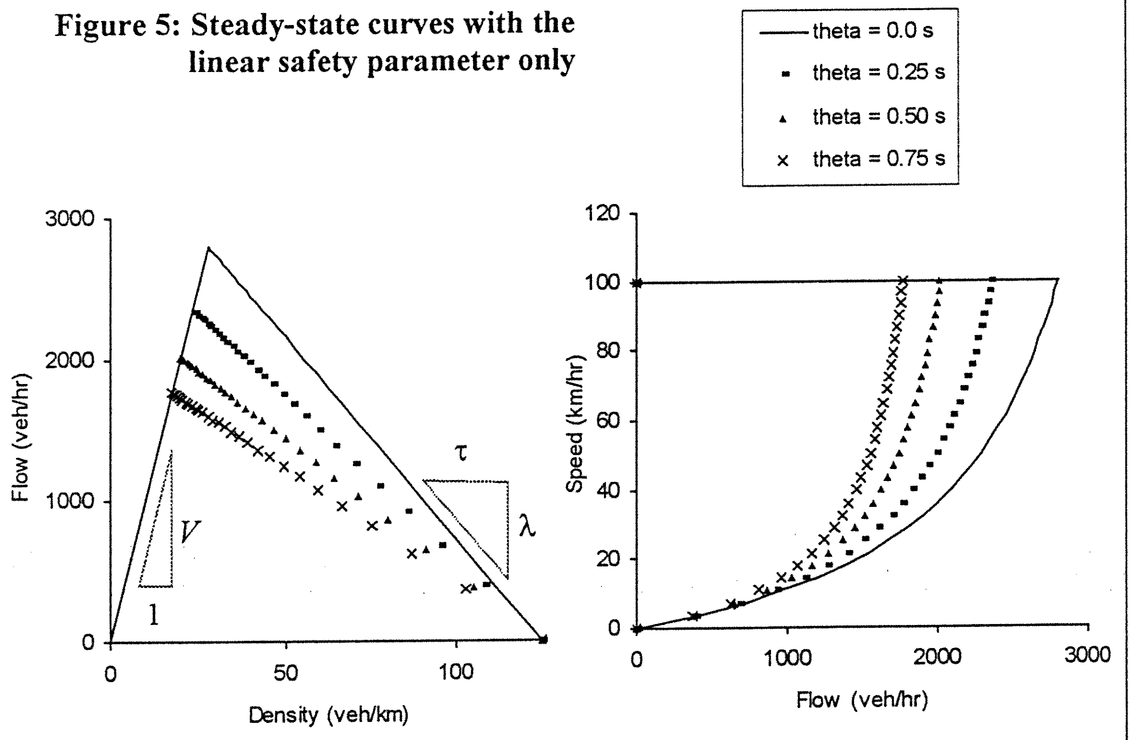
$$(6.6) \qquad v_{\mathrm{f}}(t+\tau) \le \beta^{\mathrm{quad}}\left\{-\theta+\sqrt{\theta^2-\frac{2}{\beta^{\mathrm{quad}}}\left(-x_{\mathrm{l}}(t)-\frac{v_{\mathrm{l}}(t)^2}{2\beta^{\mathrm{quad}}}+x_{\mathrm{f}}(t+\tau)+\lambda\right)}\right\}.$$

$$(6.7) \qquad a_{\mathrm{f}}(t+\tau) = \beta^{\mathrm{quad}}\left\{\frac{v_{\mathrm{l}}(t)}{v_{\mathrm{f}}(t+\tau)}\left(1+\frac{a_{\mathrm{l}}(t)}{\beta^{\mathrm{quad}}}\right)-(1+\theta)\right\}$$

$$(6.8) \qquad s_0(v,\theta) = (\theta+\tau)v+\lambda, \qquad\qquad \forall\, v < V,$$

where $\lambda = 1/K$, as before.

Figure 5: Steady-state curves with the linear safety parameter only



Since the derivative of headway is now negative for all v, the speed at capacity is always equal to the maximum speed (V). Given the value of capacity (Q), θ can be calculated as:

$$(6.9) \qquad \theta = \frac{1}{Q}-\frac{\lambda}{V}-\tau.$$

The macroscopic curve is again composed of two segments: a linear segment equivalent to the maximum speed (V) for all values of spacing satisfying the safety constraint (6.6), and the boundary based on the steady state spacing for $v < V$ (6.8). The traffic stream parameters V=100 km/hr, K=125 veh/km, and Q = 2000 veh/hr yield the model parameters $\lambda = 8.0$ m and $\theta+\tau = 1.5$ s. Assuming for example, that τ equals 1.0s, this implies a linear safety parameter of 0.5s, which seems quite plausible. Figure 5 shows the macroscopic curves for this model when the parameters V, λ, and τ are held constant at 100 km/hr, 8.0 m, and 1.0 s, respectively, while the value of θ is varied. An analysis of the stability of the constrained car following model (6.7), analogous to that carried out in section 5, yields the same relationship between stability and the linear safety parameter: asymptotic stability implies $\theta > 0$, instability implies $\theta < 0$, and the stability boundary implies $\theta = 0$.

7 Discussion and Conclusions

As can be seen from the results in Figures 2, 4, and 5, there is a clear trend toward higher capacities with lower values of the safety parameters. This implies that for given values of maximum speed, effective vehicle length, response time, and desired deceleration (β_f), higher capacity implies both a lower level of safety and lower stability in the traffic stream. It may also be observed that for the quadratic and logarithmic models, (Figures 2 and 4) the speed at capacity decreases as the safety parameter increases. If asymptotic stability is considered to be a necessary property of the traffic stream, the stability boundary represents a theoretical limit on the macroscopic curve, as a function of the maximum speed, effective vehicle length, and response time. This envelope takes on a triangular shape in the space of flow vs. density (see Figures 2, 4, and 5). The left side of the envelope represents uncongested conditions, and is defined by the slope V from the origin to a maximum capacity value, Q_{max}. The right side of the envelope represents congested conditions, and is defined by the slope $-\lambda/\tau$, from the point (0,K) to the maximum capacity, Q_{max}. Since real traffic is not necessarily stable, nor homogeneous, this envelope may in fact be exceeded by empirical data, particularly when they are measured over relatively short intervals. It would be interesting to investigate, however, whether empirical data respect this boundary when the appropriate values of V, λ, and τ are known. The capacity corresponding to the stability boundary is given by $Q_{max} = V/(\tau V+\lambda)$. The parameters V= 100 km/hr, λ = 8.0 m, and τ = 1.0 s, yield Q_{max} = 2795 veh/hr. This is considerably higher than empirically observed capacities, and suggests that the safety parameters will take on positive values when these models are calibrated to empirical data.

An important property of the constrained car following models (equations 3.3, 4.6, and 6.7) is that they are *linear models*, and thus very inexpensive from a computational point of view. Of course, one can only use these models when the safety constraint is active: when inactive, it must be handled using the appropriate unconstrained, and non-linear, model. However, as a network becomes congested, the fraction of linear calculations should steadily increase, and thus the efficiency of the model will increase. Given the size of real networks, and the interest in real-time applications of micro-simulation, this increase in efficiency as congestion increases should prove to be an important advantage of these models. Another important property of behavioural models is that most of the parameters correspond to measurable quantities, and all have a physical interpretation. This allows one to form an opinion about the plausibility of the model, by constructing theoretical (macroscopic) curves based only on the knowledge of these parameters, and comparing them with empirical curves. Conversely, one may compute values of the behavioural parameters from empirically obtained macroscopic parameters, and compare the former to corresponding empirical measurements. Using typical macroscopic parameters, the latter approach was used to obtain very realistic values of response time, suggesting that these behavioural models may be quite plausible, indeed.

By comparing the shapes of the macroscopic curves (Figures 2, 4, and 5) to empirical data, one may consider the plausibility of the different behavioural assumptions considered. On the basis of purely qualitative comparisons with standard curves in the literature (May, 1990), the logarithmic model appears to be the most plausible of the three. Although the decrease in speed, as uncongested flow approaches capacity, may in reality be much less pronounced than commonly believed (Cassidy, 1998), the logarithmic model is the only one of the three that predicts a gradual decrease in speed in this region of the curve. Unlike the quadratic model (with linear safety parameter), which has not only been implemented by Gipps (1981), but also in a very modern micro-simulation package (Barceló *et al*, 1994), the logarithmic model has yet to be implemented in a computer program. As an avenue of further research, implementation and testing of this model in a micro-simulation environment may be very promising.

References

Barceló, J, Ferrer, J.L, and R. Grau (1994) AIMSUN2 and the GETRAM Simulation Environment. *Internal Report*, Departamento de Estadística e Investigación Operativa. Universitat Politècnica de Catalunya.

Cassidy (1998) Bivariate relations in nearly stationary highway traffic. *Transportation Research B* 32B (1), pp. 49-59.

Gazis, D.C, Herman R, and R.B. Potts (1959) Car following theory of steady state flow. Operations Research 7 (4), pp. 499-505.

Gazis, D. C, Herman, R, and R.B. Potts (1961) Nonlinear follow-the-leader models of traffic flow. *Operations Research* 9 (4), pp. 545-567.

Gerlough D. L, and Huber, M. J. (1975) *Traffic Flow Theory*, Transportation Research Board Special Report 165, Washington D.C.

Gipps, P.G. (1981) A behavioural car following model for computer simulation. *Transportation Research 15B*, pp. 105-111.

Herman, R, Montroll, E. W, Potts, R.B, and Rothery, R.W. (1959) Traffic dynamics: analysis of stability in car-following. *Operations Research* 7 (1), pp. 86-106.

Herman, R. (1991) Traffic dynamics through human interaction. *Journal of Economic Behaviour and Organization 15* (2), pp. 303-311.

Krauss, S. (1998) The role of acceleration and deceleration in microscopic traffic flow models. Preprint No. 980478, Transportation Research Board, 77th Annual Meeting, January 11-15, 1998, Washington D.C.

Leutzbach, W. (1988) *Introduction to the Theory of Traffic Flow*, Springer-Verlag Berlin, Heidelberg.

May, A. D. (1990) *Traffic Flow Fundamentals*. Prentice Hall, Englewood Cliffs.

From Traffic Flow To Queueing Theory

Michael Mahut, University of Montreal, Centre for Research on Transporation (C.R.T.)
Montreal, Canada

Abstract

Simplified theories – or models – of traffic flow have received considerable attention over the last decade. One motivation for the development of such models is the need for faster-than-real-time traffic models that can be used for dynamic traffic assignment on large networks for Intelligent Transportation Systems (ITS) applications. A simple traffic flow model is proposed here which is defined on a discrete notion of flow and continuous space and time. It is found that the model can be solved by an algorithm that only evaluates the flow at the entrance and exit points of an arc. It is also found that the model yields the same waiting time as a single-server first-in-first-out queue, and that the spatial extension of delay propagation along an arc is governed by the same rules that determine queue length in a point queue model. The mathematical structure of the model is thus very similar to that of a point queue, but the model is also shown to yield a two-linear-segment relationship between flow and density.

Acknowledgements

This work was supported in part by a post-graduate scholarship of the Natural Sciences and Engineering Research Council of Canada (NSERC). The author would like to thank Professor Michael Florian of the University of Montreal, Canada, and Vittorio Astarita of the Universita degli Studi della Calabria, Italy, for many valuable comments and discussions during the preparation of this work.

1 Introduction

Over the last decade, several simplified models of vehicular traffic have been proposed in the literature. These have included discrete-flow models (Nagel and Schreckenberg, 1992), (Schreckenberg *et al*, 1995), and continuous-flow models (Newell, 1993), (Daganzo, 1994). Although discrete and continuous-flow are fundamentally different approaches, the macroscopic relationships between flow and density associated with these models are very similar. It should be noted that in the case of discrete-flow models, this relationship is a result of microscopic interactions between the vehicles – and hence an output – while in the case of continuous flow, this is an exogenous relationship – and hence an input. The discrete-flow models mentioned above, which are cellular automaton (CA) traffic models, yield a two-linear-segment relationship, which is also the general shape of the relationship underlying the model proposed by Newell. The model proposed by Daganzo – known as the *cell transmission* model – includes a third linear segment defining a lower maximum flow than that obtained using the two segments alone.

From the point of view of computational effort, all of these approaches are similar. The discretization schemes may be different, but the discrete quantities of space, time, and flow that these models work with are generally of the same order of magnitude. These quantities are usually smaller in CA – and hence there are more calculations to execute – but these models use simpler mathematical operations. The general point of similarity between these different models is that the coarser the discretization the lower the fidelity of the results, and hence comparable results require comparable computational effort. An exception to this rule is a method discussed by Newell for solving the model he proposed, which can be applied over an entire arc at a time. Although Newell does not provide an automated algorithm for this method – it is carried out "by hand" – it is nevertheless significant that the fidelity of the results does not degrade with increasing arc length. This solution is thus of a different class than the other, automated algorithms. Moreover, it demonstrates that with respect to the number of calculations, the other algorithms are not of the lowest computational order needed to move flow along an arc in a way that is consistent with a two-linear-segment flow-density relationship.

In what follows, a simplified discrete-flow traffic model is presented, which can be solved by an automated algorithm on an arc basis, i.e., without evaluating the flow at intermediate points along an arc. The structure of the model is found to be very similar to a capacitated point queue, where the arc length defines the maximum number of vehicles that may be in queue. For an arbitrarily long arc, vehicle delay is found to respect Lindley's recursion for a single-server (GI/G/1) FIFO (first-in-first-out) point queue, and the spatial extension of delay propagation is found to behave as the queue length in a point queue. An important difference between this model and a point queue is that delayed vehicles do not stack up "vertically" at the server, but along a diagonal line in space time defined by the (constant) negative wave speed associated with the model. As a result, the notion of queue length does not refer to the number of vehicles in queue at any moment in time, but rather to the spatial extension of the queue. This is the distance between the last vehicle on the arc to have incurred any delay, and the exit of the arc. This model is furthermore found to yield the two-linear-segment relationship between flow and density when the average speed of vehicles is the same over some space-time domain. For these reasons this model is referred to here as a *space-time queue* (STQ). Numerical results are presented for an example with time-varying demand and supply.

2 Model Statement

To begin with, some traffic flow parameters will be defined. All vehicles are assumed to be identical, and are described by two parameters. The effective vehicle length, denoted by λ, is equal to the inverse of the maximum – or *jam* – density, denoted by K. This parameter is thus equal to the average vehicle length plus the average distance between vehicles when stopped in queue. The response time, denoted by τ, is the average time required for the driver/vehicle to observe, decide, react, and ultimately affect its space-time trajectory. The model developed here is concerned only with longitudinal dynamics, and is thus restricted to the context of a single arc in a network. The arc is defined by two parameters: V denotes the maximum –or *desired* – speed of the vehicles, and X denotes the arc length. The primary variables are space (x), time (t), and cumulative vehicle count (n). The process of traffic flow may thus be described by any one of the following three relationships: $x(t,n)$, $t(n,x)$, or $n(x,t)$. Space and time are continuous, while vehicle count is discrete. Space and time are scaled to λ and τ, i.e., the units of space, time and speed are λ, τ, and λ/τ, respectively. For clarity, these units will often be included in the formulas that follow.

Cellular Automaton (CA) traffic models, as well as some car-following models, move vehicles by maximizing their speeds in the presence of certain constraints. Such constraints ensure that the vehicle trajectories respect certain bounds on position, speed, and acceleration: in particular, they ensure that accidents do not occur. These constraints are used to determine, at each discrete point in time, what the vehicle's speed will be over the next time step. In such models – indeed, in all car-following models – congestion is caused by vehicle deceleration due to these constraints. The constraints used to define the STQ are a simplified form of the car-following model proposed by Mahut (1999a/b), which is based on the well-known idea of safe-stopping distances. Removal of the acceleration constraint from this model leaves only the upper and lower bounds on speed to constrain the movement of a vehicle with no leader. Removal of the deceleration constraint from this model leaves only the following constraint to ensure that accidents do not occur:

$$x(t,n) \leq x(t-\tau,n-1)-\lambda \ . \tag{2.1}$$

Without a constraint on deceleration, the stopping distance is zero. As a result, the speed at which a vehicle may be travelling is not constrained by the presence of obstacles (vehicles or traffic signals) downstream – until the presence of the obstacle requires the vehicle to actually stop. This in turn implies that the constraints on a vehicle's trajectory need only be concerned with when a vehicle passes some position, but not how fast (c.f. the description of car-following models above). More precisely, since vehicles may have zero speed, the STQ constraints are concerned with when a vehicle may *arrive to* and *depart from* some position x, which motivates the following notation:

$$(2.2)\qquad \begin{cases} t^A(n,x) & = \textit{the time of arrival of vehicle n to position x} \\ t^D(n,x) & = \textit{the time of departure of vehicle n from postion x} \end{cases}$$

By convention, the arrival and departure times refer to the position of the front of the vehicle. It can be seen then that each vehicle's space-time trajectory will be composed of straight line segments of only two different slopes: one representing the maximum speed (V), the other representing zero speed. This property is also characteristic of a (deterministic) CA model often used for modelling traffic, known as CA-184 (Nagel, 1996). Using the above notation, the constraints and objective function defining the STQ queue will now be presented.

There are two boundaries to the arc: the entrance $(x=0)$, and the exit $(x=X)$. A demand process is defined at the entrance, and a delay process is defined at the exit. These two processes are analogous to the arrival and service processes, respectively, of a point queue. The demand process yields the earliest possible time that a vehicle may depart from the position $x=0$, due to its trajectory upstream of this point – or, for that matter, due to some model used to generate vehicles at an origin in a network. This time, $t^{dm}(n)$, called the *demand time* of vehicle n, is defined in terms of a demand sequence, $\alpha(n)$, as follows:

$$(2.3)\qquad t^{dm}(n) = t^{dm}(n-1) + \left(\tau + \frac{\lambda}{V}\right) + \alpha(n)$$

By convention, $t^{dm}(0)$ is equal to $(-\tau - \lambda/V)$. The quantity $t^{dm}(n) - t^{dm}(n-1)$ is referred to as the *inter-demand* time of vehicle n: $\alpha(n)$ is the variable part of this quantity. The role of the term $(\tau + \lambda/V)$ will be discussed later on. The demand time defines a lower bound on the departure time from $x=0$ as follows:

(2.4) $$t^{D}(n,0) \geq t^{dm}(n) = t^{dm}(n-1) + \left(\tau + \frac{\lambda}{V}\right) + \alpha(n) \qquad \forall n.$$

A delay time, $\omega(n)$, is assigned to each vehicle at $(x = X)$. The delay time defines a lower bound on the departure times from the exit of the arc as follows:

(2.5) $$t^{D}(n,X) \geq t^{A}(n,X) + \omega(n) \qquad \forall n.$$

The demand and delay sequences may follow any arbitrary distributions. These two sequences, along with the values of λ, τ, V and X constitute the inputs to the model.

An important property of this model is that vehicle delays will always be incurred at integer distances from the exit of the arc. This is because the only possible delays – or perturbations – to the traffic stream must originate at the exit of the arc, due to the delay sequence $\omega(n)$: there are no sources of delay within the arc itself. Since vehicles will always decelerate instantaneously when delayed, any and all delay experienced by vehicle n due to $\omega(n)$ will be incurred at position $x = X$. If vehicle $n+1$ should furthermore be delayed by vehicle n, any and all such delay will be incurred at position $x = X - \lambda$. This argument extends to all vehicles so delayed, and thus all delays are incurred at integer distances from the exit. If one furthermore constrains X to be a whole multiple of λ, this property can be extended to a network of arcs: if the last vehicle on an arc is stopped at position $x = \lambda$, the first vehicle on an upstream arc will stop at position $x = X$. Under this last constraint, not only will all vehicles stop at integer distances from the exit, but also at integer distances from the entrance. This in turn allows the constraints on the vehicle trajectories to be defined only on integer values of x, even though the model itself is defined on continuous space. For the remainder of this discussion it will be assumed that x is a non-negative integer.

A parallel consequence of this last constraint is that the distances traveled during the moving segments of all vehicles' trajectories are always whole multiples of λ. This in turn guarantees the following relationship at all times:

(2.6) $$t^{A}(n,x) = t^{D}(n,x-\lambda) + \frac{\lambda}{V} \qquad \forall n, \forall x \geq \lambda.$$

This last result allows the model to be written in terms of either $t^{A}(n,x)$ or $t^{D}(n,x)$. The remaining constraints on the vehicle trajectories may now be developed. There are two principal types of constraints on a vehicle's movement: *demand* constraints, and *supply* constraints. The *demand* constraints ensure that a vehicle's speed lies between zero and the maximum speed, V. As such, they define the earliest possible times that a vehicle may arrive and depart from some position x as a function of its trajectory upstream of this position. These constraints are stated as follows:

$$t^A(n,x) \geq t^D(n,x-\lambda)+\frac{\lambda}{V} \qquad \forall n, \forall x > 0, \tag{2.7}$$

$$t^D(n,x) \geq t^A(n,x) \qquad \forall n, \forall x > 0\,. \tag{2.8}$$

Analogous constraints in a discrete-time model might be stated as follows:

$$x(t,n) \leq x(t-\tau,n)+V\tau\,, \qquad \forall n, \forall t > 0, \tag{2.9}$$

$$x(t,n) \geq x(t-\tau,n) \qquad \forall n, \forall t > 0, \tag{2.10}$$

where t is a whole multiple τ. The *supply* constraints provide the earliest times that a vehicle may arrive and depart from some position x as a function of the trajectory of the next downstream vehicle. These constraints are stated as:

$$t^A(n,x) \geq t^A(n-1,x+\lambda)+\tau \qquad \forall n > 1, \forall x < X \tag{2.11}$$

$$t^D(n,x) \geq t^D(n-1,x+\lambda)+\tau \qquad \forall n > 1, \forall x < X \tag{2.12}$$

These constraints are derived by writing (2.1) in terms of $t(n,x)$, rather than $x(t,n)$. The objective of speed maximization is represented here by minimizing the departure time for each vehicle from each position x:

$$MIN[t^D(n,x)] \tag{2.13}$$

This completes the definition of the model. As an aside, it may be noted that combining (2.6), (2.8) and (2.12) yields:

$$t^D(n,x)-t^D(n-1,x) \geq \left(\frac{\lambda}{V}+\tau\right) \tag{2.14}$$

This result defines the minimum time separation, or headway, between successive vehicles. Since it is not possible that the temporal separation between two vehicles be less than $\tau + \lambda/V$, this constant appears in the definition of inter-demand times given by (2.3). Moreover, since (2.14) is also valid if $t^D(n,x)$ is replaced by $t^A(n,x)$, the minimum time between successive departures from the arc is given by:

$$(2.15) \qquad t^D(n,X) - t^D(n-1,X) \geq \left(\tau + \frac{\lambda}{V}\right) + \omega(n-1)$$

The quantity on the right-hand side of (2.15) is referred to as the service time of vehicle *n*-1. It represents the duration between the departure times of vehicles *n* and *n*-1 when the supply constraints are active. Another property of some interest is the minimum distance separation, or spacing, between successive vehicles at a given moment in time. This spacing can only be achieved if the leading vehicle moves at speed *V* for at least one τ, in which case (2.9) and (2.1) can be combined to yield:

$$(2.16) \qquad x(t,n-1) - x(t,n) \geq (\lambda + V\tau)$$

3 The Cell-Based Solution

The solution to this model is trivial: it consists of taking the maximum over all of the constraints for $t^A(n,x)$ and $t^D(n,x)$ as defined above for the different regions of the (n,x) domain, and making the appropriate substitutions using (2.6). This solution is given by the following relationships:

$$(3.1) \quad t^D(n,x) = t^{dm}(n,x) \qquad n = 1, x = 0,$$

$$(3.2) \quad t^D(n,x) = t^D(n,x-\lambda) + \frac{\lambda}{V} \qquad n = 1, x > 0,$$

$$(3.3) \quad t^D(n,0) = MAX\left[t^{dm}(n), t^D(n-1,\lambda) + \tau\right] \qquad n > 1,$$

$$(3.4) \quad t^D(n,x) = MAX\left[t^D(n,x-\lambda) + \frac{\lambda}{V}, t^D(n-1,x+\lambda) + \tau\right] \qquad n > 1, 0 < x < X,$$

$$(3.5)\quad t^D(n,X) = MAX\left[t^D(n,X-\lambda)+\frac{\lambda}{V}, t^D(n-1,X)+\tau+\frac{\lambda}{V}\right]+\omega(n) \quad n>1.$$

Of course, $t^D(n,x)$ must be calculated in a sequential way, due to the dependency of $t^D(n,x)$ on $t^D(n,x-\lambda)$ and $t^D(n-1,x+\lambda)$. The fact that these dependencies always move forward in time allows the model to be solved by discrete-event ("event-based") simulation. This property is also a basic requirement of any sensible model of physical interactions. Due to the discrete nature of both space and flow, the calculations may also be carried out using a spreadsheet. This solution to the model will be referred to as the *cell-based space-time queue* (CSTQ), since it is based upon a discretization of space into cells of length λ, much like CA traffic models.

A Numerical Example

A simple example of the CSTQ was solved using a spreadsheet. The arc length was 20λ, and the maximum speed was λ/τ. For simplicity, $\alpha(n)$ was set to zero for all n. This represents a traffic stream moving at its maximum speed, and at the maximum flow rate permitted at this speed. This special case demonstrates some important properties of the model. The delay sequence, $\omega(n)$, was drawn from a negative exponential distribution with a mean of τ.

Figure 1 shows a sample of the output from the spreadsheet. As discussed earlier, the trajectories are composed of straight-line segments, with only two different slopes. On the right-hand side of the plot, the vertical line segments at $x = 20$ indicate the delays incurred due to $\omega(n)$. This plot clearly shows the propagation of these delays upstream through the platoon of vehicles. The speed with which these delays propagate is $-\lambda/\tau$, as indicated by a dashed line drawn through $(20,20)$ with this slope. This property suggests that the model has a constant negative wave speed, and thus a two-linear-segment flow-density relationship. Such a relationship is indeed derived for this model in a later section.

It should also be noted that, given the demand process used in the example, the delays remain of exactly the same magnitude as they move upstream. Since the demand flow represents a traffic stream in "steady-state" (i.e., moving at constant speed, flow, and density), and at the minimum spacing at this speed, this property indicates that the model is on the boundary between asymptotic stability and instability. Asymptotic stability of a car-following model is concerned with whether a perturbation to the first vehicle of a platoon moving at constant speed and maximum density, increases or decreases in magnitude as it moves upstream from one vehicle to the next. Since perturbations to the traffic in this model are measured in terms of delay, and these delays remain of exactly the same magnitude as they move upstream, this model can be said to be boundary-asymptotically stable. It may be noted that this property is also true of the original car-following model from which the supply constraints were derived, when the deceleration parameter for the leading and following vehicles is taken to be the same (Mahut, 1999a/b).

4 The Arc-Based Solution

In most applications of traffic models, statistics are calculated solely on the basis of $t^D(n,0)$ and $t^D(n,X)$, which define the movement of vehicles from one arc to another. For practical purposes, it is thus sufficient to know only these processes. With this in mind, a more compact solution to the STQ can be derived as follows. Substituting (3.4) for the first term inside the *MAX* operator in (3.5) yields:

$$t^A(n,X) = MAX\left[t^A(n,X-2\lambda)+\frac{2\lambda}{V}, t^D(n-1,X)+\tau+\frac{\lambda}{V}\right]+\omega(n) \tag{4.1}$$

Substituting (3.4) once again for the first term inside the *MAX* operator in (4.1), and repeating this operation $(X-1)$ times, yields:

$$t^D(n,X) = MAX\left[t^D(n,X-\lambda)+\frac{\lambda}{V}, \underset{i\in\{0,\ldots,X-2\}}{MAX}\left[t^D(n-1,X-i\lambda)+\tau+\frac{(1+i)\lambda}{V}\right]\right]+\omega(n) \tag{4.2}$$

Combining the demand constraints, (2.7) and (2.8), yields:

$$t^D(n,x) \geq t^D(n,x-\lambda)+\frac{\lambda}{V}, \tag{4.3}$$

which may be applied to the terms inside the second *MAX* operator in (4.2), to yield:

$$t^D(n,X) = MAX\left[t^D(n,0)+\frac{X\lambda}{V}, t^D(n-1,X)+\left(\tau+\frac{\lambda}{V}\right)\right]+\omega(n) \qquad n>1 \tag{4.4}$$

For $n=1$, the departure time from the exit of the arc is simply:

$$t^D(n,X) = t^D(n,0)+\frac{X}{V} \qquad n=1 \tag{4.5}$$

A relationship for $t^D(n,0)$ may be developed in a similar way. Substituting (3.4) for the second term inside the *MAX* operator in (3.3) yields:

$$(4.6) \qquad t^D(n,0) = MAX\left[t^{dm}(n), t^D(n-1,\lambda) + \tau + \frac{\lambda}{V}, t^D(n-2,2\lambda) + 2\tau\right]$$

Substituting (3.4) once again for the last term inside the *MAX* operator in (4.6), and repeating this operation $(X-1)$ times, yields, for $n > X$:

$$(4.7) \qquad t^D(n,0) = MAX\left[t^{dm}(n), \underset{i\in\{1,\dots,X-1\}}{MAX}\left[t^D(n-i,(i-1)\lambda) + i\tau + \frac{\lambda}{V}\right], t^D(n-X, X\lambda) + X\tau\right].$$

The supply constraint (2.12) may now be applied to the terms inside the second *MAX* operator in (4.7), which then simplifies to:

$$(4.8) \qquad t^D(n,0) = MAX\left[t^{dm}(n), t^D(n-1,0) + \tau + \frac{\lambda}{V}, t^D(n-X, X\lambda) + X\tau\right] \qquad n > X.$$

For $n \le X$, it is not possible that $t^D(n,0)$ be affected by the delays experienced at the exit by any preceding vehicles, and hence:

$$(4.9) \qquad t^D(n,0) = t^{dm}(n,0) \qquad n \le X.$$

The equations (4.4), (4.5), (4.8) and (4.9) will be referred to as the *arc-based space-time queue* (ASTQ). This solution is based solely upon the departure times of each vehicle at the entrance and exit of the arc. Clearly, this solution has an important advantage over the CSTQ in terms of computational effort. It is readily apparent that the computational cost is constant in the arc length, and thus linear only in the number of vehicles to pass through the arc. The computational cost is thus comparable to that of a point queue model. The data requirements are quite modest as well: each arc must store the last departure time at the entrance, and the last *X* departure times at the exit.

5 The Equivalent Point Queue

In this section, the relationship between the STQ and the well-known point queue (PQ) model is considered. This relationship will be developed under the assumption of an arbitrarily long arc, on which the exit delays can never propagate back to the arc entrance. Under this assumption, it can be seen that:

$$(5.1)\qquad t^D(n,0)=t^A(n,0)=t^{dm}(n,0)=t^D(n-1,0)+\tau+\frac{\lambda}{V}+\alpha(n)\qquad \forall n.$$

The total delay time of vehicle n, denoted by W_n, is now defined as:

$$(5.2)\qquad W_n=t^D(n,X)-\omega(n)-\frac{X\lambda}{V}-t^D(n,0).$$

This is simply the total travel time along the arc as defined by the departure times at the entrance and exit, minus the exit delay and the minimum travel time. Since the exit delay is not included, this is analogous to the waiting time in queue – rather than in the system – in a point queue. Substituting (4.4) into (5.2) yields:

$$(5.3)\qquad W_n=MAX\left[0,t^D(n-1,X)-t^D(n,0)+\tau+\frac{\lambda}{V}-\frac{X\lambda}{V}\right].$$

Substituting (5.1) into (5.3) and reducing, yields:

$$(5.4)\qquad W_n=MAX[0,W_{n-1}+\omega(n-1)-\alpha(n)].$$

This last equation is Lindley's waiting time recursion for a first-in-first-out (FIFO) single-server (GI/G/1) point queue (Lindley, 1952), which demonstrates that the waiting time in the STQ (for an arbitrarily long arc) is exactly as would be determined by a point queue with arrival and service processes defined by $\alpha(n)$ and $\omega(n)$, respectively. It should be noted that $\alpha(n)$ and $\omega(n)$ do not represent the inter-arrival and service times, but only the variable portions of these quantities. The inter-arrival and service times are given by $\alpha(n)+\tau+\lambda/V$ and $\omega(n)+\tau+\lambda/V$, respectively – see (2.15) and (5.1), above. Of course, had it been assumed from the beginning that the traffic could be modeled as a point queue, (5.4) could have been obtained directly from the inter-arrival and service time distributions, as:

$$(5.5)\qquad W_n=MAX\left[0,W_{n-1}+\left(\tau+\frac{\lambda}{V}+\omega(n-1)\right)-\left(\tau+\frac{\lambda}{V}+\alpha(n)\right)\right],$$

and thus:

$$(5.6)\qquad W_n=MAX[0,W_{n-1}+\omega(n-1)-\alpha(n)].$$

This is precisely Lindley's observation that "the waiting-time distribution depends only on the distribution of the *difference* between the service time and [inter-arrival] time and not on their individual distributions" (Lindley, 1952). What may appear to be counter-intuitive here is that the variability of the inter-arrival and service time distributions should be reduced in this way, leading to shorter delays and queue lengths than would otherwise be calculated.

The derivation of (5.4) of course suggests that not only is the waiting time in the STQ equivalent to that obtained in a point queue, but that there may be an equivalent notion of queue length as well. Intuitively, an important difference between the two models is that in the STQ, sequential vehicles experiencing delay due to a specific exit delay do not experience this delay at the same time. As a result, any notion of a queue in this model that grows and decays like a point queue defined by $\alpha(n)$ and $\omega(n)$ cannot refer to the state of the vehicles as seen at one instant in time. This makes the comparison with point queues somewhat difficult, as the usual approach to defining the relationship between waiting time and queue length depends on being able to define the state of the system at each point in time – for example, see Little (1961). The second basic difference between the two models is that in the STQ, sequential vehicles experiencing delay due to a specific exit delay do not experience this delay at the same position.

These differences can be overcome by the development of an *equivalent point queue*, which is done by graphical means as follows. For simplicity, it will be assumed that the demand process (2.3) now represents the demand time for exiting the arc, i.e., $x = 0$ is now taken to be the arc exit. Since arc length is not considered in this development, the vehicles will be allowed to accumulate indefinitely along the negative *x*-axis. A third axis is added to the plot, which starts at the point $(0,0)$ with slope $-\lambda/\tau$. This axis lies in the *x*-*t* plane, and will be called the *z*-axis. One unit on this axis is equivalent to a space-time displacement of $(-\lambda,\tau)$. It may be noted that the positions and times at which any delays experienced by vehicles in the STQ must lie above the *z*-axis (see Figure 1). As a result, the demand process $t^{dm}(n)$ may be translated to the *z*-axis as an *arrival process,* $z^A(n)$. The next step is to translate the delays as defined on the *t*-axis to the *z*-axis, in such a way that the departure times from the arc, $t^D(n,X)$, remain unchanged. The departure times thus translated to the *z*-axis define a departure process, $z^D(n)$. It is very easy to show that quantities measured on the *t*-axis can be translated to the *z*-axis in this way by division by $(\tau+\lambda/V)$. To simplify the formulas that follow, $(\tau+\lambda/V)$ will hereafter be denoted by C. These transformations were carried out on the trajectories shown in Figure 1, and the resulting "trajectories" are shown in Figure 2. Where these trajectories cross the horizontal and vertical axes remains unchanged relative to the bottom right corner of the graph, which is now labeled as $(0,0)$, rather than $(20,20)$. The arrival and departure processes from the *z*-axis are defined as:

$$(5.7) \qquad z^A(n) = \frac{t^{dm}(n)}{C} = \frac{t^{dm}(n-1)+\alpha(n)+C}{C} = z^A(n-1)+1+\frac{\alpha(n)}{C}$$

$$(5.8) \qquad z^D(n) = MAX\left[z^A(n), z^D(n-1)+1\right]+\frac{\omega(n)}{C}$$

Together, these two processes define what will be referred to as the *point space-time queue* (PSTQ), since they represent a point process along a line – namely, the *z*-axis. This line is neither space, nor time, but a one-dimensional space-time. What is normally referred to as the state of the (point queue) system "at some point in time, *t*", is now the state of the (STQ) system at some point in *space-time*, *z*. This state is the queue length, and is calculated as the number of arrivals to the *z*-axis minus the number of departures from the *z*-axis that have occurred before reaching some given value of *z*.

The separation between arrival and departure along the *z*-axis now represents a modified notion of waiting time, which will be denoted by W_n'. Following (5.5) and (5.6), this quantity is given by:

$$(5.9) \qquad W_n' = MAX\left[0, W_{n-1}' + \left(1+\frac{\omega(n-1)}{C}\right) - \left(1+\frac{\alpha(n)}{C}\right)\right] = MAX\left[0, W_{n-1}' + \frac{\omega(n-1)-\alpha(n)}{C}\right]$$

Multiplying (5.9) by C recovers the original waiting time relationship (5.4) – note that by the translation rule given above, $W_n = W_n'C$. The queue length as defined for the PSTQ can also be translated onto the *x*-axis, but the notion of discrete vehicles occupying discrete cells on the arc obviates the need for a multiplication factor. Since the exogenous delays in the STQ always occur at the arc exit, the queue length as given by the PSTQ simply yields the distance from the back of the queue to the exit of the arc. This does not, however, imply that there are this many vehicles in queue *at some instant in time*: what the queue length in these models provides is the physical distance between the end of the arc and the last vehicle on the arc to have incurred any delay.

Since both the demand times and departure times at the arc exit remain unchanged, the PSTQ is equivalent to the STQ for an arbitrarily long arc. At the same time, the PSTQ is a single-server FIFO point queue; and thus, steady-state point queue formulas must be applicable to the STQ when the effect of arc length can be ignored. Of course, it is also to be expected that a capacitated PSTQ, equivalent to the STQ with a fixed arc length, can be further defined by merely adding the appropriate constraint on the queue length.

6 The Fundamental Diagram

An underlying relationship between flow and density will now be derived for the STQ. For the purpose of discussing such a relationship, it is important that these two quantities be defined upon the same domain. It was thus decided to use the definitions of flow and density (or concentration) as defined by Edie, based on a sequence of vehicle trajectories defined upon an area of space-time (Edie, 1974). The quantities of interest are the area of the space-time domain in question (XT), the total distance traveled by all vehicles over this domain (NX), and the total time spent by all vehicles over this domain (NT). Flow (*q*) and density (*k*) are then defined as:

$$(6.1) \qquad q = \frac{NX}{XT}, \qquad\qquad k = \frac{NT}{XT}.$$

The space-time domain may be of any general shape. These definitions will now be applied to an arbitrary number (*m*) of trajectories defined over a space-time domain, with one restriction: all trajectories must be either *constrained*, or *unconstrained*. The trajectory of vehicle *n* is *constrained* if the supply constraints (2.11-12) are active – otherwise the trajectory of vehicle *n* is said to be *unconstrained*. It will also be required that the average speed of the vehicles, defined as $v = NX/NT$, be equal to the average speed of *each* vehicle over the domain in question. In the case of constrained trajectories, this last condition is enforced by defining the domain over a sequence of trajectories that each spans a distance *X*, and a duration *T*. Rather than being rectangular, the domain is defined using lines of slope $-\lambda/\tau$ to demarcate the trajectory segments of interest, as shown in Figure 3. The total time and total distance traveled by the *m* vehicles over this domain are simply *mT* and *mX*, respectively.

The area of this space-time domain can be broken down into two exact parallelograms, as shown in Figure 4: one defined by trajectory segments at speed *V*, the other by those at zero speed. The area of the former is given by $mX(\tau + \lambda/V)$, while the area of the latter is $m\lambda(T - X/V)$. Substituting the appropriate quantities into Edie's definitions for flow and density yields:

$$(6.2) \qquad q = \frac{X}{\tau X + \lambda T}, \qquad k = \frac{T}{\tau X + \lambda T}, \qquad T \geq X/V.$$

This may be re-written as:

(6.3) $$q(k) = \frac{1-\lambda k}{\tau}, \qquad k \geq k^*$$

where k^* is the inverse of the minimum spacing between two trajectories at speed V, as given by (2.16). When all of the trajectories are unconstrained over some space-time domain, it is clear that the average speed of the traffic is V, and thus the relationship between q and k is simply:

(6.4) $$q = kV \qquad k \leq k^*$$

The relationships (6.3) and (6.4) are exactly the two-linear-segment flow-density relationship, as derived for CA traffic models (Nagel, 1996), when τ is equal to the CA time step. The relationship proposed by Newell is also of this general form (Newell, 1993). It should be noted that (6.3) and (6.4) were derived on the assumption that the average speed of each vehicle is the same, rather than on the stronger assumption that each vehicle's speed is *constant*, as is done in the analysis of car-following models (Gerlough and Huber, 1975). Constant speed is a limiting case of the conditions used here, as the number of delays goes to infinity, and the average delay duration goes to zero.

Numerical Results

In this section, the inputs and outputs of one simulation run of the arc-based model will be discussed. Demand $\alpha(n)$ and delay $\omega(n)$ sequences were drawn from negative exponential distributions with means denoted by $\alpha(t)$ and $\omega(t)$, respectively. In order to obtain a wide range of density values, the means of the demand and delay distributions were varied over time. At the entrance, the demand flow, $(\alpha(t)+C)^{-1}$, started off very low and gradually increased to the maximum flow permitted by the arc. At the exit, the mean supply rate, $(\omega(t)+C)^{-1}$, started at 1800 veh/hr and then decreased gradually to almost zero. Figure 5 shows a plot of flow vs. density for one run of the model. Superimposed on this plot is the two-segment linear relationship between flow and density derived above. The flows and densities were calculated over a rectangular domain, defined by the arc length and a 60s duration (60τ), using Edie's definitions as described above. The result is a scatter of points that follows the two parts of the linear-segment relationship, but at some point traces a path between them. The curvilinear "shape" of this scatter is quite realistic in comparison with empirical data described in the literature – see, for example May (1990). It should be noted that the discrepancy between the derived two-linear-segment relationship and the numerical simulation results is due to the use of such a large space-time domain. In general, the closer the actual conditions in any space-time window correspond to the assumptions in the above derivation (i.e., that all vehicles be either in a constrained or unconstrained state), the closer the numerical results should be to the derived relationship. This idea is very similar to that presented by Cassidy on the effect of data aggregation on the observed relationship between flow and density (Cassidy, 1998).

A second simulation was run where the demand flow peaked at 1800 veh/hr. The resulting flow-density plot was similar to that shown in Figure 5, but this time there were no points above 1800 veh/hr: instead, there was a roughly horizontal line of points at this value, connecting the two arms of the linear-segment relationship. The overall "shape" of this scatter resembled the three-linear-segment flow density relationship used in the cell-transmission model (Daganzo, 1994). It appears then, that when the data are aggregated in an arbitrary way, the flow-density plot may take on a variety of curvilinear "shapes", depending on the local demand and supply conditions.

7 Conclusions

A simplified model of traffic flow was presented, defined on a discrete notion of flow and continuous space and time. The model – named the space-time queue (STQ) – was solved in two ways. One solution is based on a discretization of space into cells equal in length to an average vehicle, and is called the cell-based STQ (CSTQ). The second solution, called the arc-based STQ (ASTQ), is defined only upon the counting processes at the entrance and exit of an arc. The computational effort required to solve the latter solution is thus constant in the arc length, and linear in the number of vehicles to pass through the arc. This unique property is shared only by a method suggested by Newell for solving a simplified kinematic wave model (Newell, 1993), which is based on a continuous notion of flow.

A point queue model equivalent to the STQ was developed by graphical means, and it was thus demonstrated that when the effect of a finite arc length can be ignored, waiting time in the STQ is identical to that obtained using a point queue (with the appropriate definitions of inter-arrival and service time), and the spatial extension of delay propagation behaves exactly as the queue length in a point queue. The interpretation of the length of a point queue as the spatial extension of the STQ is quite useful, as it may be used to estimate when, or with what probability, the delays at the exit of the arc will "spill-back" to the entrance. In general, however, the conditions under which steady-state formulas are applicable are not often found in realistic situations of practical interest (Hurdle, 1990).

The steady-state relationship between flow and density was derived for the model, and is given by a two-linear-segment function. Not surprisingly, this relationship is identical to the fluid-dynamical limit of CA traffic models (Nagel, 1996). Moreover, it is of the same general form as the relationship underlying Newell's simplified kinematic wave model (Newell, 1993). Numerical results were obtained for the flow-density relationship by simulation, and were found to produce a curvilinear pattern, much like the empirical "fundamental diagram" in the traffic engineering literature (May, 1990). The reason for the discrepancy between the numerical results and the derived two-linear-segment relationship is that the space-time domain over which the points were measured was quite large, and hence some points represent averages between the two linear segments of the derived relationship. This supports the idea that the observed shape of empirical flow-density relationships may be quite sensitive to how the data are aggregated, as suggested by Cassidy (1998).

Another similarity with Newell's model is the uniqueness of the solution, which is not true of macroscopic models in general (Lebaque, 1996). The STQ most closely resembles CA traffic models, in particular the deterministic CA-184 (Nagel, 1996). One distinction between the STQ and CA traffic models concerns the question of calibration. Calibration is the process of choosing appropriate values for the model's parameters, in order to reproduce empirical data. The discretization of flow, space and time in CA restricts the possible values of maximum speed and minimum spacing to discrete values as well. These parameters can be calibrated to any desired accuracy in the STQ, because time is real-valued in this model.

Overall, the STQ appears to have considerable potential as the basic mechanism of flow propagation for a high-speed network simulation model. Such models are currently needed for dynamic traffic assignment modules that must be executed in real-time, in support of Intelligent Transportation System (ITS) applications, such as the determination of real-time route-guidance and network control strategies. This model is readily applicable, in principle, to a general network. Moreover, the use of discrete vehicles makes the model more amenable to the use of fairly complex models of driver/vehicle behaviour at nodes, such as those used in micro-simulation traffic models (Barceló et al, 1994). An important restriction of the ASTQ is that in its current form, it does not permit vehicles to enter or exit the traffic stream between nodes, and thus cannot model the effects of lane changing on multi-lane arcs. The CSTQ, on the other hand, could easily be modified to model this behaviour, but this model would then have no real advantage over the use of existing cell-based approaches such as CA traffic models.

References

Barceló, J., Ferrer, J. L., and R. Grau (1994). AIMSUN2 and the GETRAM Simulation Environment. Internal Report, Departamento de Estadística e Investigación Operativa. Universitat Politècnica de Catalunya. See also http://tss-bcn.com.

Cassidy, M. (1998). Bivariate relations in nearly stationary highway traffic. Transportation Research *B*, 32B(1):49-59.

Daganzo, C. F. (1994) The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research B, 28B(4):269-287.

Edie, L. C. (1974). Flow Theories, in *Traffic Science* (D. C. Gazis, ed.). John Wiley & Sons, New York. pp. 7-11.

Gerlough, D. L., and Huber, M. J. (1975). *Traffic Flow Theory*. Transportation Research Board Special Report #165, Transportation Research Board, Washington, D.C.

Hurdle, V. F. (1990). Queueing Theory Applications, in *Concise Encyclopedia of Traffic and Transportation Systems* (M. Papageorgiou, ed.). Pergamon, 1991. pp. 337-341.

Lebacque (1996). The Godunov scheme and what it means for first order traffic flow models. *Transportation and Traffic Flow Theory* (J. P. Lesort ed.), Proceedings of the 12th International Symposium on Transportation and Traffic Theory (ISTTT), pp. 647-677.

Lindley, D. V. (1952). The Theory of Queues with a Single Server. *Proceedings of the Cambridge Philosophical Society*, 48:277-89.

Little, J. D. C. (1961). A proof for the queueing formula $L = \lambda W$. *Operations Research*, 9:383-7.

Mahut, M. (1999a). Speed-maximizing car-following models based on safe stopping rules. TRB Preprint 990351. Transportation Research Board, 78th Annual Meeting, January 10-14, 1999.

Mahut, M. (1999b). Behavioural Car Following Models. Report CRT-99-31. Centre for Research on Transportation. University of Montreal. Montreal, Canada.

May, A. D. (1990) *Traffic Flow Fundamentals*. Prentice Hall, Englewood Cliffs.

Nagel, K. (1996). Particle hopping models and traffic flow theory. *Physical Review E*, 53(5):4655-4672.

Nagel, K., and Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221-2229.

Newell, G. F. (1993). A simplified theory of kinematic waves in highway traffic. Part I: General Theory. Part II: Queueing at freeway bottlenecks. Part III: Multi-destination flows. *Transportation Research B*, 27B(4):281-313.

Schreckenberg, M., Schadschneider, A., Nagel, K., and N. Ito. (1995). Discrete stochastic models for traffic flow. *Physical Review E*, 51:2939-2949.

Figures

Figure 1: Space-Time Trajectories

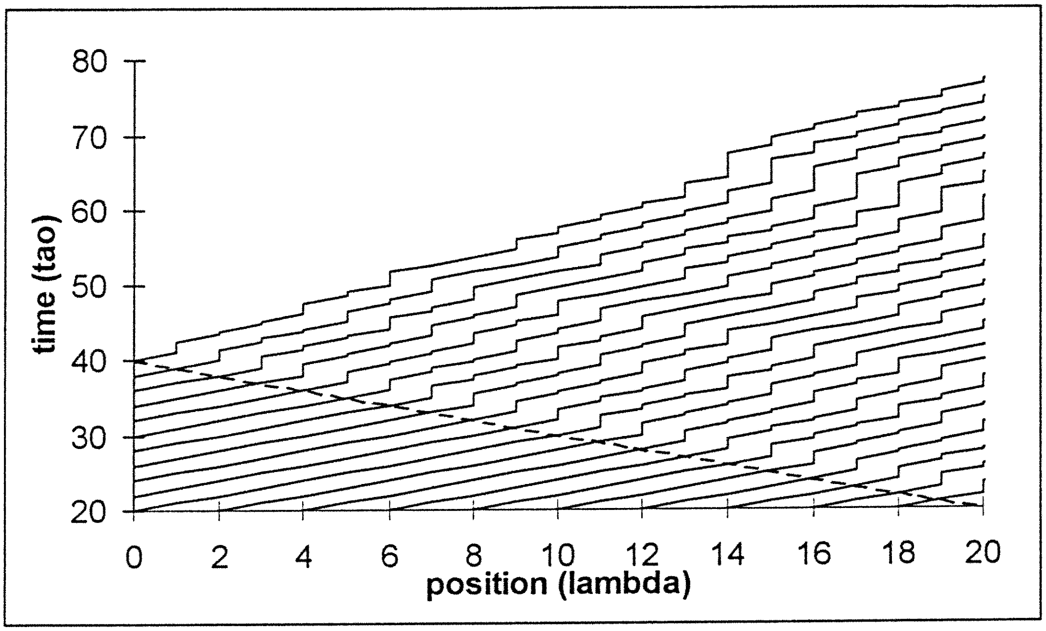


Figure 2: The "Point" Space-Time Queue

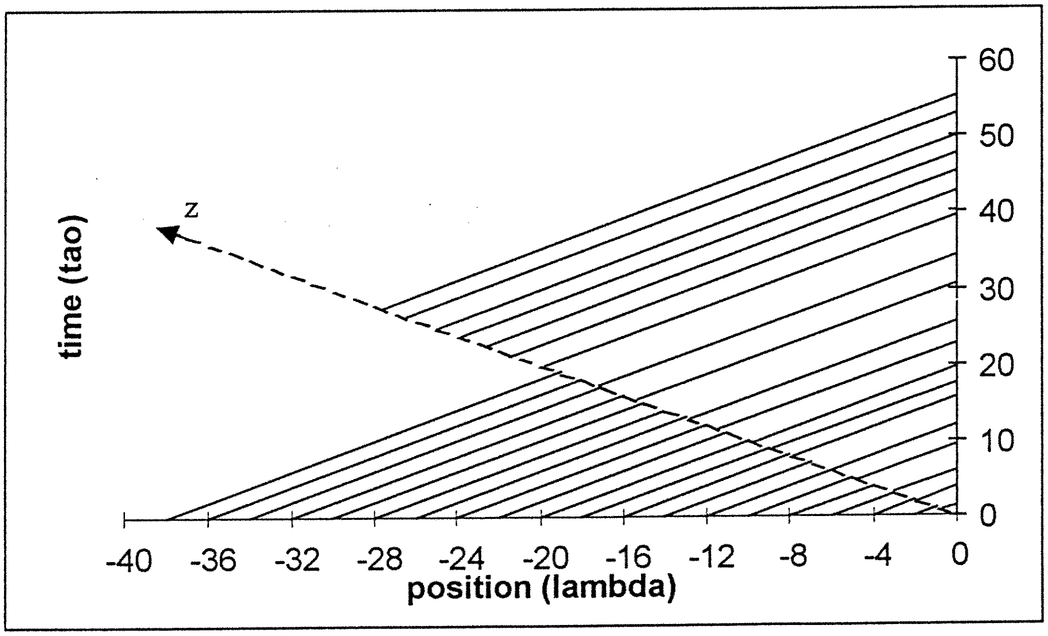


Figure 3: Space-Time Domain For Constrained Trajectories

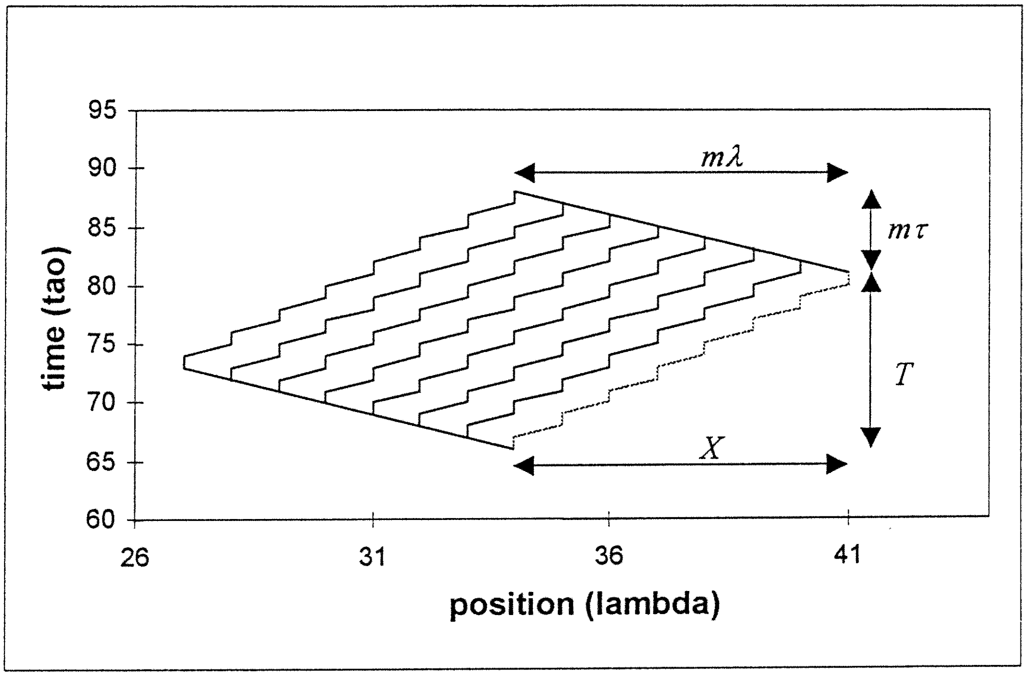


Figure 4: Space-Time Areas For Constrained Trajectories

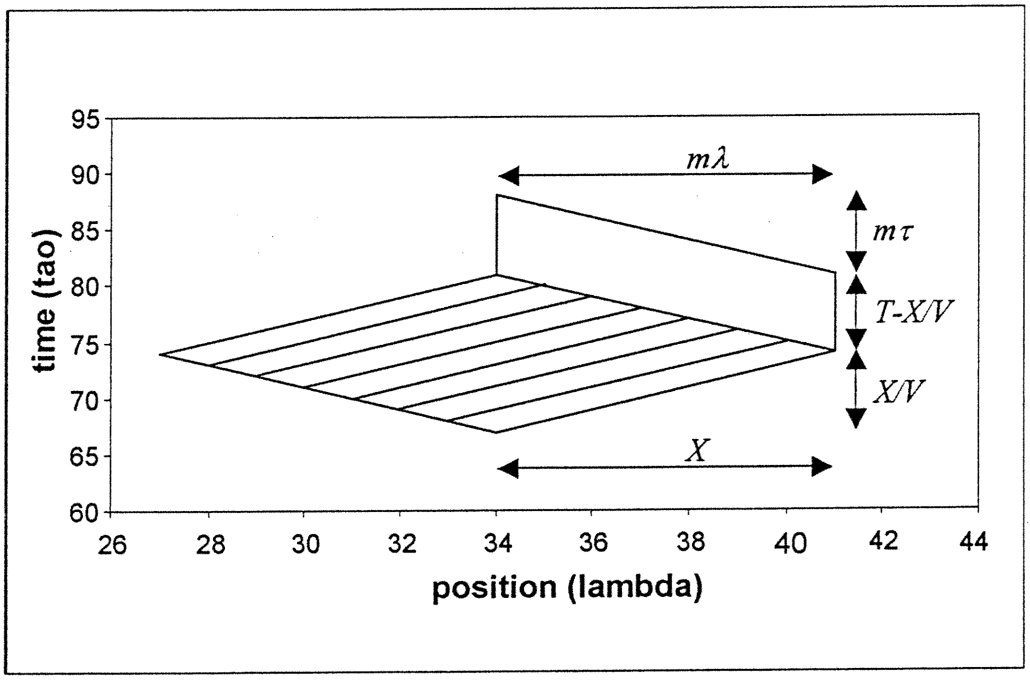
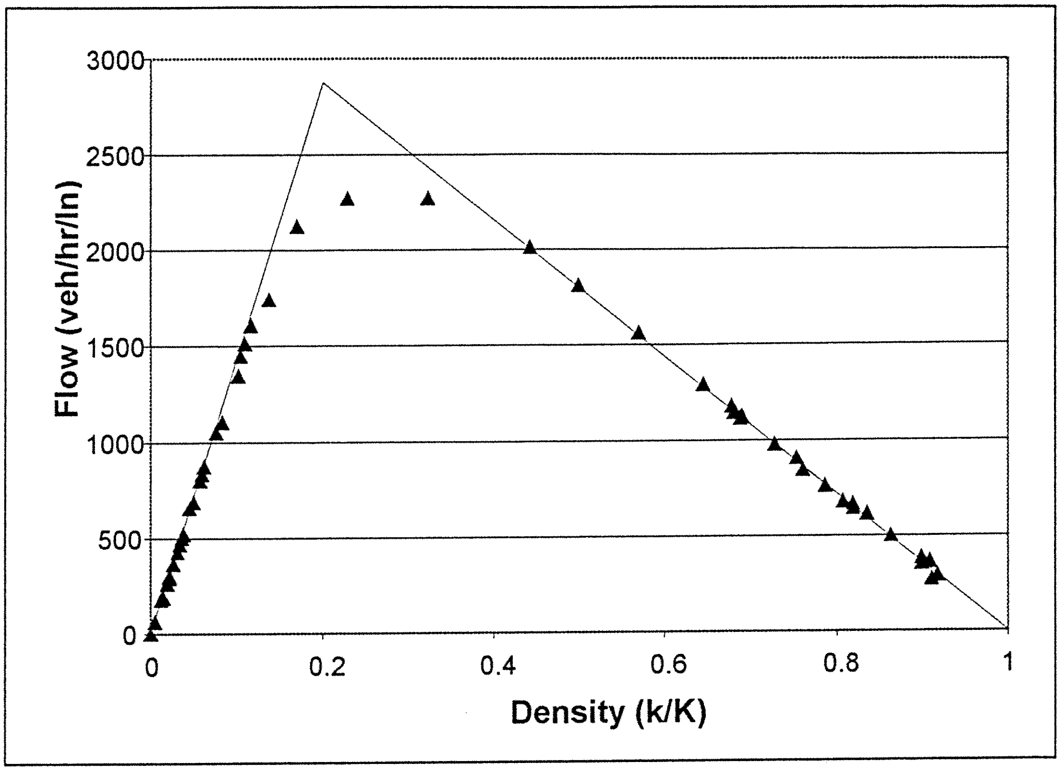


Figure 5: The Fundamental Diagram



A Discrete Flow Arc Model of Traffic Dynamics Based on the Space-Time Queue

Michael Mahut, University of Montreal, Centre for Research on Transportation
Montreal, Canada

Abstract

This work is an extension of a simplified discrete-flow model of traffic dynamics recently proposed by the author. The main limitation of the earlier work is that it only applies in a strict way to single-lane flow, or to situations that can reasonably be modeled as such. The model proposed here captures the reductions in flow due to lane-changing maneuvers that result in crossing, or "weaving", patterns of vehicle trajectories within an arc. This is done without losing the arc-based nature of the solution algorithm to the original one-lane model. As with the original model, the assumptions used here are consistent with a "microscopic" description of the trajectory of each vehicle. A small example in the form of a microscopic animation is presented and solved by hand. Simulation results obtained by computer are presented for a larger example, and are found to compare very well with those obtained with the AIMSUN2 traffic simulator. Execution time of the proposed model is below one tenth of that required by this traffic simulator.

Acknowledgements

This work was supported in part by a post-graduate scholarship of the Natural Sciences and Engineering Research Council of Canada (NSERC). The author would like to thank Professor Michael Florian, as well as Shane Velan and Karim Er-Rafia, of the University of Montreal, Canada, and Vittorio Astarita of the Universita degli Studi della Calabria, Italy, for many valuable comments and discussions during the preparation of this work.

1 Introduction

This work is an extension of a simplified discrete-flow model of traffic dynamics recently proposed by the author (Mahut, 2000). The main limitation of the earlier work is that it only applies in a strict way to single-lane flow, or to situations that can reasonably be modeled as such. It is well known that multi-lane flow cannot be modeled in this way in general, due to the possibility of conflicting movements within an arc, often referred to as "weaving" conditions (Transportation Research Board, 1998). This situation arises when the turning movements at the exit of an arc require vehicle trajectories to cross within the arc. The effects are most visible in non-signalized ("uninterrupted flow") conditions, such as highways (*ibid.*), and result in a reduced overall flow capacity of the arc. Although there is some published work on the modeling of multi-lane discrete-flow traffic, it is in the context of micro-simulation approaches where each vehicle's trajectory is updated frequently (Gipps 1986; Rickert *et al*, 1997). The main concern of these efforts is upon modeling the driver decision as to when and why to change lanes, as a function of local information that is updated frequently.

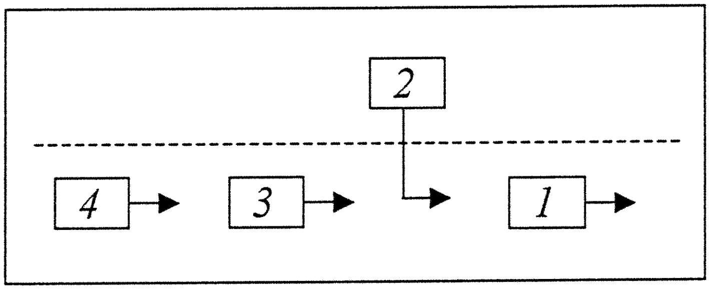
The main objective of the work presented here is to maintain the arc-based property of the solution algorithm of the underlying (one-lane) traffic flow model, i.e., that the computational effort required per vehicle per arc be independent of the time spent by the vehicle on the arc. An important advantage of such a model over the well-known micro-simulation approaches is that real-time applications, such as route-guidance and adaptive signal control, could benefit from the significantly shorter execution times that can be obtained. Another disadvantage of micro-simulation models is that it can be quite difficult to calibrate these models for medium to large-scale networks (Hugusson and Andersson, 1999). In general, simpler models should be easier to calibrate. Since this approach does not permit the vehicle trajectories to be regularly updated, the structure of this model is quite different from those referenced above. The main focus of this work is on modeling appropriately the reduction in flow capacity that results from a given traffic demand that is lane-based, i.e., given the local flows by exit lane for the flows arriving by each entrance lane. As a result, the lane choice is taken as given. The intention is to represent only mandatory lane changes, due to the lane permissions associated with the path a vehicle is following. The question of when (or where) within the arc the actual lane-change maneuver takes place must also be considered in this type of model.

This reduction in flow is due to the fact that when a vehicle moves from one lane to a neighbouring lane, it temporarily blocks both of these lanes, effectively taking up twice as much space as it does when moving forwards within a single lane. If a vehicle is to move across two lanes at once, it will occupy three times as much space, etc… This space must be empty before a vehicle can occupy it. Thus, if one vehicle is occupying some amount of space that a neighbouring vehicle is about to occupy, one of the vehicles must decelerate or accelerate in order to provide sufficient longitudinal separation between the two vehicles such that the lane change may take place. Since the traffic flow model upon which this work is based assumes that all vehicles are moving as quickly as the prevailing constraints will permit, it is not possible that this separation be achieved through the acceleration of one of these vehicles. Thus, one of the two vehicles must decelerate, and hence incur delay.

An example of this situation is shown in Figure 1.1. The vehicles are moving from left to right, and the lanes are numbered from bottom to top, starting at 1. The vehicle labeled 2 must move from lane 2 to lane 1, continuing its trajectory between the vehicles labeled 1 and 3. It should be noted that the minimum (longitudinal) separation between two vehicles is generally a function of vehicle speed, and thus the space required is greater than the physical length of the vehicle as seen in the figure. Thus, in order for vehicle 2 to move into lane 1 between vehicles 1 and 3, vehicle 3 must decelerate. Alternatively, vehicle 2 may decelerate and attempt to move to lane 1 behind vehicle 3. In this case, vehicle 4 may decelerate, or, vehicle 2 may decelerate again, and attempt to change lanes behind vehicle 4. This process may continue as long as there is a sequence of vehicles in lane 1 that are sufficiently close together.

Micro-simulation models of traffic flow resolve the decision as to which vehicle will slow down in such a situation using *gap-acceptance* models (Barceló *et al*, 1994; Van Aerde, 1999). A gap acceptance model is defined upon two vehicles, one of which must have priority over the other. The decision is then made by the lower priority vehicle as to whether the space – or *gap* – is sufficient to allow the lane change to take place. In the example of Figure 1.1, it would generally be assumed that vehicle 2 has lower priority than those vehicles already in lane 1. Given, for example, that the driver of vehicle 2 accepts the gap between vehicles 1 and 3, there is no guarantee that the driver of vehicle 3 will find the remaining space between vehicles 2 and 3 sufficient. The deceleration of vehicle 3 under these circumstances is not determined by a gap-acceptance model, but rather by a car-following model. As a result, vehicle 3 may decelerate after vehicle 2 has begun to move laterally, but the effect is essentially the same as if vehicle 3 had decelerated first in order to provide the necessary space.

Figure 1.1: Lane Changing



Because a driver cannot accept a gap that is smaller than the physical space required for his vehicle, an unlimited number of gaps may be refused when the unoccupied space between two sequential vehicles is less than the vehicle length. Under these conditions, a driver attempting to change lanes may be delayed indefinitely, blocking the lane from which the lane change is to be made. This in turn can have a drastic effect on the through flow of this lane, which in turn may lead to severe congestion. In this situation, the assumptions underlying the gap-acceptance logic fail to accurately represent what happens in reality, when the driver of a high priority vehicle decelerates in order to permit the lower priority vehicle to change lanes (Hugusson and Andersson 1999). For this reason, a simulation model may predict much higher congestion than occurs in reality. As a result, more recent lane-changing rules used in micro-simulation models tend to include some form of *courtesy yielding* logic for the higher priority vehicles.

It should be noted that each vehicle in a simulation model is typically given a route to follow, and that the turning movement from one arc to another is usually restricted to certain lanes. Thus, severe congestion can occur because a single vehicle is unable to change lanes in order to leave the arc by a lane that permits a required turning. One way of reducing the severity of this problem is to allow vehicles to change their routes when a required lane change cannot be made without incurring excessive delay. This logic is used, in conjunction with the courtesy yielding rules, by at least one micro-simulator (Barceló *et al*, 1994). Although it is arguable that this type of logic is realistic – i.e., drivers do in fact change, or lose, their paths due to the inability to access the necessary lanes – it can underestimate congestion and lead to unrealistic flows if the original paths are abandoned too quickly. Recent experiences with some micro-simulators have indicated that the prediction of traffic conditions can be significantly improved – in some particular cases – when the lane choice logic considers not only the next arc on a vehicle's path, but the next two arcs (Ben Akiva *et al*, 2000; Barceló *et al*, 2000). This was found to be of particular importance when the arcs in question were fairly short, and drivers (in reality) were choosing lanes by which to exit one arc based on the knowledge of which lane would be used to exit the next arc.

Another decision that must be modelled by the lane-changing rules concerns the location at which a driver first attempts to make a lane change, given that the desired exit lane has not yet been reached. For this purpose, the arc is usually divided spatially into three zones (Barceló *et al*, 1994; Van Aerde, 1999). In the first zone, the driver does not attempt to changes lanes; in the second zone, the driver attempts to change lanes, but does so in a non-aggressive manner; in the third zone, the driver attempts to make the necessary lane changes in a more aggressive manner. The smaller the gap a driver is willing to accept, the more aggressive his behaviour is said to be. Thus, as the vehicle approaches the (downstream) end of the arc, the driver becomes increasingly aggressive about looking for gaps, but in a discrete, deterministic way. In general, experience with traffic simulation has indicated that fairly sophisticated driver decision processes must be modelled if the traffic conditions are to be predicted with any accuracy.

The idea of choosing the position at which the lane change may first be executed purely as a function of the distance from the end of the arc may also eventually require revision. Imagining a signalized roadway, where there is enough demand that significant queues are usually present at the start of the green light, it makes intuitive sense that a driver will attempt to make any necessary lane changes *before* arriving to the back of the queue. This makes sense for the simple reason that the spacing between vehicles increases with speed, which is a direct consequence of the well-known fundamental diagram of vehicular traffic (May, 1990). Thus, the driver is much more likely to find an acceptable gap before arriving to the back of the queue.

The model presented below is concerned primarily with modelling the traffic flow given the lanes by which each vehicle enters and exits an arc. Because the solution algorithm is strictly arc-based, the choice of exit lane must be known at the moment a vehicle enters an arc, and cannot be revised afterwards. Rather than using gap acceptance logic, which can potentially lead to excessive congestion, a simple first-in-first-out (FIFO) rule is used instead. Under this assumption, the vehicle to decelerate is always the one that arrived later to the arc. The question of where a vehicle ought to change lanes is addressed by an assumption that takes into account the downstream traffic conditions, and prevents a vehicle from blocking the lane by which it entered the arc.

2 Model Definition

In this section, the simplified discrete flow traffic model proposed by Mahut (2000) – referred to as the space-time queue (STQ) – will be extended to the case of a multi-lane arc. The STQ is based primarily on the following two constraints, referred to as the cell-based demand and supply constraints, respectively:

(2.1) $$t^D(n,x) \geq t^D(n,x-\lambda)+\frac{\lambda}{V} \qquad \forall n, \forall x > 0,$$

(2.2) $$t^D(n,x) \geq t^D(n-1,x)+\tau+\frac{\lambda}{V} \qquad \forall n > 1, \forall x < X,$$

and the objective:

(2.3) $$MIN[t^D(n,x)]$$

where:
$t^D(n,x)$ = departure time of vehicle n from position x.
τ = reaction time of the driver/vehicle system
λ = effective vehicle length (physical length and buffer space)
V = maximum vehicle speed (speed limit), in λ/τ
X = the length of the arc, in λ

The relationship in (2.1) describes the constraint on the trajectory of vehicle n as determined by the vehicle preceeding it, while (2.2) describes the constraint of the trajectory of vehicle n on itself. It may be noted that in both cases, the right-hand side refers to an earlier moment in time than the left-hand side, indicating that causes must precede effects. When the order of vehicles between two fixed in space points – $x=0$ and $x=X$ – is unchanging, as would be the case on a one-lane road, the strict FIFO (first-in-first-out) property implied by this ordering can be exploited to yield the following relationships, referred to as the arc-based space-time queue (ASTQ) (*ibid.*):

(2.4) $$t^D(n,X) = MAX\left[t^D(n,0)+\frac{X\lambda}{V}, t^D(n-1,X)+\left(\tau+\frac{\lambda}{V}\right)\right]+\omega(n) \qquad n>1$$

(2.5) $$t^D(n,X) = t^D(n,0)+\frac{X}{V} \qquad n=1$$

(2.6) $$t^D(n,0) = MAX\left[t^{dm}(n,0), t^D(n-1,0)+\left(\tau+\frac{\lambda}{V}\right), t^D(n-X,X\lambda)+X\tau\right] \qquad n>X$$

(2.7) $$t^D(n,0) = t^{dm}(n,0) \qquad n \leq X$$

where:
$x = 0$ = the upstream, or entrance, position of the arc
$x = X$ = the downstream, or exit, position of the arc
$t^{dm}(n,0)$ = the minimum value of $t^D(n,0)$ as due to processes upstream of the arc, such as a vehicle generation procedure, referred to as the *demand time* of vehicle n
$\omega(n)$ = the delay incurred by vehicle n at $x = X$ due to processes downtream of the arc, such as the downstream node.

The following discussion will focus on the extension of equations (2.4) and (2.6), of which (2.5) and (2.7) are special cases, respectively.

2.1 The Demand Constraint

This section is concerned primarily with the extension of the first term inside the *max* operator in equation (2.4) – referred to as the *demand constraint* – to define the earliest time a vehicle may exit an arc, $t^D(n,X)$, as a function not only of its entrance time, $t^D(n,0)$, but also of the decelerations incurred due to lane changing maneuvers. Some preliminary notation is as follows:

n = the vehicle's arc-based identifier: vehicles are labelled in the order in which they enter the arc.
$m^A(n)$ = the lane by which vehicle n enters the arc, called the *arrival* lane.
$m^D(n)$ = the lane by which vehicle n exits the arc, called the *departure* lane.

Some preliminary assumptions are as follows; for convenience, they are grouped into paragraphs and labeled:

(A.1) Each vehicle is assigned (or chooses) an arrival lane and a departure lane before entering the arc. The departure lane choice cannot be reconsidered once the vehicle has entered the arc. FIFO is enforced by destination lane, i.e., all vehicles destined for the same lane must exit the arc in the order in which they entered. Overtaking may thus only occur between vehicles destined for different destination lanes.

(A.2) The cell-supply constraint (2.2) ensures a minimum time separation between two vehicles that sequentially pass the same position in space. This constraint is extended to the multi-lane case by defining the departure time from position x of vehicle n on lane m as $t^D(n,x,m)$. Considering the FIFO rule mentioned above, the only vehicles that may have been the last to occupy a specific position (x,m) on the arc before vehicle n are the last vehicles to have arrived to the arc for each origin-destination lane pair, before the arrival of vehicle n. These vehicles are referred to as the *leaders* of vehicle n, and are denoted by $l_{ij}(n)$. The set of leaders is denoted by $L(n)$. The (multi-lane) cell-supply constraint may now be stated formally as:

$$(2.8) \quad t^D(n,x,m) \ge MAX\left[t^D\left(l_{ij}(n),x,m\right)\right] + \left(\tau + \frac{\lambda}{V}\right) \qquad \forall(i,j).$$

If $l_{ij}(n)$ never occupies position (x,m), then $t^D\left(l_{ij}(n),x,m\right)$ is considered to be zero by default.

(A.3) Vehicles n and $l_{ij}(n)$ are said to be in *spatial conflict* if the vehicles' trajectories diverge (share the same arrival lane), merge (share the same departure lane), or cross in space along the arc. The subset of leaders that satisfies this condition will be reffered to as the set of spatial leaders of n, denoted by $SL(n)$. The arrival *gap* between vehicles n and $l_{ij}(n)$, denoted by $\alpha_{ij}(n)$, is furthermore defined as:

$$(2.9) \quad \alpha_{ij}(n) = t^D(n,0,I(n)) - t^D\left(l_{ij(n)},0,i\right) - \left(\tau + \frac{\lambda}{V}\right).$$

The set of spatial-temporal leaders of vehicle n, denoted by $STL(n)$, is now defined as:

$$(2.10) \quad STL(n) = \left\{l_{ij}(n) \mid \alpha_{ij}(n) < 0\right\}$$

In plain language, there is a conflict between vehicle n and each member of $STL(n)$, such that one of the two vehicles must be delayed if the two trajectories are to cross in space without violating the cell-supply constraint (2.8).

(A.4) If a vehicle's departure lane is not the same as its arrival lane, the vehicle will move from its arrival lane to its departure lane in one movement, i.e., at one instant in time. Moreover, this movement will occur at one position along the arc, denoted by $x^0(n)$. Thus, the lateral movement of a vehicle is assumed to be instantaneous. This assumption is also used in multi-lane cellular automata traffic models (Rickert *et al*, 1997). Considering the lateral movement of vehicle n from $m^A(n)$ to $m^D(n)$, this implies that:

$$(2.11) \quad t^D\left(n,x^0(n),m\right) = t^D\left(n,x^0(n),m^A(n)\right), \qquad \forall m \in \left\{m^A(n),\dots,m^D(n)\right\} \setminus m^D(n),$$

i.e., no delay is incurred on any of the intermediate lanes since the vehicle has no forward motion on these lanes. The position $x^0(n)$ is moreover the first position at which vehicle n incurs any delay, i.e., every vehicle moves at its maximum speed in its arrival lane from $x=0$ to $x=x^0(n)$, and thus:

$$(2.12)\quad t^A\left(n,x^0(n),m^A(n)\right)=t^D(n,0,i)+\frac{x^0(n)}{V}$$

The position $x^0(n)$ is furthermore defined as:

$$(2.13)\quad x^0(n)=MIN\left\lfloor x^0\left(l_{ij}(n)\right)\right\rfloor-\lambda \qquad \forall l_{ij}(n)\in STL(n)$$

(A.5) All lane-changing conflicts are resolved by using a strict FIFO rule, i.e., no gap acceptance models are applied.

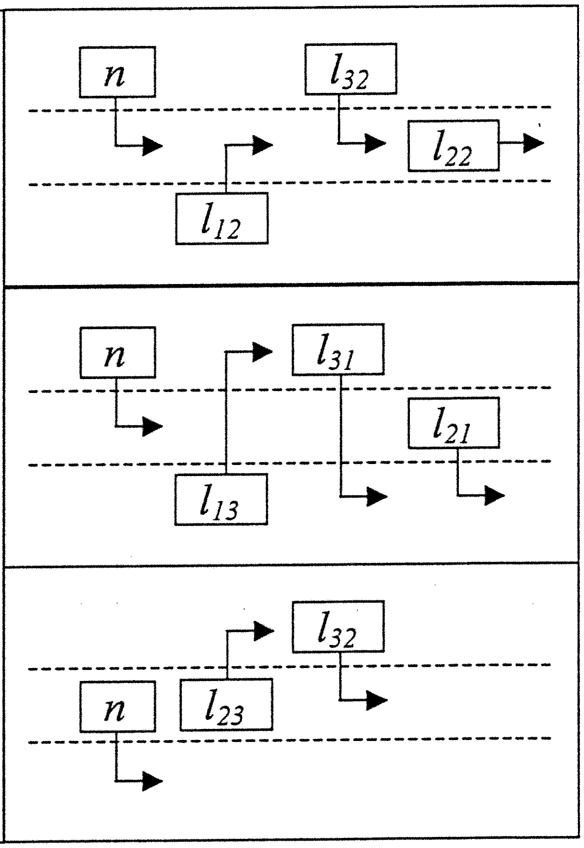
In order to determine how delays due to lane-changing are passed from one vehicle to another, it is necessary to make some assumptions about driver behaviour that are appropriate for a simplified model. The fundamental assumption made here is that once a vehicle has stopped at $x^0(n)$, it will leave its arrival lane as soon possible – as permitted by the cell-supply constraint (2.8) – and thus incurs as little delay as possible in this lane. The remainder of the delay is incurred in the departure lane. The total delay due to lane-changing – referred to as the *total arc delay* – is denoted by $\Delta(n)$, while the amount incurred only in the origin lane – referred to as the *initial arc delay* – is denoted by $\Delta^0(n)$. Figure 2.1 shows the three cases for calculating the initial and total arc delay of a vehicle, as a function of the initial and total arc delays of its spatial-temporal leaders, and of the geometry of the spatial conflict between it and each of these leaders. These calculations are based on the direct application of the cell-supply constraint (2.8). On the left side of the figure is an algorithm that identifies three types of spatial conflicts. On the right side, examples are given for each of the three cases, in the same order that they are identified in the algorithm. Vehicle motion is again from left to right, and the lanes are again numbered from bottom to top, starting at 1.

The three cases are distinguished by the contribution of $l_{ij}(n)$ to $\Delta(n)$: $\Delta\left(l_{ij}(n)\right)-\alpha_{ij}$ is used when a spatial conflict remains after both vehicles have changed lanes (i.e., both vehicles are destined for the same lane); $\Delta^0\left(l_{ij}(n)\right)-\alpha_{ij}$ is used when a spatial conflict remains after vehicle n has changed lanes, but not after $l_{ij}(n)$ has changed lanes; $-\alpha_{ij}$ is used when no spatial conflict remains after vehicle n has changed lanes. It should be noted that if $l_{ij}(n)\in STL(n)$, vehicle n always changes lanes before $l_{ij}(n)$. This property is a direct consequence of the assumption that each vehicle incurs as little delay as possible in its origin lane, and the definition of $x^0(n)$ given in (2.13). For this reason, there are no cases concerned with whether a spatial conflict exists after $l_{ij}(n)$ has changed lanes, but before n has changed lanes. In all three cases, $\Delta^0(n)$ is always determined by $-\alpha_{ij}$, given that a spatial conflict exists. Thus, vehicle n must wait in its arrival lane exactly as long as it takes to satisfy (2.8) with respect to all of its spatial leaders. When vehicle n has no lane change to execute, the initial arc delay is set equal to the total arc delay, by convention.

It can be seen that the relationships given for $\Delta(n)$ and $\Delta^0(n)$ are simply adaptations of Lindley's waiting time recursion for a single-server FIFO queue (Lindley, 1952), where the leader's delay as perceived by the follower depends on the geometry of the spatial conflict between the two vehicles.

Figure 2.1: Delay Rules for Lane Changing

```
$\forall (i,j)$ do :
    if $\langle m^D(n) = m^D(l_{ij}) \rangle$
        $\Delta(n) = \max\left[\Delta(n), \Delta(l_{ij}) - \alpha_{ij}, 0\right]$
        $\Delta^0(n) = \max\left[\Delta^0(n), -\alpha_{ij}, 0\right]$
    elseif $\langle m^D(n) \in \{i, \ldots, j\} \rangle$
        $\Delta(n) = \max\left[\Delta(n), \Delta^0(l_{ij}) - \alpha_{ij}, 0\right]$
        $\Delta^0(n) = \max\left[\Delta^0(n), -\alpha_{ij}, 0\right]$
    elseif $\langle n \sim l_{ij} \rangle$
        $\Delta(n) = \max\left[\Delta(n), -\alpha_{ij}, 0\right]$
        $\Delta^0(n) = \max\left[\Delta^0(n), -\alpha_{ij}, 0\right]$
    endif
end
```



The last vehicle to have arrived to the arc destined for $m^D(n)$, before the arrival of vehicle n, is denoted by $l^D(n)$. This vehicle will be the last to have exited the arc by $m^D(n)$, before the departure of vehicle n. The extended version of the demand constraint

(2.4) on the departure time of vehicle n from the exit of the arc may now be stated as follows (the lane identifier is not included in $t^D(\ )$ when it is not necessary):

$$(2.14) \qquad t^D(n,X) \ge MAX\left[t^D(n,0)+\frac{X\lambda}{V}+\Delta(n), t^D\left(l^D(n),X\right)+\left(\tau+\frac{\lambda}{V}\right)\right]+\omega(n)$$

2.2 The Supply Constraint

The arc-supply constraint (2.6) was derived in the context of a sequence of vehicles in single file, where each sequential pair of vehicles must respect the (original) cell-supply constraint (2.2). By assuming that each vehicle moves as quickly as possible at all times – see (2.3) – it is ensured that if (2.2) is active for any pairs of sequential vehicles in the sequence $(n(t), n-1(t-\tau),\ldots,n-X(t-X\tau))$, these pairs form a continuous sequence of the form $(n-X(t-X\tau), n-X+1(t-(X-1)\tau),\ldots)$. This is because no independent causes of delay are permitted within the arc, and thus a vehicle can only be delayed at position x, if its leader is delayed at position $x+\lambda$, and if the delay the leader incurs at $x+\lambda$ is greater than the arrival gap, $t^D(n,0)-t^D(n-1,0)-\tau-\lambda/V$ – c.f. (2.13). In this way, all delays incurred on the arc are ultimately derived from the sequence $\omega(n)$. The basic idea underlying the extension of (2.6) to multiple lanes is that, rather than treating the arc delays as an independent mechanism of delay, they are treated as temporary spatial conflicts that can potentially pass along residual amounts of the $\omega(n)$ from a leader to its followers, in exactly the same way as just described for the one-lane case.

It may be instructive to first consider how the ASTQ – (2.4), (2.5), (2.6), (2.7) – can be applied to describe the motion of each vehicle on a cell-by-cell basis. Equations (2.4) and (2.6) can be combined to define the entire trajectory of vehicle n as follows:

$$(2.15) \quad t^D(n,x)=\max\left[t^D(n,0)+\frac{x\lambda}{V}, t^D(n-(X-x),X)+(X-x)\tau\right] \qquad 0<x<X.$$

It is straightforward to show that (2.2) is redundant for $0<x<X$, as long as it respected at $x=0$ and $x=X$ by the use of (2.4) and (2.6) applied to each lane. The two terms inside the *max* operator of (2.15) are referred to as the *non-local* demand and supply constraints, respectively. At every position x, $0<x<X$, one of these two constraints must be active. The trajectory of a vehicle is *constrained* when the non-local supply constraint active; when the non-local demand constraint is active, the trajectory is said to be unconstrained. By convention, when both constraints are active, the trajectory is said to be unconstrained, since a vehicle in this state is not incurring any delay. Thus, if the supply constraint is inactive at $x=0$, it may become active at some later position $x<X$, before the vehicle leaves the arc. This is the position at which the vehicle first incurs delay on the arc, and it is only of direct interest if it is desired to know the temporal distribution of the length of the queue. If the empirical measures of the system are restricted to arc-based statistics (as is done here), then it is only of interest when the

position at which delay is first incurred reaches $x = 0$, which is exactly what the ASTQ yields.

Thus, an important assumption made here is that the position at which vehicle n changes lanes, $x^0(n)$, when none of its leaders changes lanes – and thus no valid $x^0(l_{ij})$ exist – is the position at which the non-local supply constraint becomes active. However, the way in which this constraint is defined in (2.15) must be extended to the multi-lane scenario. This is accomplished by identifying multi-lane sequences of vehicles, over which the cell-supply constraint must be applied, by identifying conflicts between each vehicle and its leaders. By counting forwards $x - X$ vehicles along any such sequence starting at vehicle n, it is possible to define a non-local supply constraint for vehicle n with respect to the departure times of vehicles from *each* lane. In the one-lane case, every vehicle has only one leader, every pair of sequential vehicles is in spatial conflict, and there is only one sequence of vehicles with active cell-supply constraints. Thus, the vehicle identifier is sufficient in this case in establishing the position (i.e., order) of vehicle n relative to any other vehicle in the sequence, and thus sufficient as a notation for defining the non-local supply constraint. In the multi-lane case, a vehicle must have associated with it a *count* that defines its order in a sequence relative to each lane of the arc.

It must also be taken into consideration that the sequences so defined are time dependent. After vehicle n has changed lanes, for example, the conflicts with its leaders may change, as may the conflicts with the following vehicles for which n is a leader. Thus, the counts of a vehicle with respect to each lane must be time-dependent. It was observed in the previous section that it is not necessary to know when or where $l_{ij}(n)$ will change lanes in order to determine if it will delay vehicle n, because of the assumption that all vehicles travel at the maximum speed between $x = 0$ and $x = x^0(n)$. By the same rationale, it can be determined at the moment a vehicle enters an arc what the counts associated with this vehicle will be both before and after it changes lanes. The basic idea is that changing lanes does not change the original set of leaders, but only the temporal gap and (arrival, departure) lane pair used to identify the spatial and temporal conflicts with this set. In this way, two sets of counts are determined for each vehicle at the moment it is ready to enter an arc: $c^A(n,k)$ is the count for vehicle n with respect to lane k before it changes lanes, while $c^D(n,k)$ is the count for vehicle n with respect to lane k after it changes lanes.

It is assumed that immediately after changing lanes, a vehicle incurs its residual arc delay, $\Delta(n) - \Delta^0(n)$, at position $x^0(n)$ in lane $m^D(n)$. Once the total arc delay has thus expired, all temporal conflicts between vehicle n and the set $L(n)$ have similarly expired. At this moment, the only conflict of interest is between vehicle n and $l^D(n)$ – its departure-lane based leader. Thus, by identifying vehicles in order of their arrival to the arc, by departure lane, and denoting this identifier by $d(n)$, it is known that a the count of vehicle n with respect to $m^D(n)$ becomes $d(n)$ after $\Delta(n)$ expires, at which time its count with respect to all other lanes becomes zero.

To summarize, the trajectory of each vehicle is broken down temporally into three states: before $\Delta^0(n)$ has expired; after $\Delta^0(n)$ has expired but before $\Delta(n)$ has expired, and after $\Delta(n)$ has expired. Each state is associated with a different vector of counts. Moreover, spatial conflicts identified with the vehicle before $\Delta^0(n)$ has expired are based on $(m^A(n), m^D(n))$, while those identified with the vehicle after $\Delta^0(n)$ has expired are based on $m^D(n)$ alone.

Procedures for determining $c^A(n,k)$ and $c^D(n,k)$ are given below as algorithms *CA(n)* and *CD(n)*, respectively. The basic logic is as follows. When a vehicle is identified as a leader for vehicle *n*, the size of the gap $(\alpha_{ij}(n))$ between the two vehicles determines which of the count vectors is to be used in determining the vector $c^A(n,k)$. This is analogous to the rules given in Figure 2.1, where the geometry of spatial conflict determines what value of arc delay is "passed along" from $l_{ij}(n)$ to *n*. Then, once the value of $\Delta^0(n)$ has been determined, the vector $c^D(n,k)$ can be calculated in much the same way. The only difference is that for determining spatial conflict only $m^D(n)$ is now considered, and that for the determining temporal conflict, $\alpha_{ij}(n)$ has effectively increased by the value of $\Delta^0(n)$. Denoting the number of lanes on a given arc by *M*, it is readily apparent that the worst case complexity of these two algorithms is in the order of M^3. The outer *for* loop, governed by $\forall(i,j)$, results in M^2 iterations, while the inner *for* loops, governed by $\forall k$, result in *M* iterations each.

Algorithm $CA(n)$

```
$\forall(i,j)\, do:$
  $if\langle \Delta(l_{ij}) - \alpha_{ij} \le 0 \rangle$
    $if\langle j \in \{m^A(n),\ldots,m^D(n)\} \rangle$
      $c^A(n,j) = max\lfloor c^A(n,j), n^D(l_{ij}) + 1 \rfloor$
    $endif$
  $elseif\langle \Delta^0(l_{ij}) - \alpha_{ij} \le 0 \rangle$
    $if\langle j \in \{m^A(n),\ldots,m^D(n)\} \rangle$
      $\forall k\, do:$
        $if\langle c^D(l_{ij},k) > 0 \rangle$
          $c^A(n,k) = max\lfloor c^A(n,k), c^D(l_{ij},k) + 1 \rfloor$
        $endif$
    $endif$
  $elseif \langle n \sim l_{ij} \rangle$
    $\forall k \; do:$
      $if \langle c^A(l_{ij}, k) > 0 \rangle$
        $c^A(n,k) = \mathbf{max}\lfloor c^A(n,k), c^A(l_{ij}, k) + 1 \rfloor$
      $endif$
  $endif$
$end$
```

Algorithm $CD(n)$

```
$\forall (i, j) \, do:$
  $if \langle \Delta(l_{ij}) - \alpha_{ij} - \Delta^0(n) \leq 0 \rangle$
    $if \langle j == m^D(n) \rangle$
      $c^D(n, j) = \mathbf{max}\lfloor c^d(n, j), n^D(l_{ij}) + 1 \rfloor$
    $endif$
  $elseif \langle \Delta^0(l_{ij}) - \alpha_{ij} - \Delta^0(n) \leq 0 \rangle$
    $if \langle j == m^D(n) \rangle$
      $\forall k \; do:$
        $if \langle c^D(l_{ij}, k) > 0 \rangle$
          $c^D(n,k) = \mathbf{max}\lfloor c^D(n,k), c^D(l_{ij}, k) + 1 \rfloor$
        $endif$
    $endif$
  $elseif \langle m^D(n) \in \{i, \ldots, j\} \rangle$
    $\forall k \; do:$
      $if \langle c^A(l_{ij}, k) > 0 \rangle$
        $c^D(n,k) = \mathbf{max}\lfloor c^A(n,k), c^A(l_{ij}, k) + 1 \rfloor$
      $endif$
  $endif$
$end$
```

Since the counts $c^A(n,k)$ yield the position of a vehicle in a sequence associated with a specific lane, the departure times that are used in conjunction with these counts must be sequenced not by the index *n*, but by the counting process as seen at the exit position of each lane. As the index of vehicle *n* with respect to the departure lane $m^D(n)$ is denoted by $d(n)$, the notation $n^{-1}(d,k)$ will be used to identify the vehicle associated with a given position in the counting process as seen at the exit of lane k. Denoting the last vehicle to have arrived to the arc by lane $m^A(n)$ before the arrival of vehicle n as $l^A(n)$, the multi-lane version of equation (2.6) can now be written as:

$$(2.16)\ t^D(n,0) = \mathbf{max}\left[t^{dm}(n), t^D\left(l^A(n),0\right) + \left(\tau + \frac{\lambda}{V}\right), \underset{\forall k}{\mathbf{max}}\left[t^D\left(n^{-1}\left(c^A(n,k) - X, k\right), X\right)\right] + X\tau\right]$$

It should be noted that in equation (1.6), the *count* of a vehicle is simply its identifier, n. As a result, this quantity depends only on the count of the next downstream vehicle, but not on the time separation between the two vehicles. Under the conditions described above this property is lost, because the $c^A(n,k)$ are a function of the arrival time, and are thus time dependent. This yields a circularity in equation (2.16), which requires a procedure to ensure consistency, i.e., that the $t^D(n,0)$ obtained is the same as that used to calculate it. The criterion proposed here for the uniqueness of the solution is to choose the lowest possible value of $t^D(n,0)$ that is consistent. In this way, each vehicle enters the arc at the earliest possible time, which seems reasonable.

The basic property of this model that can be exploited is that the counts can only change at discrete points in time, and that these points are given by the solutions of $\Delta(l_{ij}) - \alpha_{ij} = 0$, and $\Delta^0(l_{ij}) - \alpha_{ij} = 0$, $\forall(i,j)$, in terms of $t^D(n,0)$. These solutions determine which of the possible cases is identified by algorithm $CA(n)$. These solutions are of course themselves time independent, and only those that are greater than $t^{dm}(n)$ – which is determined exogenously – are of any practical interest. Thus, algorithm $CA(n)$ can be implemented iteratively as shown in algorithm $TDS(n)$. The process is initialized with $t^{dm}(n)$, and on each pass through the algorithm, the current value of $t^{dm}(n)$ -- denoted by $t_0^{dm}(n)$ – is used to calculate the next value – denoted by $t_1^{dm}(n)$ – as follows:

$$(2.17)\quad t_1^{dm} = \underset{\forall(i,j)}{\mathbf{min}}\left\lfloor t^D(n,0) > t_0^{dm} \mid \left(\Delta(l_{ij}) - \alpha = 0\right) \cup \left(\Delta^0(l_{ij}) - \alpha = 0\right)\right\rfloor,$$

Thus, the lowest solution to the equations used to distinguish between different cases in algorithm $CA(n)$, greater than the current value of $t^{dm}(n)$, is used to initialize the next loop. The process continues as long as the set defined on the right-hand side of (2.17) is non-empty, as long as the *supply time* – denoted by $t^{sp}(n)$ and given by equation (2.16) – is non-increasing, and as long as the demand time is less than the supply time. The maximum number of iterations through the outer *while* loop in this algorithm is $2M^2$. The calculation of the supply time, by equation (2.16), requires M iterations, while the demand time calculation of equation (2.17) requires M^2 iterations. Both of these calculations are dominated by the call to $CA(n)$, which has a worst-case complexity in the order of M^3. The worst-case complexity of algorithm $TDS(n)$ is thus in the order of M^5.

Algorithm $TDS(n)$

```
$t_1^{dm} = 0$
$t_0^{dm} = t^{dm}(n)$
$t_1^{sp} = \infty$
$t_0^{sp} = \infty$
$while\left\langle \left(t_1^{dm} > t_0^{dm}\right) \cap \left(t_1^{sp} \le t_0^{sp}\right) \right\rangle$
    $t_0^{dm} = t_1^{dm}$
    $t_0^{sp} = t_1^{sp}$
    $CA(n)$
    $t_1^{sp} = \mathbf{max}\left[ t^{dm}(n), t^D\left(l^A(n),0\right) + \left(\tau + \frac{\lambda}{V}\right), \underset{\forall k}{\mathbf{max}}\left[t^D\left(n^{-1}\left(c^A(n,k) - X, k\right), X\right)\right] + X\tau \right]$
    $if\left\langle t_1^{sp} \le t_0^{dm} \right\rangle$
        $return\left(t_0^{dm}\right)$
    $endif$
    $t_1^{dm} = \underset{\forall(i,j)}{\mathbf{min}}\left[t^D(n,0) > t_0^{dm} \mid \left(\Delta\left(l_{ij}\right) - \alpha = 0\right) \cup \left(\Delta^0\left(l_{ij}\right) - \alpha = 0\right)\right]$
    $if\left\langle t_1^{dm} \ge t_1^{sp} \right\rangle$
        $return\left(t_1^{sp}\right)$
    $endif$
$endwhile$
$if\left\langle t_1^{sp} \le t_0^{sp} \right\rangle$
    $return\left(\mathbf{max}\left[t_0^{dm}, t_1^{sp}\right]\right)$
$else$
    $return\left(\mathbf{max}\left[t_0^{dm}, t_0^{sp}\right]\right)$
$endif$
```

3 A Numerical Example

In this section, a small example of the above procedures is presented. In addition to giving the results of the calculations for a sequence of vehicles entering a short arc, an "animation" is provided to show the cell-by-cell trajectories of the vehicles that the above assumptions imply. The animation is shown in Figure 3.1. The arc is 9 cells long, and three lanes wide. Vehicle movement is from left to right, and the lanes are numbered from bottom to top, starting at 1. The maximum speed on the arc is λ/τ, and hence the minimum spatial separation between two vehicles (at the same point in time) traveling at this speed is 2λ. The minimum temporal separation (at the same point in space) between two vehicles traveling at this speed is 2τ. For simplicity, all vehicles with the same arrival lane share the same departure lane. Vehicles entering by lane 1 are all destined for lane 3; those entering by lane 2 remain on lane 2; those entering by lane 3 are destined for lane 1. The number used to identify each vehicle is the arc-based identifier, n. The example was conceived in such a way that all delays incurred are multiples of τ, and hence each vehicle is exactly on a cell at each increment of τ in time. Thus, the animation moves forward through time in constant discrete intervals of τ. The numbers on the cells indicate the exact position of vehicle n at the value of t indicated to the left of each picture in the animation. The position $x = 0$ is the last cell upstream of the arc. It can be seen that any delay incurred by vehicle n at this position is the value of $\omega(n)$ for the upstream arc.

Figure 3.2 contains a table showing the arc delays and counts calculated for each vehicle, along with the basic information upon which these calculations are based. In particular, the demand time, arrival lane, and departure lane are indicated in the table. Each row in the table indicates the calculations made with respect to a specific leader. For brevity, the argument n is generally not included in identifiers, and the value of k, where applicable, is written as a sub-script. The departure lane counts in the last three columns are calculated only after the arc delays have been determined for all leaders. Below the calculations for each leader, a row in bold font indicates the maxima over all leaders for the various values as determined for each vehicle n. For brevity, only the last vehicles to have arrived by each lane are considered as leaders in the calculations given, as this is sufficient for obtaining the correct results in this particular example. Vehicles 1 to 9 required no iterations in algorithm *TDS*, as the consistency criteria was satisfied for the initial value of $t^{dm}(n)$, given in the 5th column. For this reason, each leader is considered only once in determining the arc delays and counts for each of these vehicles. Vehicles 10 and 11 each required one iteration, and thus there are two sets of calculations for each leader for these vehicles.

The rules that govern the cell-by-cell movements are summarized as follows. Every vehicle moves forward in its arrival lane until the non-local supply constraint becomes active, at which time the vehicle waits for a duration $\Delta^0(n)$, before moving instantaneously to its departure lane. The vehicle is then delayed for a duration $\Delta(n) - \Delta^0(n)$ in its departure lane, at the same position x at which the lateral movement was made. After the total arc delay has been incurred, each vehicle is free to traverse the remaining distance at the maximum speed. The latter statement is only true because there are no node delays in this particular example, i.e., $\omega(n) = 0$, $\forall n$. In the general case, every vehicle moves forward in its departure lane after its total arc delay has expired, following exactly the same rules as in the original (one-lane) STQ.

Figure 3.1: Animation of Vehicle Trajectories

	x =0	x =1	x =2	x =3	x =4	x =5	x =6	x =7	x =8	x =9
t = 1	2 1									
t = 2	3	2 1								
t = 3	5 4	3	2 1							
t = 4	6	5 4	3	2 1						
t = 5	8 7	6	5 4	3	2 1					
t = 6	9	8 7	6	5 4	3	2 1				
t = 7	11 10	9	8 7	6	5 4	3	2 1			
t = 8	11 10	9	8	7	6	5 4	3	2 1		
t = 9	11 10	9	8	7	6	5	4	3	2 1	
t = 10	11 10	9	8	7	6	5	4	3	2	1
t = 11	11 10		9	8	7 6	5	4	3	2	
t = 12		11 10		9	8	7 6	5	4 3		2
t = 13			11 10		9	8	7 6	5	4 3	
t = 14				11 10		9	8	7 6	5	4 3
t = 15					11 10		9	8	7 6	5
t = 16						11 10		9	8	7 6

Figure 3.2: Calculations for Animation Example

n	m^A	m^D	d	t^{dm}	l_{ij}	α_{ij}	Δ	Δ^0	c_1^A	c_2^A	c_3^A	$t^D(n,0)$	c_1^D	c_2^D	c_3^D
1	1	3	1	1											
							0	**0**			**1**	**1**			**1**
2	3	1	1	1	1	-2	2	2			2				
							2	**2**	**1**		**2**	**1**	**1**		
3	2	2	1	2	1	-1	1				2				
					2	-1	3		2		3				
							3	**3**	**2**	**1**	**3**	**2**		**1**	
4	1	3	2	3	1	0	0	0	0	0	2				
					2	0	2	0	2	0	3		2		3
					3	-1	1	1	3	2	4				
							2	**1**	**3**	**2**	**4**	**3**	**2**		**3**
5	3	1	2	3	2	0	2	0	2	0	3				
					3	-1	1	1	3	2	4				
					4	-2	3	2	4	3	5		4	3	5
							3	**2**	**4**	**3**	**5**	**3**	**4**	**3**	**5**
6	2	2	2	4	3	0	3		3	2	4				
					4	-1	2		4	3	5				
					5	-1	3		5	4	6				
							3	**3**	**5**	**4**	**6**	**4**			**2**
7	1	3	3	5	4	0	2	0	4	3	5				3
					5	0	2	0	5	4	6		5	4	6
					6	-1	1	1	6	5	7				
							2	**1**	**6**	**5**	**7**	**5**	**5**	**4**	**6**
8	3	1	3	5	5	0	3	0	5	4	6				3
					6	-1	1	1	6	5	7				
					7	-2	3	2	7	6	8		7	6	8
							3	**2**	**7**	**6**	**8**	**5**	**7**	**6**	**8**
9	2	2	3	6	6	0	3		6	5	7				3
					7	-1	2		7	6	8				
					8	-1	3		8	7	9				
							3	**3**	**8**	**7**	**9**	**6**			**3**
10	1	3	4	7	7	0	2	0	7	6	8				
					8	0	2	0	8	7	9				
					9	-1	1	1	9	8	10				
							2	**1**	**9**	**8**	**10**	**NA**			
				11	7	4	0	0			4				4
					8	4	0	0	4						
					9	3	0	0		4					
							0	**0**	**4**	**4**	**4**	**11**			**4**
11	3	1	4	7	7	0	3	0	6	5	7				
					8	0	2	0	7	6	8				
					9	-1	1	1	8	7	9				
							3	**1**	**8**	**7**	**9**	**NA**			
				11	8	4	0	0			4		4		
					9	3	0	0	4						
					10	-2	2	2	5	5	5				
							2	**2**	**5**	**5**	**5**	**11**	**4**		

This example provides an opportunity to clarify an important detail regarding the solution algorithm. In the application of (2.16) in algorithm *TDS*, if the vehicle identified as $n^{-1}\left(c^{A}(n,k)-X,k\right)$ for some (n,k) has not yet left the arc, the value of $t^{D}\left(n^{-1}\left(c^{A}(n,k)-X,k\right),X\right)$ is not known. This situation occurs in the evaluation of $t^{D}(n,0)$ for vehicles 10 and 11, with respect to the original demand times of 7 for these vehicles. When this occurs, the supply time is considered to be infinite, and the algorithm continues. If repeated iterations of the algorithm can obtain no lower supply time, the supply time for this (arrival lane, departure lane) pair cannot be evaluated until the vehicle identified as $n^{-1}\left(c^{A}(n,k)-X,k\right)$ leaves the arc. In this situation, the lane k is said to be *inactive*, and a Boolean variable is used to indicate this state. When the next vehicle leaves the arc by this lane, it is then known that a supply time calculation is "waiting" for this event, and algorithm *TDS* is called to re-start the process. This situation occurs when the number of vehicles on the arc destined for some lane k is equal to the maximum number that can be stored in one lane of the arc.

4 Simulation Results

The algorithms presented above were programmed in C++, and tested on the network shown in Figure 4.1. The arc in the middle is 960 metres long, and 2 lanes wide. The remaining 4 arcs are each 480 metres long and consist of one lane. The effective vehicle length was 7.5 m, and the reaction time was 1.0 s. The same maximum speed was used for all 5 arcs, and the network was tested at three different speeds: 54, 81, and 108 km/hr (these speeds were chosen because they are multiples of λ/τ). The demand consisted of a flow from node 1 to node 3, and from node 2 to node 4. Thus, all vehicles entering the middle arc by the right lane were obliged to exit by the left lane, and vice-versa. In all tests, the two demand flows were equal to the maximum one-lane flow at the maximum speed. The maximum flows corresponding to the speeds given above (in increasing order of speed) are 2400, 2700, and 2880 vehicles/hour. Due to the 'weaving' of the two demand flows on the middle arc, the average flow on this arc tends to be significantly less than the total demand from its two upstream arcs. The simulated time of each test was 60 minutes.

The same network was tested using the AIMSUN2 micro-simulator. The parameters were chosen in such a way as to be comparable with those given above. Since AIMSUN2 uses Gipps' car-following model (Barceló *et al*, 1994), which uses a temporal safety factor of 1.5 (Gipps, 1981), the reaction time used in AIMSUN2 was 0.67s. The desired speeds of the vehicles were set equal to the maximum speeds on the arcs, and the effective vehicle lengths were as used above. The lengths of the zones were determined as follows. Zone 2 – the position at which a vehicle begins to look for a possible lane change – was set equal to the length of the middle arc, as this is consistent with the assumptions of the model presented here. What length to use for zone 1 – the position at which a driver begins to look more aggressively for a lane change opportunity – was less obvious. It was decided to set this value to roughly one half the length of zone 2. The number was not chosen precisely; rather, a value was selected for each of the speeds in such a way that the position at which vehicles were observed (in the animation) to begin changing lanes was roughly the same.

Time-series plots of flow and density are shown in Figures 4.2 and 4.3, respectively, for the 81 km/hr test. Qualitatively, these plots are indicative of the general patterns observed for the two models. The results obtained with both models tend to stabilize after a short period of time, but those obtained with AIMSUN2 tend to have a slightly higher variance about this value. A second test was run with the same demands, but with the outflow of the middle arc controlled by a signal. The signal ran on a 60 sec cycle, with 30 sec green and red phases. During the red phase, no flow was allowed to exit. The time-series plots of flow and density for the 81 km/hr test are shown in Figures 4.4 and 4.5, respectively. In this case, the density plot obtained with AIMSUN2 yields a slightly lower variance than that obtained with the STQ-based model.

Figures 4.6 and 4.7 show the average flows and densities, respectively, obtained with each model for each of the 3 speeds for the two tests – without the signal, and with the signal, respectively. The data from the signalized test are denoted with an *s* following the model identifier, either *STQ* or *A2*, in the legends. For the first test, the average flows produced by the models are very similar, and appear to be invariant across the three speeds. The average densities for this test are also very closely matched for the two models, but there is a definite decreasing trend with speed. The second test shows a slightly increasing trend in flow with speed, which is again captured by both models. The average densities for this test do not show a definite trend with speed. A fairly significant discrepancy is found for this statistic between the two models for the 108 km/hr test.

Figure 4.1: Test Network

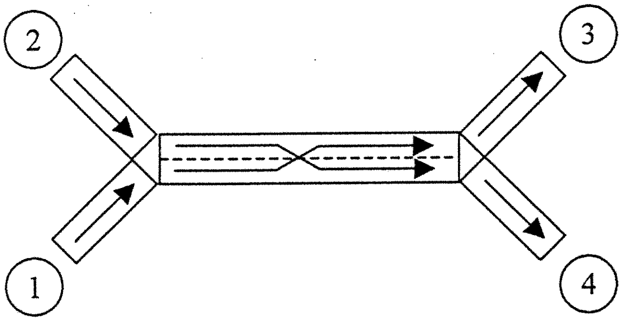


Figure 4.2: Flow vs. Time (non-signalized)

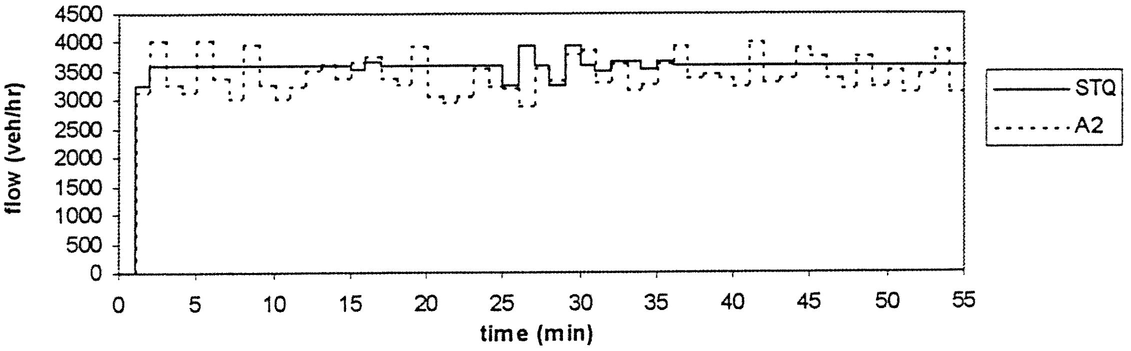


Figure 4.3: Density vs. Time (non-signalized)

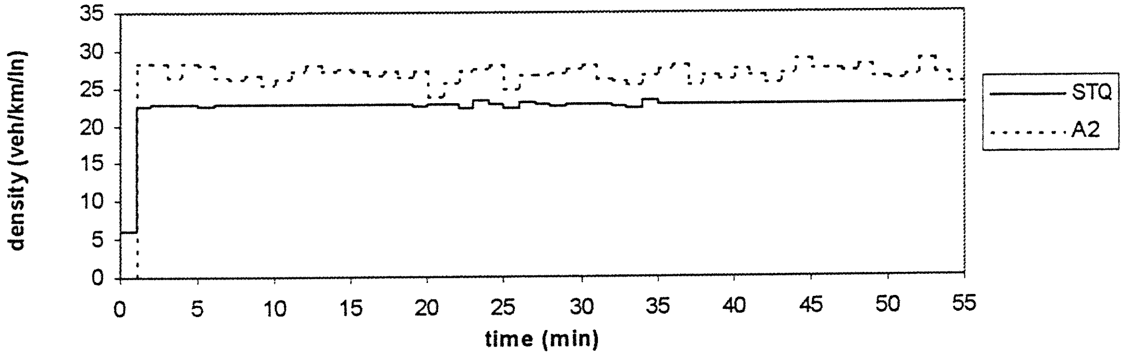


Figure 4.4: Flow vs. Time (signalized)

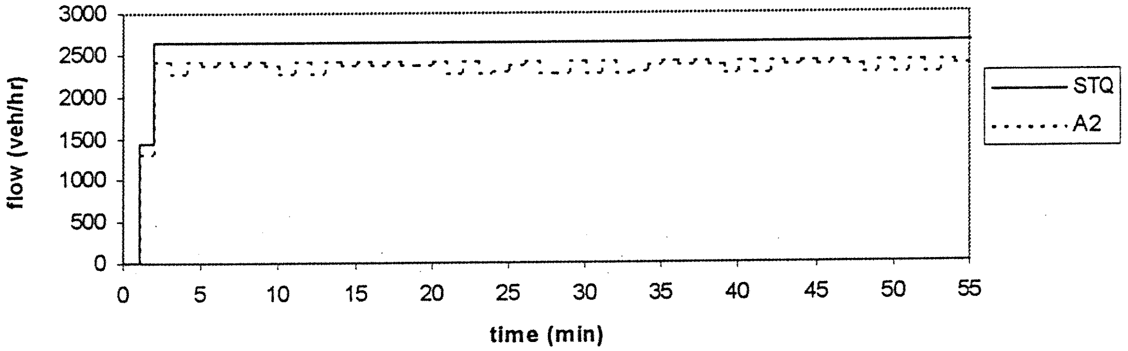


Figure 4.5: Density vs. Time (signalized)

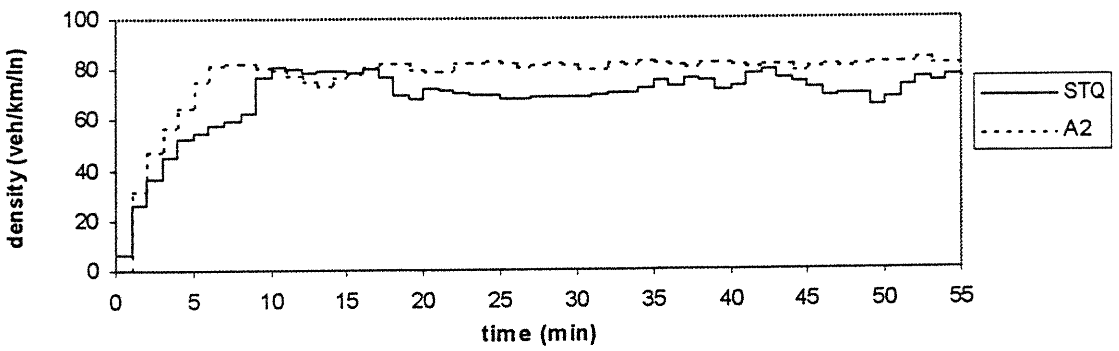


Figure 4.6: Average Flow vs. Maximum Speed

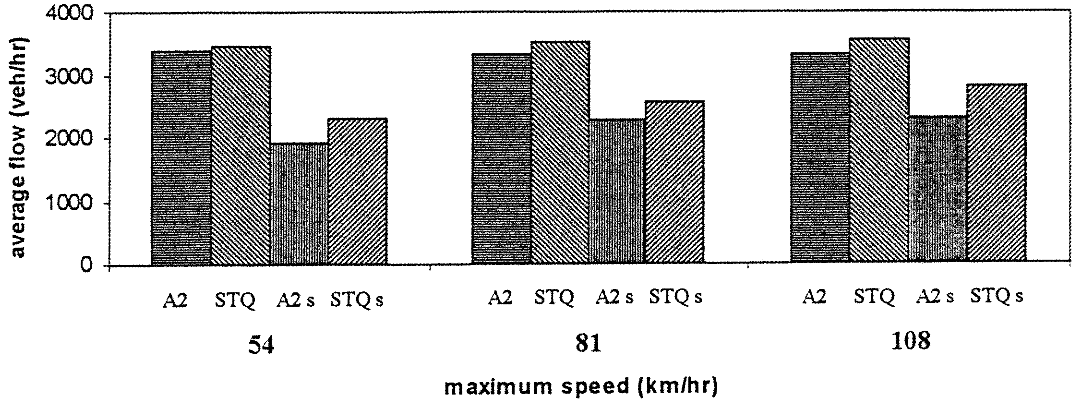
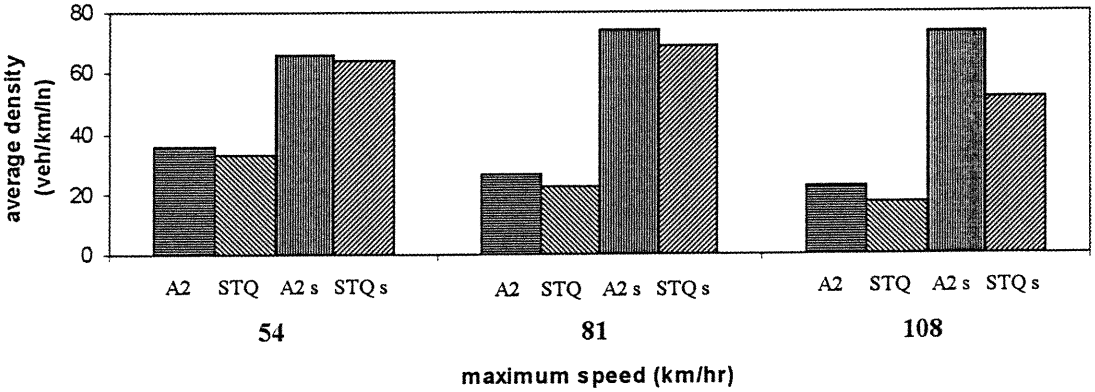


Figure 4.7: Average Density vs. Maximum Speed



5 Conclusions

A simplified model of longitudinal traffic dynamics recently proposed (Mahut, 2000) was extended to a general multi-lane arc model. The principal concern was to properly capture the effects of vehicle *conflicts* – the diverging, merging, and crossing of vehicle trajectories – within the arc, without sacrificing the arc-based nature of the original solution algorithm. Another objective was that the model be consistent with a precise description of the vehicle trajectories on a cell-by-cell basis. This allows the model assumptions to be verified using an animation approach, much like the graphical animation of micro-simulators is used to verify the inputs and assumptions of such models. A small animation example was presented, along with the associated calculations. The vehicle trajectories seemed quite plausible.

Tests were executed on a larger scale network by programming the algorithms in C++. The network consisted of 5 arcs, with two origin-destination flows, and was simulated over a period of 60 minutes. Adding a traffic signal to the network produced a second set of results. These two scenarios were also tested using the AIMSUN2 micro-simulator.

The outputs were average flows and densities for one-minute intervals. For both of these statistics, both models stabilized to fairly stationary conditions after a short period. The STQ-based outputs generally had lower variance than AIMSUN2 in the time-series plots, though not always. Both scenarios were run at three different speeds with both models. The average densities and flows were observed. The flows obtained with the STQ-based model were consistently higher than those obtained with AIMSUN2, while the densities obtained were consistently lower. In general, these statistics were found to match quite well between the two models. A significant discrepancy was found between the average densities obtained with the two models for the highest speed (108 km/hr) test, when the traffic signal was present.

The execution time of AIMSUN2 was roughly 10 times longer than that required for the STQ-based model in the first test. This ratio went up to roughly 15 for the second test. The execution times were virtually the same for the STQ-based model, at a little under 1 sec. The increased ratio was entirely due to a longer execution time for AIMSUN2, which was attributable to the higher densities (and hence higher travel times) that occur in the second test. This ratio should increase not only with increasing arc densities, but with increasing arc lengths as well. It should be noted that AIMSUN2 is a very sophisticated micro-simulator, and no doubt some of the execution time was associated with model features that the code developed here simply did not have. The calculation of vehicle emissions (pollutants) is but one example. The models were tested on a Pentium II 450 MHz computer under the Windows 98 operating system.

Overall, the performance of the model proposed here appears to be very promising. The solution algorithm is arc-based, i.e., the computational effort per vehicle per arc is independent of the vehicle's travel time on the arc. This feature results in particularly fast execution times, which are of interest for real-time applications such as route guidance and adaptive network control. The results of the tests matched quite well in general between this model and the AIMSUN2 micro-simulator. It is significant that the results obtained with the STQ-based model were in general more 'fluid' (i.e., higher flows and lower densities) than those obtained with AIMSUN2. This difference may be explained by the fact that the STQ does not take into consideration the limits on acceleration and deceleration that are respected by the micro-simulation model. Although the simulation tests presented were by no means trivial, the testing of the model is still in the early stages. Further testing with more complex patterns of vehicle interaction within the arc would be a desirable next step. The model also requires the addition of a sufficiently general node model, which uses the event-based solution approach, in order to model large-scale networks of practical interest.

References

Barceló, J. (2000). Personal communication.

Barceló, J., J. L. Ferrer and R. Grau (1994). AIMSUN2 and the GETRAM Simulation Environment. *Internal Report*, Departamento de Estadística e Investigación Operativa. Universitat Politècnica de Catalunya. See also http://www.tss-bcn.com.

Ben Akiva, M., H. Koutsopoulis, and T. Toledo (2000). MITSIMLab: Recent Developments & Applications. Presented at the INFORMS Spring 2000 Conference, Salt Lake City, May 7-10, 2000. See also http://www.its.mit.edu.

Gipps, P. G., (1986). MULTSIM: A Model for Simulating Vehicular Traffic on Multi-lane Arterial Roads. *Mathematics and Computers in Simulation* 28, pp. 291-295.

Hugosson, B., and H. Andersson (1999). Evaluation of AIMSUN2 in Stockholm. The "SMARTEST" Project. http://www.its.leeds.ac.uk/smartest.

Leutzbach, W. (1988) *Introduction to the Theory of Traffic Flow*, Springer-Verlag Berlin, Heidelberg.

Lindley, D. V. (1952). The Theory of Queues with a Single Server. *Proceedings of the Cambridge Philosophical Society*, 48:277-89.

Mahut, M. (1999b). Behavioural Car Following Models. Report CRT-99-31. Centre for Research on Transportation. University of Montreal. Montreal, Canada.

Mahut, M. (2000). From Traffic Flow To Queueing Theory. To be presented at the 8th Annual Meeting of Euro Working Group on Tranportation. Rome, Itatly, September 11-14, 2000.

May. A. D. (1990) *Traffic Flow Fundamentals*. Prentice-Hall, Englewood Cliffs, NJ.

Rickert, M., K. Nagel, M. Schreckenberg, and A. Latour (1997). Two lane traffic simulations using cellular automata. *Physica A*, 234:687.

Traffic Science. D.C. Gazis, (ed). Wiley, New York, 1973.

Transportation Research Board (1998). Special Report 209: Highway Capacity Manual, 3rd ed. National Research Council.

Van Aerde, M. (1999) INTEGRATION© Release 2.20 for Windows: User's Guide.

A Discrete Flow Model For Dynamic Network Loading

Michael Mahut, University of Montreal, Centre for Research on Transportation
Montreal, Canada

Abstract

This paper presents a node model for use in conjunction with the discrete-flow discrete-event arc model proposed by Mahut (2000b), which is based on a simplified model of traffic dynamics proposed by the same author (Mahut, 2000a). A discrete-flow node problem is presented as the determination of the next vehicle to enter a node given the necessary information concerning the first vehicle on each of its upstream lanes. A solution algorithm is proposed, and is embedded in an event-based network-loading procedure. This procedure ensures that the algorithm is executed at any time the relevant inputs should change in the course of the simulation. Heuristic rules for choosing arrival and departure lanes in conjunction with the arc model are suggested. The model is tested on a small network under time-varying supply and demand conditions. The output is found to compare quite well with that obtained using the INTEGRATION micro-simulator. The execution time of the proposed model is on the order of one two-hundredth of that required by INTEGRATION.

Acknowledgements

This work was supported in part by a post-graduate scholarship of the Natural Sciences and Engineering Research Council of Canada (NSERC). The author would like to thank Professor Michael Florian, as well as Shane Velan and Karim Er-Rafia, of the University of Montreal, Canada, and Vittorio Astarita of the Universita degli Studi della Calabria, Italy, for many valuable comments and discussions during the preparation of this work.

1 Introduction

This paper presents a general node model for use in conjunction with a discrete-flow discrete-event arc model recently proposed by Mahut (2000b). Due to the discrete-event nature of the model, the available information concerning other discrete-flow approaches to modeling traffic – such as the many micro-simulation traffic models (Gabard, 1991; Rothery, 1999) – is generally inapplicable as these models use a discrete-time approach to solve the underlying traffic model. The output required of a discrete-event procedure is quite different than that of a discrete-time approach, as discussed below. The model takes as inputs the next downstream arc and lane of the first vehicle on each of the upstream lanes of a node, as well as the earliest time at which each of these vehicles may exit its current arc. Pair-wise precedence relationships are established between these vehicles using the familiar concept of *gap-acceptance* (Velan, 1997). The output of the model is the identity of the next vehicle to enter the node – referred to as the *node leader* – and the time at which this event will occur. These results remain valid as long as the inputs to the model do not change between the time this event is predicted, and the time it takes place. If a change to these inputs should occur during this time interval, these results are re-calculated, and a new node leader and event time are established. The output of a discrete-time model is, by contrast, the velocity and/or acceleration of the first vehicles on each of the upstream lanes of a node. These results remain valid for a fixed interval of time, and cannot be re-calculated before this time interval has expired.

It is found that in some cases, the pair-wise precedence relationships determined by the gap acceptance logic do not directly imply a logical choice for the next vehicle to enter the node. A solution algorithm is proposed which avoids the unnecessary delays that can be implied by a naïve interpretation of the pair-wise relationships. This algorithm is embedded in a discrete-event network-loading procedure, which re-evaluates the selection of the node leader any time the relevant inputs should change for any node. The procedure is analogous to an event-based approach for modeling a network of capacitated queues, where the event of a vehicle entering service at one server may simultaneously affect the states of two other servers: the server upstream of the queue from which the client has just departed (should this queue be at its maximum storage capacity), and the server downstream of the queue the client is about to enter (should this queue be empty).

Heuristics are then proposed for the problem of choosing arrival and departure lanes for the arc model proposed by Mahut (2000b), as a function of the current traffic conditions on the arc. The choice of arrival lane is based on the use of strict lane permissions. Given the upstream arc from which a vehicle is arriving to a downstream arc, there is a specific set of lanes that are potential arrival lanes for this vehicle on the downstream arc. The choice of departure lane from the upstream arc is based on a less strict notion of *target* lanes. These lanes are defined for an upstream arc given the downstream arc to which a vehicle is destined. The departure lane choice generally respects the target lanes, unless these lanes are inaccessible due to excessive traffic congestion. Two different rules for departure lane choice are proposed. The complete model is then tested on a small network under time-varying demand and supply conditions. The results obtained using the two different heuristics are presented, along with results obtained with the INTEGRATION micro-simulator (Van Aerde, 1999). The results are found to compare quite well in general. Some notable discrepancies are found to occur under specific conditions, but it is not clear that these differences imply a deficiency in the model. The execution time of the proposed model is roughly one two-hundredth of that required by INTEGRATION.

2 The Node Model

This section presents a node model that is combined with the arc model referenced above. Essentially, the problem that must be solved at a node is the determination of precedence among conflicting vehicles in a realistic way. This definition applies to any discrete-flow traffic model, regardless of whether it uses car-following (CF), cellular automata (CA), or the space-time queue (STQ) as the mechanism of moving vehicles within an arc. All such models also share a similar network representation. Since each discrete unit of flow represents one vehicle, each arc is defined by an integer number of lanes. What differs among the different approaches is the information that is available to describe the state of each vehicle (position, speed, acceleration, etc...), and the nature of the solution algorithm used to solve the flow propagation model. With respect to the latter, the space-time queue is fundamentally different from both CF and CA models, because it is solved using a *discrete-event*, rather than a *discrete-time* approach. A discrete-event algorithm is based on the notion that specific causes have specific consequences at *predictable* moments in time. This approach is used for simulating queueing systems, where the service time of each client is drawn randomly from a distribution at the moment the client begins service. As a result, the moment when the next client may begin service can be predicted at this time. A basic property of such models is that time is real-valued. Discrete-time algorithms, on the other hand, are based on the notion that the underlying system evolves in a continuous – rather than discrete – way, and provide a description of the system at regular intervals in time. This approach is used for CF and CA traffic models, where the time interval is typically on the order of one second or less. As already mentioned above, node models intended for use with a discrete-time traffic model are fundamentally different than those required by a discrete-event approach.

2.1 A Procedure For Determining Precedence Among Conflicting Vehicles

It should be understood that a node model is primarily concerned with the (spatial-temporal) conflicts of vehicle trajectories at an intersection, and thus only requires information pertaining to the first vehicle on each upstream lane. Each such vehicle must have the following information associated with it: the downstream arc for which it is destined, referred to as the "supply arc" of the vehicle, the lane by which it will arrive to this arc (the arrival lane), and the lane by which it will leave this arc (the departure lane). The arc model then uses this information to determine the earliest time that the vehicle may execute the movement from its current arc to its supply arc, which is referred to as the *demand time* of the vehicle – see (Mahut, 2000b). These four data define the state of the vehicle. Given the supply arcs and arrival lanes of the first vehicles on all of the lanes upstream of a node, the spatial conflicts among these vehicles can be identified. Spatial conflict implies that the trajectories of two vehicles – as given by the pairs of lanes (upstream, downstream) associated with their movements at a node – either merge, diverge, or cross at the node. The condition of spatial conflict between the first vehicles on two upstream lanes (u_1, u_2) will be denoted as $u_1 \sim u_2$.

A node model must also enforce any control measures that are in place at the intersection, such as traffic signals. The state of a node, at any moment in time, will include the state of any control mechanism and the state of the first vehicle on each of its upstream lanes. The node state also includes the identity of the next vehicle to execute a movement from its current (upstream) arc to its supply (downstream) arc. This vehicle will be referred to as the *node leader*. The (discrete-event) node problem is thus defined as identifying the next node leader as a function of the node state. A desirable property of any node model that solves this problem is that it yields a unique solution. Given any node state, there must only be one possible node leader. Of course, it is also important that the model be plausible, i.e., *realistic*. Once the node leader has been chosen in this way, the demand time of this vehicle yields the time at which it will enter the node, i.e., the intersection.

Micro-simulation models of traffic resolve conflicts between vehicle trajectories at nodes in essentially the same way as conflicts due to lane changing, by using gap-acceptance models (Velan, 1997). Only the type of information used to define the state of a vehicle restricts the type of gap acceptance model that can be applied. Knowledge of the upstream and downstream lanes of the movement associated with each vehicle, along with the corresponding demand time, is sufficient for the general type of gap-acceptance model used in CF-based simulation models, many of which can thus be directly applied here (*ibid.*). As in the case of lane changing, the purpose of such a model is to determine precedence between two conflicting vehicles given their states. This in turn requires that one of the vehicles be given priority over the other. The arc pair (upstream arc, downstream arc) associated with a vehicle's movement at the node is generally used to determine priority. The precedence relationship that results between the first vehicles on two upstream lanes (u_1, u_2) will be denoted by $\rightarrow$, i.e., $u_1 \rightarrow u_2$ implies that the first vehicle on lane u_1 must follow the first vehicle on lane u_2.

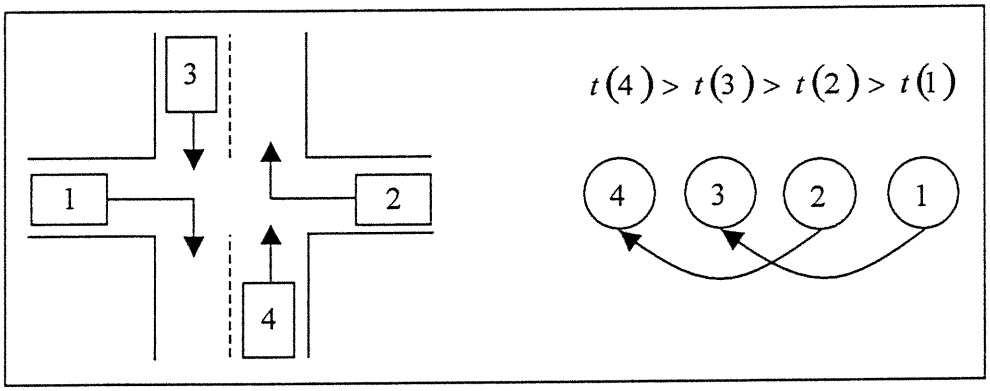
The general form of such a model is given below, as algorithm $\mathit{Resolve}(u_1, u_2)$, where u_1 and u_2 denote two upstream lanes of a node. The demand time of the first vehicle on lane u is denoted by $t(u)$. An exogenous function $gap(u_1, u_2)$ is called to calculate the required gap. A trivial $gap(u_1, u_2)$ function would simply return zero in all cases, resulting in a FIFO (first-in-first-out) ordering of all mutually conflicting vehicles. A set of mutually conflicting vehicles is defined here as a set of vehicles such that each pair is in spatial conflict.

```
$\mathit{Resolve}(u_1, u_2)$
    $if\langle u_1 \sim u_2 \rangle$
        $if\langle t(u_1) - t(u_2) \geq gap(u_1, u_2) \rangle$
            $u_1 \rightarrow u_2$
        $else$
            $u_2 \rightarrow u_1$
        $endif$
    $endif$
$end$
```

The remaining discussion will assume that an exogenous function $gap(u_1, u_2)$ is used to solve the gap acceptance problem, i.e., it will be assumed that a model is available, which takes as inputs the states of two vehicles, and returns as an output which vehicle will precede the other. The vehicle that is thus determined to precede the other will be referred to as the *leader*, while the succeeding vehicle will be referred to as the *follower*. The discussion will now focus on the problem of how to solve the node model, given the results of the gap acceptance model for each pair of vehicles.

This problem is trivial if there is only one pair of conflicting vehicles, or if there is more than one such pair, but no vehicle belongs to more than one of them. An example of such a situation is given in Figure 2.1. This figure contains a graph representation of the results obtained by calling $Resolve(u_1, u_2)$ successively for all pairs (u_1, u_2), as indicated by the arcs. The nodes of the graph represent the vehicles in question, or equivalently, the upstream lanes of the node. For convenience, the nodes are drawn from left to right in decreasing order of $t(u)$. A simple rule for determining the node leader in this case is to traverse the list from right to left, and to select the first node that does not have an outgoing arc, which would be node 3. Thus, vehicle 3 is the node leader for this node state.

Figure 2.1: First Node Conflict Example



A more interesting situation is depicted in Figure 2.2. There are two pairs of conflicting vehicles, but vehicle 1 belongs to both pairs. The results obtained from $Resolve(u_1, u_2)$ are again shown to the right of the diagram. In this case, using the rule above would yield vehicle 4 as the node leader. This may not be the most reasonable solution, under the general assumption that each vehicle attempts to proceed forward as quickly as possible. This solution requires vehicle 3 to be delayed unnecessarily, since vehicle 1 must wait for vehicle 4. It seems reasonable then that vehicle 3 should be the next node leader, by virtue of $t(3) < t(4)$, which requires invalidating the result $3 \rightarrow 1$. One way to modify the above rule might be to consider only those arcs in the graph that point to the left. This would imply that if a vehicle has already been discarded as a possible candidate, it may no longer prevent other vehicles from being chosen. Applying this rule in Figure 2.2 indeed yields vehicle 3 as the node leader. This modified rule does not however provide the most reasonable solution for the example presented in Figure 2.3. Applying the modified rule, the node leader is vehicle 2. In this case, however, since vehicles 3 and 4 must wait for vehicle 2, vehicle 1 is delayed unnecessarily. The most reasonable node leader is thus vehicle 1.

The proposed solution is reasoned as follows. If a vehicle is the follower in even one conflict, *all* of the conflicts in which it is the leader are ignored. Of course, before such a decision can be made, it must be ascertained that the conflict in which this vehicle is the follower is in fact valid, i.e., that the leader in *this* conflict is not a follower in some other conflict, and so on... The idea is to first identify all vehicles that are not followers in any conflicts, and proceed from there. The procedure is given in algorithm *ResolveNode*, below. This algorithm uses two lists to store the vehicles: a *leader* list, and a *follower* list.

Figure 2.2: Second Node Conflict Example

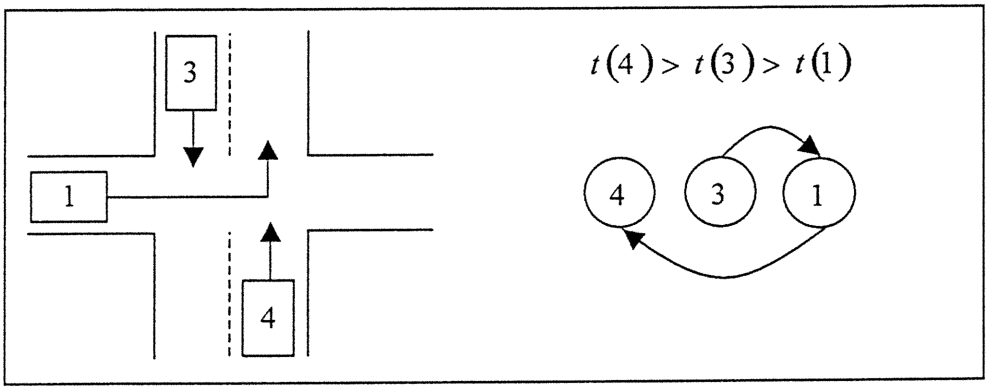
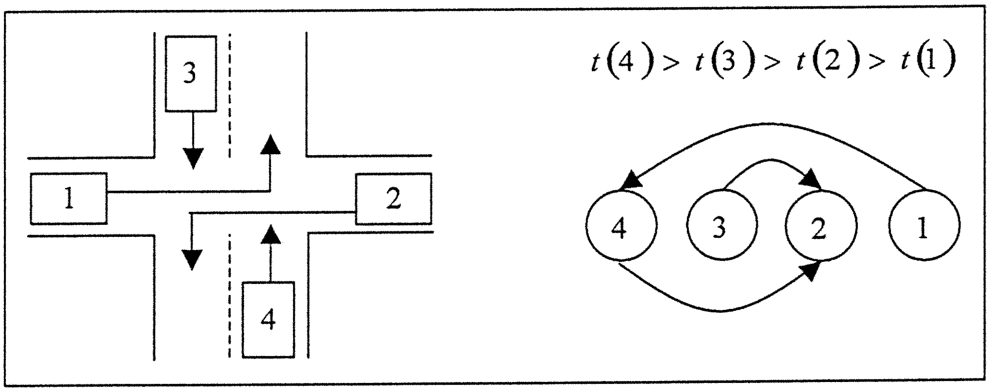
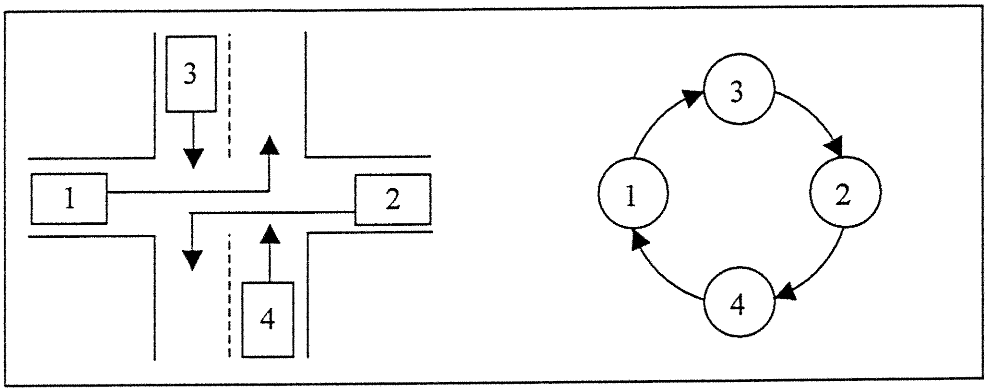


Figure 2.3: Third Node Conflict Example



One issue that has not been discussed is the possibility of cycles in the graph. If cycles are possible, the algorithm may not move any vehicles to the leader list in step 2. In this case, step 3 will fail to remove any vehicles from the follower list, and the algorithm will cycle indefinitely. One way to avoid this problem is to explicitly identify any cycles in the graph if no vehicle is found in step 2. The vehicle that is moved from the follower list to the leader list may then be chosen as the one with the lowest demand time among all those belonging to any cycle. Another approach would be to impose certain constraints on the $gap(u_1, u_2)$ procedure that ensure that no cycles can occur. Figure 2.4 shows another possible set of pair-wise precedence relationships for the node state of Figure 2.3, but the nodes are drawn in a way that reflects their geometric arrangement rather than their temporal relationships. If it is assumed that all cycles must form a loop of this kind when the nodes are drawn in this way, it can be ensured that no cycles will form as long as, for any given vehicle, the gap with respect to another vehicle in the clockwise direction is always strictly less than the gap required in the counter-clockwise direction, or vice-versa. In general, it appears that adding constraints of this kind could easily render the $gap(u_1, u_2)$ procedure unrealistic, and is not recommended.

Figure 2.4: Fourth Node Conflict Example



```
ResolveNode

step 1: initialization
    Identify all of the vehicles for which the requested movement at the intersection is currently permitted by the traffic signal. Initialize the follower list with these vehicles. The leader list is empty.

step 2:
    Identify all the vehicles in the follower list that are not followers in any conflicts with other members of the follower list. Add these vehicles to the leader list.

step 3:
    Identify all the vehicles remaining in the follower list that have a conflict with any of the vehicles just added to the leader list in step 2. Remove these vehicles from the follower list and discard them.

step 4:
    If the follower list is not empty, go back to step 2.

step 5:
    Find the vehicle in the leader list with the earliest demand time. This vehicle is the node leader, and the demand time of this vehicle is the next event time of the node.
```

The worst-case complexity of *ResolveNode* can be analyzed as follows. Letting M denote the number of lanes upstream of a given node, step 1 clearly requires M operations to verify the signal permissions and create the original follower list. Step 2 requires that each pair of vehicles remaining in the follower list be evaluated. Denoting the number of vehicles in the follower list at the beginning of iteration i by M_i, the number of operations in step 2 of iteration $\underline{i}$ – denoted by K_2^i – is given by:

$$(2.1)\qquad K_2^i = M_i(M_i - 1)/2\,.$$

It should be noted that in step 3, it is sufficient to compare the vehicles remaining in the follower list only with those that have just been moved to leader list in the previous step 2. Denoting by l_i the number of vehicles moved to the leader list in step 2 of iteration i, the number of operations in step 3 – denoted by K_3^i – is given by:

$$(2.2)\qquad K_3^i = (M_i - l_i)l_i$$

Denoting by f_i the number of vehicles discarded from the follower list in step 3 of iteration i, M_{i+1} may be defined recursively as $M_{i+1} = M_i - l_i - f_i$. From the discussion above, it should be evident that $1 \le l_i \le M_i$, for all i. The lower bound for the value of f_i is zero. The number of operations in step 2 (2.1) may thus be written as:

$$(2.3)\qquad K_2^{i+1} = (M_i - l_i - f_i)(M_i - l_i - f_i - 1)/2$$

Considering equations (2.2) and (2.3), it is apparent that the values of K_2^i and K_3^i are both maximized when l_i and f_i are minimized, i.e., $l_i = 1$ and $f_i = 0$, $\forall i$. The worst case thus requires exactly M iterations, and the maximum number of operations is given by:

$$(2.4)\qquad M + \frac{M(M-1)}{2} + \sum_{i=1}^{M-1}\left[(M-i)i + \frac{(M-i)(M-i-1)}{2}\right].$$

With some manipulation, this reduces to $(M^3 - 2M)/3$. The complexity of the worst-case scenario is thus in the order of M^3.

2.2 The Node Constraint

The demand time of a vehicle is constrained not only by the rules defining the arc model, but also by the presence of conflicting vehicles in the intersection. In general, these are vehicles with which the vehicle in question is in conflict, and to which it was obliged to yield the right of way. It is also possible that a vehicle in the intersection entered it before the vehicle in question became first on its lane, and thus the latter was never in a position to precede the former. Thus, a constraint on the demand time of a vehicle beyond those defining the arc model must be applied, to ensure a minimal temporal separation between conflicting vehicles as they pass through an intersection. In micro-simulation models of traffic, the minimum separation between conflicting vehicles at a node is primarily a function of the turning speed, which is associated with the lane pair (upstream lane, downstream lane) of the preceding vehicle (Barceló *et al*, 1994). The turning speed is literally the speed at which a vehicle is assumed to have passed through the intersection, and is calculated as a function of very detailed geometric information (*ibid.*). Clearly, if a vehicle makes a very sharp turn at an intersection, there is an upper bound on the speed at which it can safely execute this turn. This speed serves to reduce the maximum flow per lane that may be observed for a given turning – the lower the speed, the lower the maximum flow. The turning speeds are assumed to be known, and are denoted by $V(u,s)$, where u once again denotes an upstream lane, and s denotes a downstream lane. Denoting the demand time of the last vehicle to have passed through the intersection from lane u to lane s by $t^{last}(u,s)$, and the arrival lane of the first vehicle on lane u as $m^A(u)$, the following *node constraint* is defined on demand time:

$$(2.5) \quad t(u_1) \geq t^{last}(u_2,s) + \left(\tau + \frac{\lambda}{V(u_2,s)}\right) \qquad \forall (u_2,s) \mid (u_2,s) \sim (u_1, m^A(u))$$

3 A discrete-event network-loading algorithm

In this section, the model presented above is embedded in a discrete-event network-loading algorithm. The algorithm consists of a single event procedure, much like the underlying procedure of a discrete-event algorithm for modeling a network of capacitated queues. This procedure for this event – referred to as the *node event* – is executed at the moment a node leader enters an intersection, i.e., at the demand time of the node leader. This procedure executes the ***ResolveNode*** algorithm presented above in order to determine a new node leader at each node event. Before selecting a new node leader, it must be determined if there is a new first vehicle on the upstream lane of the current node leader, and if so, determining its state. This requires the selection of the supply arc, arrival lane, and departure lane of the new vehicle. The arc model then returns the demand time, which is further constrained by the node constraint (2.5) above. By applying (2.5) to all vehicles in conflict with the current node leader, the updating of the node state is completed. With the node state updated, ***ResolveNode*** can be applied to determine the new node leader. If a node leader is found, a procedure called *Schedule()* is called to post the next event in the central event list. For a description of the basic process of discrete-event simulation, the reader may consult any general text on simulation such as (Law and Kelton, 1991).

A node event at a particular node, at which a vehicle moves from lane u to lane s, can further affect the node states of two other nodes: the upstream node of lane u, and the downstream node of lane s. The former case occurs when the lane u is *inactive* (Mahut, 2000b). This means that the calculation of the demand time for a vehicle to enter this arc currently depends on the departure time of the next vehicle from lane u, and thus the calculation has been postponed until this event. In physical terms, this means that the number of vehicles currently on the arc destined for lane u is equal to the maximum number of vehicles that can be stored on one lane. As a result, the time when the next vehicle may enter this lane depends on the departure time of the vehicle that is currently first on this lane, which is not yet known. The latter case occurs when the node leader is first for the departure lane of its supply arc. In other words, the departure lane is empty, and there is now a new first vehicle that may impact the determination of the node leader at its downstream node (however unlikely such an impact may be). In the case where all of the upstream lanes of the downstream node of lane s are currently empty, this case serves to initialize the recursive node-event process at this node. The entire process is initialized by the use of special origin nodes that generate vehicles and inject them into the network.

A schematic description of this procedure is as algorithm $NodeEvent(node)$, below. The variables u and s denote the upstream and downstream lanes, respectively, of the node leader. The upstream and downstream nodes of a lane (m) are denoted by $head(m)$ and $tail(m)$, respectively. A complete simulation model would include other recursive event procedures as well. Other events would be used for changing the state of traffic signals and collecting time-dependent statistics, for example. The event procedure for changing the state of a signal would be required to cancel the current node event, update the signal state, call $\boldsymbol{ResolveNode}$, and if applicable, schedule the next node event.

```
NodeEvent(node)

    if (lane u is not empty)
        Determine the state of the next vehicle on lane u:
        select the supply arc
        select the arrival and departure lanes
        determine the demand time
    endif

    Update the state of node:
        update the demand times of all vehicles in conflict with the node
        leader of node

    ResolveNode(node)

    if (a node leader exists for node)
        Schedule (NodeEvent(node))
    endif

    if (lane s is empty)
        Determine the state of the node leader for head(s)
            select the supply arc
            select the arrival and departure lanes
            determine the demand time
        ResolveNode(head(s))

        if (a node leader exists for head(s))
            Schedule (NodeEvent(head(s)))
        endif
    endif

    if (lane u is full)
        for all lanes u' of tail(u) such that $\langle arc(m^A(u')) = arc(u) \rangle$ do:
            determine the demand time $t(u')$

        ResolveNode(tail(u))

        if (a node leader exists for tail(u))
            Schedule (NodeEvent(tail(u)))
        endif
    endif
end
```

The worst-case computational complexity of this algorithm can be determined in a straightforward way. Rather than developing the analysis in the context of a given node, with known parameters (e.g. number of upstream lanes), it will be assumed that the maximum values of such parameters for the entire network are known. In particular, the following parameters are defined:

M_0 = maximum number of lanes for any arc
B_0 = maximum number of upstream arcs for any node
N_0 = total number of nodes in the network
P_0 = maximum number of nodes per path.
Q_0 = total number of vehicles defined by the demand flows for the network

It is assumed that a data structure is used to store the paths being used, so that selecting the *supply arc* requires a constant amount of calculation. It is further assumed that whatever rules are used to select the arrival and departure lanes can be combined with the determination of the demand time without changing the order of the complexity of this calculation. This is consistent with the lane-choice heuristics suggested in the next section. A suitably modified version of algorithm $TDS(n)$, as proposed by Mahut (2000b), could then be used to determine the arrival and departure lanes, as well as the demand time, with a computational complexity in the order of M_0^5.

The maximum number of upstream lanes for any node is thus $B_0 M_0$. Thus, the procedure of updating the demand time of the first vehicle on each upstream lane of a node has a complexity in the order of $B_0 M_0 M_0^5 = B_0 M_0^6$. Following the discussion presented above, the worst-case complexity of the *ResolveNode* procedure for *any* node is in the order of M_0^3. It is assumed that the data structure used to store the node events is a sorted two-way linked list. This is the data structure used in the model implementation discussed in the next section. Although using a *heap* structure is much more efficient for inserting events that are not currently posted, a linked list offers some advantages that can be exploited in this type of model. For example, it is not uncommon that a node event already posted must be updated when the state of a node changes. A heap requires that the event be first removed from the heap, and then re-inserted with the new event time. A linked list permits the position of the event in the list to be updated directly. When the new position is very close to the old position, which is often the case in this model, the linked list can execute this procedure more efficiently than a heap. Another advantage of a (two-way) linked list is that the events can be inserted from either end. This can be exploited whenever it is known that certain types of events tend to end up near the bottom of the list upon insertion. For example, the event associated with changing the state of a traffic signal is generally predicted much further in advance than a node event. As a result, it ultimately ends up near the bottom of the list. This can be exploited by always searching for the insertion point for such an event by starting at the bottom of the list. The worst-case complexity of the *Schedule()* procedure – which includes the updating of events already posted – is thus in the order of N_0, as there is never more than one node event (and one signal event) posted for each node of the network.

It can be seen that the same types of operations are repeated throughout the rest of the *NodeEvent()* procedure. The worst-case complexity of this procedure is thus in the order of $B_0 M_0^6 + N_0$. The worst-case complexity for the entire DNL procedure is thus in the order of $Q_0 P_0 \left(B_0 M_0^6 + N_0\right)$, as there is exactly one node event per vehicle for each node on its path.

4 A Simulation Example

The above procedure was implemented in C++, in conjunction with the STQ-based arc model, and executed on the network illustrated in Figure 4.1. The numerical inputs are also shown in the figure. The necessary inputs to the model consist of the demand flows per path, parameters to describe each arc. The arc parameters consist of the following: maximum speed V, maximum density K, the negative wave speed W – which is equal to λ/τ – and the flow capacity Q. The flow capacity parameter is used to calculate a default turning speed for all turnings destined for the associated arc, thus putting a maximum inflow constraint on the arc. This in turn yields a trapezoidal – rather than triangular – steady–state relationship between flow and density as shown in Figure 4.2. An optional input is the definition of turning capacities, i.e., the number of lanes associated with each turning for each arc pair (upstream arc, downstream arc) at each node. The default value of turning capacity is the minimum of the number of lanes on each of these two arcs. The default value was used for all turnings in the network tested here.

Lane Choice Heuristics

A procedure was developed to determine a set of lanes on each of the two arcs associated with each turning. The set of lanes on the downstream arc associated with a specific turning, i.e., with a specific upstream arc, defines the lanes that may be chosen as arrival lanes by vehicles arriving from this arc. The set of lanes on the upstream arc, similarly associated with a specific downstream arc, define the *target* departure lanes for this turning. These are the lanes from which it would be desired that the departure lane be chosen upon entering the upstream arc. The procedure chooses these two lane sets for each turning at a given node in such a way as to minimize the conflicts that result from vehicles wishing to execute these turnings at the same time. The definition of a set of upstream and downstream lanes associated with each turning is a typical input for micro-simulation traffic models (Barceló *et al*, 1994). The cardinality of each of these sets is equal to the associated turning capacity, resulting in a one-to-one correspondence between the upstream and downstream lanes associated with each turning. As a result, a default choice of downstream lane for each upstream lane, for each turning is defined.

The general rule used for selecting an arrival lane, given this information, is to select the nearest active lane to the default lane. If the default lane is inactive, the remaining lanes will first be enumerated to one side of the default lane, and then to the other, until an active lane is found. The rule for selecting the direction in which to look first for an active arrival lane then depends on whether the default lane lies to the right, to the left, or within the target departure lane set for the *next* turning, i.e., at the exit of the supply arc. If the default lane lies to the left of this set, the search begins by looking to the right, and vice-versa. If the default lane lies within the target departure lane set for the next turning, the search direction chosen to be the opposite of that chosen by the last vehicle to have confronted this decision on the same arc. In this way, vehicles will generally be spread out equally to the left and right of a blocked center lane, when both of these lanes provide access to the vehicles' next arcs.

Two different heuristics were implemented for the choice of departure lane. It should be noted that if a particular lane is inactive, not only must it not be selected as an arrival lane or departure lane, but also it must not be a member of the set {*arrival lane*, ..., *departure lane*}. If any member of this set is inactive, the arc model will return an infinite value for the demand time corresponding to this lane choice (Mahut, 2000b). Thus, given the choice of arrival lane, the nearest inactive lane to the left and right of this lane define a "corridor" within which the departure lane must be chosen. Given some lane as a datum, one lane is said to be *nearer* than another lane to the given lane, if the number of lanes that must be traversed in order to reach the former is less than that required to reach the latter. The possible departure lanes thus defined are referred to as the *candidate* departure lane set.

The first rule for choosing a departure lane is as follows:

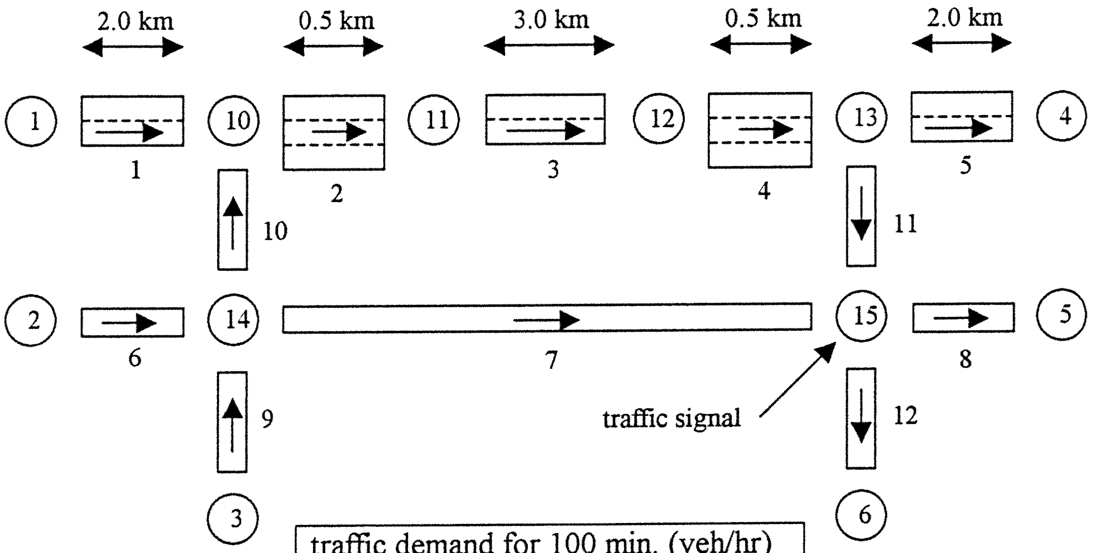
(H1) Choose the target departure lane nearest to the arrival lane, if it is active. If not, choose the candidate lane nearest to this lane.

The second rule for choosing a departure lane is as follows:

(H2) If the nearest active target departure lane is not equal to the arrival lane, choose the candidate lane that yields the lowest demand time. Otherwise, choose the arrival lane.

It may be noted that for both of these heuristics, if the arrival lane is a member of the target departure lane set, this lane will be chosen as the departure lane. As a result, the amount of lane changing is generally kept to a minimum.

Figure 4.1: Test Network

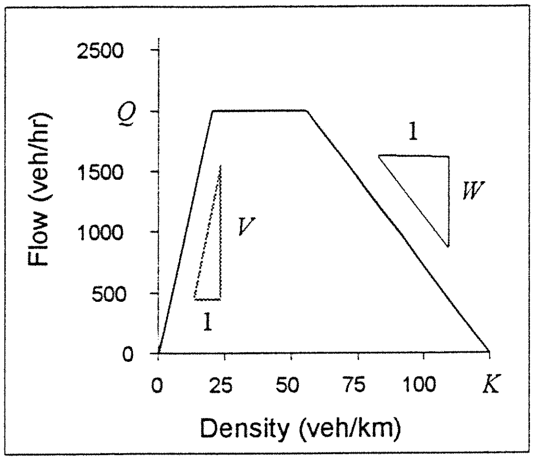


traffic demand for 100 min. (veh/hr)			
from \ to	4	5	6
1	2400	450	450
2	300*	300	
3	600*		

* demand ends after 20 minutes

arc parameters arc \	V km/hr	Q veh/hr/ln	W km/hr	K veh/km/ln
1, 2, 3, 4, 5	110	2000	24.7	110
10, 11	80	2000	29.3	110
6, 7, 8, 9, 12	60	1800	27.7	125

Figure 4.2: Three-Segment Flow-Density Relationship



Discussion of Results

The network depicted in Figure 4.1 represents a section of highway (arcs 1 to 5) with a one-lane service road (arcs 6 to 8), an on-ramp (arc 10) and an off ramp (arc 11). The only origin-destination pair with a non-zero demand and more than one path is (2,5), which was assigned to the service road. For a more in depth analysis of the temporal traffic conditions that occur in this particular example, the reader is referred to Astarita *et al* (2000).

The demand flows result in two bottlenecks occurring in the network. The first bottleneck forms at the exit of arc 2, because of the reduction in the number of lanes on the highway. The second bottleneck occurs at the exit of arc 11, due to a traffic signal. During the first 30 minutes of the simulation, arc 11 receives 25% of the green time at this intersection, and after 30 minutes it receives 67% of the green time. The signal operates on a 120 s cycle, which coincides with the interval used to measure the average flows and densities on the arcs. The network was simulated for 100 minutes. The same test was run using the INTEGRATION micro-simulator (Van Aerde, 1999). Two sets of results are shown for the STQ-based model, one corresponding to each of the departure lane heuristics mentioned above, and labeled accordingly. The third set of results, labeled *I*, was obtained using INTEGRATION. The outflows and densities observed for the four highway arcs upstream of the exit ramp are shown in Figures 4.3 and 4.4, respectively. The density plots are shown on a scale of zero to 80 vehicles per kilometre per lane, which is roughly two-thirds of the maximum density for these arcs. The differences between the STQ-based results and the INTEGRATION output are due mainly to differences in how the two models handle lane changing. As the most significant lane changing is observed on arcs 2 and 4, the observable differences are related directly or indirectly to the performance of these two arcs. Using the lane-choice heuristics described above, these are the only two arcs on which any lane changing occurs in the STQ-based model.

Considering arc 4, there is a general trend of decreasing inflow (which is equal to the outflow of arc 3) and outflow due to the combination of arc delays (arc conflicts), and node delays that are passed upstream from the congested off-ramp (arc 11) before the change of the traffic signal plan at t = 30. This outflow reduction is somewhat more pronounced in both sets of STQ-based results than in the INTEGRATION output. The former also show a certain instability in the observed flows during this period. It was found that each of the low flow measurements was due to a single vehicle being delayed for a particularly long time at node 12, due to inactive lanes on arc 4. As the arc conflicts – and hence the arc counts $\left(c^A(n,k)\right)$ – upon which the arc model calculates the demand times are based upon the given arrival and departure lanes, it is possible that more sophisticated lane-choice heuristics could yield more stable results. The effect of this reduction in flow is visible in two ways on the upstream arcs (3, 2, and 1). A similar discrepancy in flow, and an associated relative increase in density, is seen on these arcs as these delays are propagated upstream. This also explains why the increases in density occur slightly earlier in the STQ-based model than in INTEGRATION.

Another significant difference is observed on the density plots for arc 4. Although the three curves are qualitatively similar, those from the STQ-based model peak at a maximum value roughly half of that obtained with INTEGRATION. This discrepancy is due to the rules by which the location of a lane change is determined in the two models. INTEGRATION divides the arc into zones demarcated by a "soft wall" and a "hard wall" (Van Aerde, 1999), much like the zones used in the AIMSUN2 micro-simulator (Barceló *et al*, 1994). In preliminary tests of the STQ-based arc model against AIMSUN2, the parameters defining these zones were set in a way that would be roughly consistent with assumptions about lane-changing underlying the STQ-based model, and the resulting densities in general compared quite well (Mahut, 2000b). The parameters defining these zones cannot be modified in INTEGRATION, and thus it was not possible to affect this kind of calibration for the test presented here.

Lower outflows are also observed on arc 1 during the first 20 minutes of the simulation in the STQ-based results than are observed with INTEGRATION. In this case, the only factor is the lane-changing behaviour on arc 2, as there are no delays propagating from downstream of arc 2 at this time. This discrepancy in turn accounts for the higher densities observed on arc 1 during this period. It may be noted that the discrepancy between the STQ and INTEGRATION results in inflow to arcs where lane changing occurs is independent of the discrepancy in density on these arcs. The density on arc 2 obtained using *H2* was in fact higher than that obtained using INTEGRATION, although both *H1* and *H2* had lower inflows as mentioned above.

Another notable difference is the outflow of arc 4 after the signal-timing plan has changed at t = 30. All three curves arrive at roughly the same value around t = 38, but the INTEGRATION curve steadily decreases over the next 30 minutes while the STQ-based data maintain a virtually constant mean over this period. Although the difference is

Figure 4.3: Outflow vs. Time

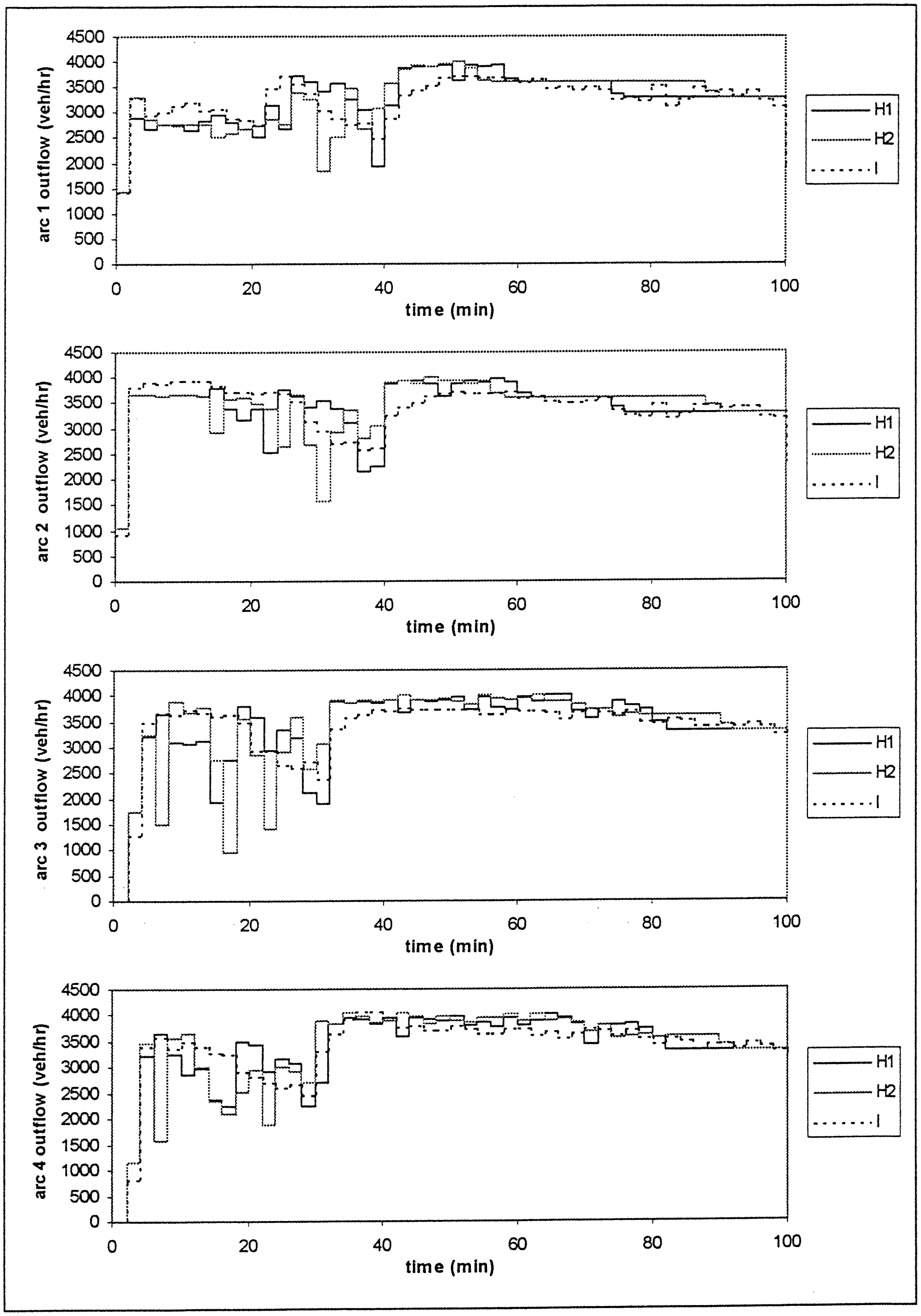
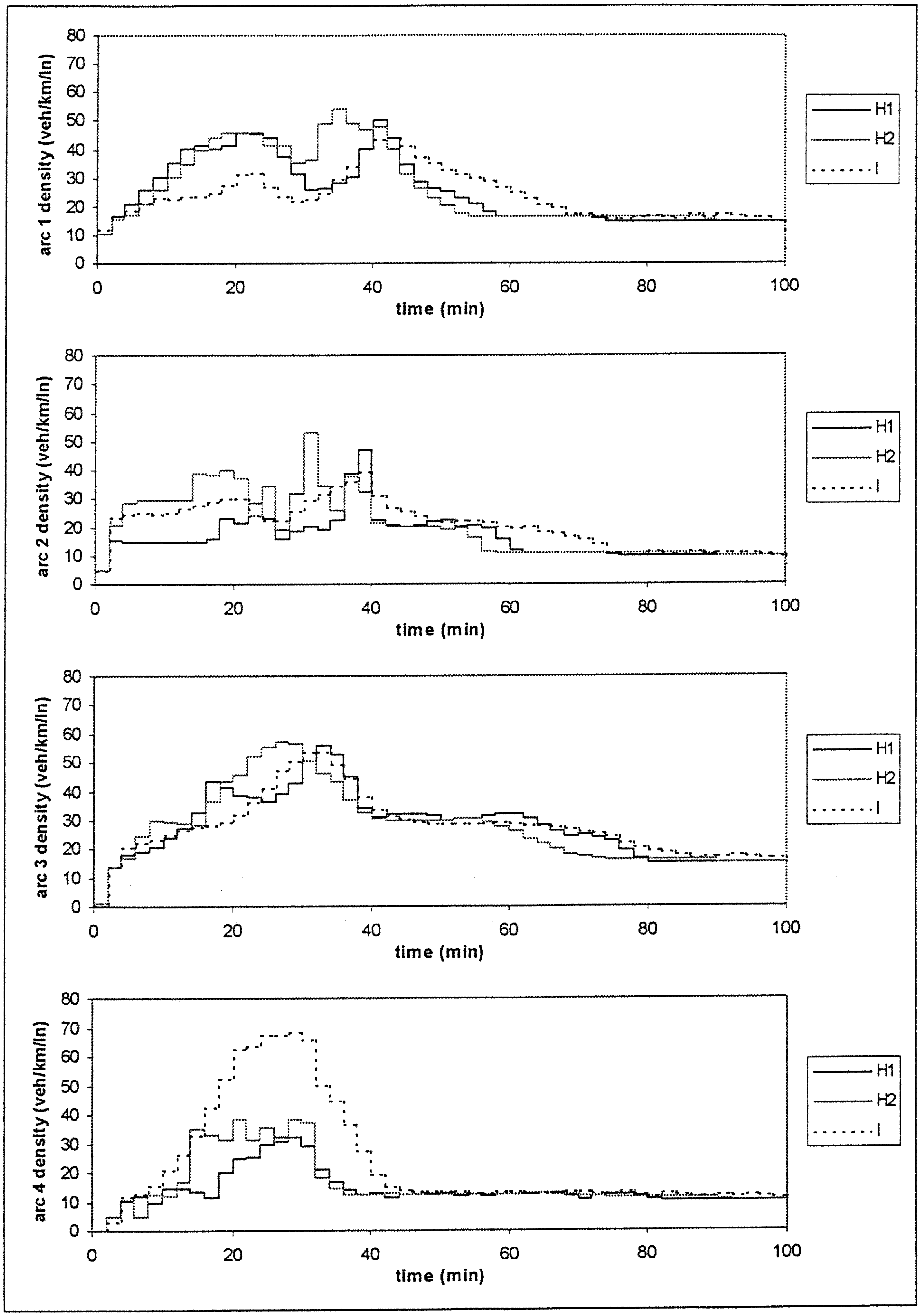


Figure 4.4: Density vs. Time



relatively small, its effect on the upstream arcs (3, 2, and 1) is significant. The high densities caused by the flow reduction before $t = 30$ tend to dissipate more quickly in the STQ-based model than in INTEGRATION, particularly on arcs 2 and 3. This discrepancy is likely due to the fact that the flow-density relationship associated with the traffic flow model of INTEGRATION is curvilinear and bounded above by the linear-segment relationship of the STQ (Astarita *et al*, 2000). As a result, the demand flow as a function of density tends to be slightly lower, particularly in the vicinity of the density values exhibited by arcs 3, 2, and 1 during this period. Overall, the results obtained with the STQ-based model compared fairly well with those obtained using INTEGRATION. It should be noted that the former required 4.2 sec of execution time on a 450 MHz Pentium II computer, while the latter required 780 sec on the same machine, for the network tested here.

Although some rather significant differences are observed between the *H1* and *H2* results, these data are generally more similar to each other than to the INTEGRATION output. Notable exceptions might be the density plots for arcs 2 and 3. It can be seen that the peak in density occurring in the INTEGRATION data on arcs 3, 2, and 1 at $t = 32$, 38, and 40, respectively, is quite closely matched by the *H1* output. The *H2* results consistently predict this peak several minutes earlier on each of these arcs. In general, the *H2* results are somewhat less "well behaved" than those of *H1*, as they exhibit more instability in general, and tend to oscillate between more extreme values.

5 Conclusions

A discrete-event discrete-flow node problem was defined, and a solution algorithm was proposed. The algorithm chooses the next vehicle to enter a node given the necessary information concerning the first vehicle on each of its upstream lanes, and the pair-wise precedence relationships that are obtained using a gap-acceptance model. The algorithm combines the pair-wise precedence relationships in such a way that unnecessary delays are avoided. A discrete-event network-loading procedure was presented, which ensures that the process of choosing the next vehicle to enter a node is executed any time there is a change in the information relevant to this decision. The algorithm was implemented in C++, using the discrete-flow arc model proposed by Mahut (2000b), which is based on a simplified model of traffic dynamics proposed by the same author, referred to as the space-time queue (STQ) (Mahut, 2000a).

Heuristics for choosing arrival and departure lanes in conjunction with the arc model were proposed, and two different departure lane-choice rules were implemented. The model was then tested on a small network, under time-varying demand and supply conditions. The results were compared with those obtained using the INTEGRATION traffic simulator. Some significant discrepancies in the output were observed for arcs on which a high degree of lane changing occurred. In general, the INTEGRATION model was able to produce higher flows on these arcs. This was particularly evident on a highway arc just upstream of an off-ramp, where congestion at the exit of this ramp was spilling back onto the highway. It might be said that the vehicles in the micro-simulator were able to react to this situation in a more efficient way than those of the STQ-based model. The latter model also produced significantly lower densities on this arc than were predicted by INTEGRATION. Overall, the results were found to compare quite well. The execution time of INTEGRATION for the network tested here was almost two hundred times that required by the STQ-based model. Clearly, the proposed model may have significant potential for real-time applications for which modern micro-simulators are currently too slow.

The results obtained with the first lane-choice heuristic were generally preferable to those obtained using the second rule. The first rule obliged vehicles to choose the closest lane to the permitted lanes for the turning at the exit of the arc, which was not currently blocked due to congestion. This rule thus minimized the number of lanes that a vehicle might have to cross at the downstream node in order to execute this turning. The second rule was based on minimizing the time at which the vehicle could enter the upstream node, as a function of the departure (and arrival) lane choice for the downstream arc. This rule resulted in more lanes being crossed at the nodes, which in turn meant more *node conflicts*. In general, the second rule produced lower flows on average on those arcs where a high degree of lane changing occurred. In general, then, it appears that the model is more efficient (produces higher flows) when conflicts between vehicle trajectories are resolved within the arcs rather than at the nodes.

The rather severe congestion simulated in this network presented a particularly difficult test for the model. Nevertheless, it was possible to reduce the cause of the discrepancies to a fairly specific situation for which the models were producing different results, for both density and flow. With respect to flow, it is not clear whether the cause of the discrepancy is related more to the lane-choice heuristics, or to the mathematical constraints underlying the arc model. Further work is required to investigate both of these possibilities. In general, lane-choice heuristics in micro-simulation models are extremely complex, and take into account very detailed information on a second-by-second basis (Rickert *et al*, 1997; Gipps 1986). The heuristics implemented here were quite primitive by comparison, and although the structure of the model does not permit the re-consideration of these decisions every second, it is quite likely that more sophisticated heuristics can be developed that will improve the model performance. It is also quite possible that the arc constraints are more conservative than those imposed by the INTEGRATION model. Slightly relaxing these constraints should produce higher flows and higher densities, which would reduce the discrepancies for both of these variables.

References

Astarita, Er-Rafia, Florian, Mahut, Velan (2000). A Comparison of Three Methods for Dynamic Network Loading. Submitted to the Annual Meeting of the Transportation Research Board, to be held in Washington, D.C., January 7-11, 2001.

Barceló, J., J. L. Ferrer and R. Grau (1994). AIMSUN2 and the GETRAM Simulation Environment. *Internal Report*, Departamento de Estadística e Investigación Operativa. Universitat Polit[illegible]cnica de Catalunya. See also http://www.tss-bcn.com.

Gabard, J. F., (1991). Car-Following Models, in *Concise Encyclopedia of Traffic and Transportation Systems* (M. Papageorgiou, ed.). Pergamon. pp. 337-341.

Gipps, P. G., (1986). MULTSIM: A Model for Simulating Vehicular Traffic on Multi-lane Arterial Roads. *Mathematics and Computers in Simulation* 28, pp. 291-295.

Law, and Kelton (1991). Simulation Modeling and Analysis, 2nd Ed. McGraw-Hill, Inc. 759 pp.

Leutzbach, W. (1988) *Introduction to the Theory of Traffic Flow,* Springer-Verlag Berlin, Heidelberg.

Mahut, M. (1999). Behavioural Car Following Models. Report CRT-99-31. Centre for Research on Transportation. University of Montreal. Montreal, Canada.

Mahut, M., (2000a). From Traffic Flow To Queueing Theory. To be presented at the 8th Annual Meeting of Euro Working Group on Tranportation. Rome, Itatly, September 11-14, 2000.

Mahut, M., (2000b). A discrete-flow arc model of traffic dynamics based on the space-time queue. Unpublished.

May, A. D. (1990) *Traffic Flow Fundamentals.* Prentice-Hall, Englewood Cliffs, NJ.

Rickert, M., K. Nagel, M. Schreckenberg, and A. Latour (1997). Two lane traffic simulations using cellular automata. *Physica A,* 234:687.

Rothery, R. W., (1999). Car Following Models. *Revised Monograph on Traffic Flow Theory.* N. H. Gartner, C. J. Messer and A. K. Rathi (eds.). http://www.tfhrc.gov.

Velan, S. M. (1997). Gap Acceptance of Permissive Movements at Signalised and Unsignalised Intersections. Master's Thesis. Queen's University. Kingston, Canada.

Transportation Research Board (1998). *Special Report 209: Highway Capacity Manual,* 3rd ed.., National Research Council.

Van Aerde, M. (1999). INTEGRATION© Release 2.20 for Windows: User's Guide.

Conclusions and Recommendations

A re-formulation of the well-known car-following paradigm based on safe stopping distances was proposed in which each vehicle is referenced to a different moment in time, and an alternative functional form was proposed for describing the deceleration trajectory of vehicles in such a model. The equilibrium relationship implied by the proposed deceleration function appears to have more desirable properties than the one derived from the well-known quadratic relationship. The functional form of the equilibrium relationship may thus prove to be very useful for curve fitting exercises aimed at characterising the flow-density relationship for different types of road facilities. Moreover, it may be conjectured that the proposed function generally produces more realistic results than the commonly used quadratic relationship. One avenue of further research in this area would be to implement a discrete-time solution procedure for the proposed car-following model, and to compare it with other well-known models on some numerical examples. In particular, it may be of interest to compare the proposed model with other models in the literature that are known to produce similar equilibrium relationships.

Building upon the analysis of the proposed car-following model, a simplified car-following relationship was further proposed. The simplified relationship possesses the particular property that for a continuous sequence of constrained trajectories, the last trajectory can be written strictly as a function of the first, with the proper temporal and spatial offsets. When the physical location of delays is fixed in space, the inverse of the vehicles' trajectories – $t(n,x)$, rather than $x(t,n)$ – can be solved for directly at these positions, as a function of $t(n,x)$ at the two neighbouring (upstream and downstream) positions. Traffic networks satisfy this property since delays occur at signalised intersections and highway merges, both of which are fixed in space.

This general property – of being able to solve the traffic flow model over arbitrarily long distances – is characteristic of only one other traffic model in the literature (Newell, 1993), although only a graphical technique is shown to have this property in the original paper. This technique is not easily adapted to an automated algorithmic solution. The practical applications of Newell's model are further limited by the fact that it is based on a continuous notion of flow. Success at extending continuous flow models to capture the effects of interactions between parallel lanes has been rather limited (Velan, 2000).

It is evident that the constraints defining the proposed discrete-flow model are equivalent to the cellular automata (CA) model known as CA-184 (Nagel, 1996A), when the reaction time is taken to be equal to the discrete time step. The fact that CA models are based on discretised space, time, and flow limits the definition of speed to discrete values as well. A well known stochastic extension of this model intended for traffic simulation defines five possible values of speed; the maximum value being equal to 135 km/h (*ibid.*). This in turn implies five discrete values for the maximum flow that can be defined for an arc, which may not provide sufficient flexibility of calibration for practical applications (Rillett, 2000). Because time is continuous in the proposed discrete-flow model, maximum speed and maximum flow are real-valued, and hence can be calibrated to any desired accuracy. The steady-state relationship between flow and density was derived for the model, and was found to be the well known two-linear-segment relationship characteristic of both Newell's simplified kinematic wave model (Newell, 1993) and CA-184 (Nagel 1996A).

This model was extended to the multi-lane case, where the effects of vehicle trajectories in conflict due to lane changing were addressed. By using a first-in-first-out (FIFO) rule to resolve all lane-changing conflicts, it was found that delays due directly to such conflicts could be calculated at the moment that a vehicle enters an arc. The FIFO assumption was used in lieu of a gap acceptance model, in constrast to multi-lane car-following and cellular automata models (Gipps, 1986; Rickert, 1997; Van Aerde, 1999; Barceló *et al*, 1994). The multi-lane arc model defines a mapping from the time-dependent lane-based demand flows, i.e., the partial demands by arrival lane and departure lane, to the arc variables of flow, density, and speed. The model is consistent with a "microscopic" description of each vehicle's trajectory. This provides a means of validating the assumptions by constructing examples and observing the movement of individual vehicles.

Two series of tests were executed on a simple network and the results were compared to those obtained with the AIMSUN2 traffic simulation package. The tests were to evaluate the effect of traffic weaving on a two-lane arc, with and without a traffic signal at the downstream end. Each case was executed for three different maximum speeds. Because the demand and supply were invariant in time (the signal produced a constant mean supply over the duration of each cycle), the arc variables stabilised to an apparently stationary condition after a short time. The models were thus compared on the basis of the average values of flow and density for the different scenarios. The majority of the test results compared very well between the proposed arc model and AIMSUN2. The execution time of the proposed model was no more than one-tenth the execution time of AIMSUN2. This ratio was found to decrease with increasing congestion in the network, as would be expected by the "arc-based" nature of the algorithm. For the same reason, this ratio would also be expected to increase with average arc length.

The question of how to appropriately resolve the conflicts between vehicle trajectories at nodes was addressed. Conflict "resolution" refers to the decision of which vehicle will be the next to proceed into the intersection. It was found that the application of the well-known concept of gap acceptance – which only applies to two vehicles at a time – might not always imply in a direct way a reasonable resolution to the general case of multiple simultaneous conflicts at a node. A solution method was proposed, which resolves the conflicts in such a way that unnecessary delays are avoided. Some realistic examples were presented to demonstrate the rationale. Simulation tests of the node model used a heuristic rule for determining the arrival and departure lanes for each vehicle for each arc. Although the departure lane from the upstream arc essentially predetermines the choice of arrival lane on an arc, the choice of departure lane can be quite critical in determining a realistic prediction of the resulting traffic conditions.

Two different heuristics for departure lane choice were tested on a small network and the results were compared with those obtained with the INTEGRATION traffic simulation package. The test results in general compared quite well. Although some noticeable differences were observed on arcs where a high degree of lane changing occurred, these differences did not have a major impact on the overall path travel times, which are the critical measure for dynamic traffic assignment algorithms. The execution time of the INTEGRATION model on the test network was almost 200 times that required by the proposed model.

It may be concluded that the simplifications made in designing the arc model did not have a major impact on the quality of the results obtained. At the same time, the reduction in computational effort relative to micro-simulation approaches was quite remarkable. Important avenues of further research involve the application of the model within a dynamic traffic assignment (DTA) algorithm, and exploring the possibility of using the model for real-time Intelligent Transportation Systems (ITS) applications, such as en-route guidance information and adaptive traffic signal control.

Clearly, the proposed model can only be used in conjunction with DTA algorithms for which the inputs are compatible with the available model outputs. In particular, DTA approaches that require the derivative of travel time with respect to demand flow – such as those inspired by static equilibrium models (Wu *et al*, 1998) – cannot be used since such derivatives cannot be obtained analytically for this model. One avenue of research is to explore the possibility of estimating such derivatives for path travel times from the observed flow rates in the simulation model. Other possible DTA approaches that do not require derivatives are the well-known Method of Successive Averages (MSA) – which has been used successfully in the DYNAMIT project (Ben Akiva *et al*, 1999) – and a technique used in conjunction with CA models in which a certain percentage of vehicles with the longest travel times are re-routed after each iteration (Gawron, 1998). Ultimately, the various potential DTA methods should be evaluated on the basis of how quickly they converge for medium and large-scale networks.

Adapting the proposed model for use in real-time ITS applications may involve more fundamental issues related to the model itself. This is because models to be used for real-time applications must have a mechanism by which the definition of the network state can be periodically updated in real-time using traffic detector data. This type of update is necessary to ensure that errors in the underlying model assumptions – in particular those concerning route choice – are not allowed to propagate within the model, leading to a divergence between the simulated and real network states. The ability of a model to accept this kind of update is the fundamental difference between *off-line* and *on-line* traffic modelling, and is currently a very active area of research in traffic simulation (Kaumann *et al*, 2000). Exactly how real-time detector data can be used as an input to the model is the crux of the problem. Relatively naive approaches simply use off-line models that can only accept inputs in the form of time-dependent path flows at the origins. Off-line models that use turning proportions at each node instead of assigning vehicles to paths are in fact more amenable to real-time updating, but these models have less predictive ability than path-based models. How to best ensure that the simulated network state within the proposed model does not deviate from the actual evolution of a real network using real-time detector data is an important and challenging direction for future research.

Bibliography

Astarita V., Er-Rafia, Florian, Mahut, Velan (2000). A Comparison of Three Methods for Dynamic Network Loading. Submitted to the Annual Meeting of the Transportation Research Board, to be held in Washington, D.C., January 7-11, 2001.

Adamo, V., V. Astarita, M. Florian, M. Mahut and J.H. Wu (1999). Modeling of spillback of congestion in link based dynamic network loading models: A simulation model with applications. *Transportation and Traffic Theory: Proceedings of the 14th International Symposium*. A. Ceder, (ed.) Pergamon, pp. 555-574.

Adamo V., V. Astarita, M. Florian, M. Mahut and J.H. Wu (1999). Analytical Modelling of Intersections in Traffic Flow Models with Queue Spill-back. Report 99-52. Centre for Research on Transportation, University of Montreal. Montreal, Canada.

Ben-Akiva, M., and M. Bierlaire (1999). DYNAMIT: A network-wide traffic prediction system. Transportation Research Board, 78th Annual Meeting, Washington , D.C. See also http://www.its.mit.edu.

Ben Akiva, M., H. Koutsopoulis, and T. Toledo (2000). MITSIMLab: Recent Developments & Applications. Presented at the INFORMS Spring 2000 Conference, Salt Lake City, May 7-10, 2000. See also http://www.its.mit.edu

Barceló, J., J. L. Ferrer and R. Grau (1994). AIMSUN2 and the GETRAM Simulation Environment. *Internal Report*, Departamento de Estadística e Investigación Operativa. Universitat Politécnica de Catalunya. See also http://www.tss-bcn.com.

Cassidy (1998) Bivariate relations in nearly stationary highway traffic. *Transportation Research B* 32B (1), pp. 49-59.

Cayford, R., W. Lin and C.F. Daganzo (1997). The NETCELL simulation package: Technical description. *California PATH Research Report UCB-ITS-PRR-97-23*.

Chang, G.L., H.S. Mahmassani, and R. Herman (1985) A macroparticle traffic simulation model to investigate peak-period commuter decision dynamics. *Transportation Research Record* 1005:107-120.

Cremer, M., and J. Ludwig (1986). A fast simulation model for traffic flow on the basis of boolean operations. *Mathematics and Computers in Simulation* 28:297-303.

Daganzo, C. F. (1994) The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research B, 28B(4):269-287.

Daganzo, C.F. (1995A). Properties of link travel time functions under dynamic loads. *Transportation Research B*, vol. 29B, No. 2, pp. 95-98.

Daganzo, C. F. (1995B). Requiem for second-order fluid approximations of traffic flow. *Transportation Research B*, 29B (4):277.

Edie, L. C. (1974). Flow Theories, in *Traffic Science* (D. C. Gazis, ed.). John Wiley & Sons, New York. pp. 7-11.

Er-Rafia, K. (2000). Un algorithme de chargement dynamique des réseaux: mésosimulation du flot dynamique avec capacités explicites. Masters thesis. Département d'informatique et de recherche opérationnelle, Université de Montréal, 133 pp.

Friesz, T.L., D. Bernstein, T.E. Smith, D.L. Tobin and B.W. Wie. A variational inequality formulation of the dynamic user equilibrium problem. *Operations Research,* Vol. 41, pp. 179-91, 1993.

Gabard, J. F., (1991). Car-Following Models, in *Concise Encyclopedia of Traffic and Transportation Systems* (M. Papageorgiou, ed.). Pergamon. pp. 337-341.

Gawron. C. (1998). Simulation-Based Traffic Assignment. Ph.D. Thesis, Univesity of Köln. Köln, Germany.

Gazis, D.C, Herman R, and R.B. Potts (1959) Car following theory of steady state flow. Operations Research 7 (4), pp 499-505.

Gazis, D. C, Herman, R, and R.B. Potts (1961) Nonlinear follow-the-leader models of traffic flow. *Operations Research* 9 (4), pp 545-567.

Gerlough, D. L., and M.J. Huber (1975) *Traffic Flow Theory,* Special Report No. 165. Transportation Research Board, National Research Council, Washington, DC, 1975.

Gerlough, D. L., in *Proceedings of the 35th Annual Meeting* edited by F. Burggrat and E.M. Ward. Highway Research Board, Washington, DC, 1956, p.543. See also J. H. Mathewson, D.L. Trautman, and D.L. Gerlough (1955). Study of Traffic flow by Simulation. *Highway Research Board Proceedings.*

Gerlough D. L, and Huber, M. J. (1975) *Traffic Flow Theory,* Transportation Research Board Special Report 165, Washington D.C.

Gipps, P.G. (1981) A behavioural car following model for computer simulation. *Transportation Research 15B,* pp 105-111.

Gipps, P.G. (1986). MULTSIM: A Model for Simulating Vehicular Traffic on Multi-lane Arterial Roads. *Mathematics and Computers in Simulation* 28, pp. 291-295.

Greenshields, B. D. (1955). A study of traffic capacity. *Proc. Highw. Res. Bd.,* Washington, D.C. 23:57.

Jayakrishnan, R., H.S. Mahmassani, and Ta-Yin Hu. An evaluation tool for advanced traffic information and management systems in urban networks. *Transportation Research,* Vol. 2C, pp. 129-47, 1994.

Herman, R, Montroll, E. W, Potts, R.B, and Rothery, R.W. (1959) Traffic dynamics: analysis of stability in car-following. *Operations Research* 7 (1), pp 86-106.

Herman (1991) Traffic dynamics through human interaction. *Journal of Economic Behaviour and Organization 15* (2), pp. 303-311.

Heydecker, B.G., and J. D. Addison (1996). Analysis of Traffic Models for Dynamic Equilibrium Traffic Assignment. Presented at the 4th Meeting of the EURO Working Group on Transportation. Newcastle upon Tyne, September 9-11.

Heydecker, B.G., and J. D. Addison (1996). An Exact Expression of Dynamic Traffic Equilibrium. *Transportation and Traffic Flow Theory*. J. P. Lesort (ed.). Proceedings of the 13th International Symposium on Transportation and Traffic Theory (ISTTT), pp. 359-383.

Hugosson, B., and H. Andersson (1999). Evaluation of AIMSUN2 in Stockholm. The "SMARTEST" Project. http://www.its.leeds.ac.uk/smartest.

Hurdle, V. F. (1991). Queueing Theory Applications, in *Concise Encyclopedia of Traffic and Transportation Systems* (M. Papageorgiou, ed.). Pergamon. pp. 337-341.

Hurdle, V. F. (1986) Technical note on a paper by Andre de palma, Moshe Ben-Akiva, Claude Lefevre, and Nicolaos Litinas entitled "stochastic equilibrium model of peak period traffic congestion." *Transportation Science*, 12(3):287-289.

Kerner, B. S., and P. Konhauser (1993) Cluster effect in initially homogenous traffic flow. Physical Review E, 48(4):R2335-2338.

Kaumann, O.,Chrobok, R., Wahle, J., and M. Shcreckenberg (2000). Traffic Forecast Using On-line Simulations. Presented at the 8th Meeting of the EURO Working Group on Transportation. Rome, Italy, September 11-14, 2000.

Krauss, S. (1998) The role of acceleration and deceleration in microscopic traffic flow models. Preprint No. 980478, Transportation Research Board, 77th Annual Meeting, January 11-15, 1998, Washington D.C.

Krauss, S., P. Wagner and C. Gawron (1997) Metastable states in a microscopic model of traffic. Physical Review E, 55(5):5597-5602.

Kuhne, R. D. (1984). Macroscopic freeway model for dense traffic stop-start waves and incident detection. Proceedings of the 9th International Symposium on Transportation and Traffic Theory, J. Volmuller and R. Hamerslag (eds.) p. 21. VNU Science Press, Utrecht, The Netherlands.

Kuhne, R. D. (1993). Non-linearity stochastics of unstable traffic flow. *Proceedings of the 12th International Symposium on Transportation and Traffic Theory*. C.F. Daganzo (ed.), pp 367. Elsevier, Amsterdam, The Netherlands.

Law, and Kelton (1991). Simulation Modeling and Analysis, 2nd Ed. McGraw-Hill, Inc. 759 pp.

Lebacque (1996). The Godunov scheme and what it means for first order traffic flow models. *Transportation and Traffic Flow Theory*. J. P. Lesort (ed.). Proceedings of the 13th International Symposium on Transportation and Traffic Theory (ISTTT), pp. 647-677.

Leutzbach, W. (1988) *Introduction to the Theory of Traffic Flow*, Springer-Verlag Berlin, Heidelberg.

Lighthill, M. J. and G.B. Whitham (1955) On kinematic waves I: Flood movement in long rivers. II: A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London*, A229:281-345.

Lindley, D. V. (1952). The Theory of Queues with a Single Server. *Proceedings of the Cambridge Philosophical Society*, 48:277-89.

Little, J. D. C. (1961). A proof for the queueing formula $L = \lambda W$. *Operations Research*, 9:383-7.

Mahmassani, H. S., T. Hu and R. Jayakrishnan. (1995) Dynamic traffic assignment and simulation for advanced network informatics (DYNAMSMART). *In Urban traffic networks: Dynamic flow modeling and control*, N.H. Gartner and G. Improta (eds.). Springer, Berlin/New York.

Mahut, M. (1999a). Speed-maximizing car-following models based on safe stopping rules. Preprint 990351, Transportation Research Board, 78th Annual Meeting, January 10-14, 1999.

Mahut, M. (1999b). Behavioural Car Following Models. Report CRT-99-31. Centre for Research on Transportation. University of Montreal. Montreal, Canada.

Mahut, M. (2000). From Traffic Flow to Queueing Theory. Presented at the 8th Meeting of the EURO Working Group on Transportation (EWGT). Rome, Italy, September 11-14, 2000.

Makigami, Y., G.F. Newell and R. Rothery (1955). Three-dimensional representations of traffic flow. *Transportation Science*, 5:302-13.

May, A. D. (1990) *Traffic Flow Fundamentals*. Prentice-Hall, Englewood Cliffs, NJ.

Nagel, K. (1996A) Particle hopping models and traffic flow theory. *Physical Review E*, 53(5):4655-4672.

Nagel, K. (1996B) From particle hopping models to traffic flow theory. Preprint No. 981331, Transportation Research Board, 77th Annual Meeting, January 11-15, 1998, Washington D.C.

Nagel, K. and H.J. Herrmann. (1993) Deterministic models for traffic jams. *Physica A*, 199:254-269.

Nagel, K. and M. Paczuski. (1994) Emergent traffic jams. *Physical Review E*, 51:2909-2918.

Nagel, K. and M. Schreckenberg. (1992) A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221-2229.

Newell, G. F. (1993) A simplified theory of kinematic waves in highway traffic. Part I: General Theory. Part II: Queueing at freeway bottlenecks. Part III: Multi-destination flows. *Transportation Research B*, 27B(4):281-313.

Payne, H. J. (1971). Models of freeway traffic and control. *In Mathematical Models of public systems*, 1:51. La Jolla, CA, Simulation Council.

Papageorgiou, M. (1990). Dynamic Modelling, Assignment, and Route Guidance in Traffic Networks. Transportation Research, 24B(6), 471-95.

Pipes, L. A. (1953). An Operational Analysis of Traffic Dynamics. *Journal of Applied Physics* 24, pp. 271-281.

Prigogine, I. and R. Herman. (1971) *Kinetic theory of vehicular traffic*. Elsevier, New York.

Richards, P. I. (1956) Shock waves on the highway. *Operations Research*, 4:42-51.

Rickert, M., K. Nagel, M. Schreckenberg and A. Latour (1997). Two lane traffic simulations using cellular automata. *Physica A*, 234:687.

Rillet, L. R., and B. Raney (2000). Transportation Planning Implications of the Highway Supply Relationship in TRANSIMS. Preprint 00-0762, Transportation Research Board, 79th Annual Meeting, January 9-13, 2000.

Rillet, L. R., K. Kim, B. Raney (2000). A Comparison of the Low Fidelity TRANSIMS and High Fidelity CORSIM Highway Simulation Models using ITS Data. Preprint 00-0678, Transportation Research Board, 79th Annual Meeting, January 9-13, 2000.

Ross, P. (1988). Traffic Dynamics. *Transportation Research B*, vol. 22B, No. 6, pp. 421-435.

Rothery, R. W. (1999). Car Following Models. *Revised Monograph on Traffic Flow Theory*. N. H. Gartner, C. J. Messer and A. K. Rathi (eds.). http://www.tfhrc.gov.

Schreckenberg, M., A. Schadschneider, K. Nagel, and N. Ito. (1995). Discrete stochastic models for traffic flow. *Physical Review E*, 51:2939-2949.

Simon, P. M., J. Esser and K. Nagel (1999). Simple Queueing Model Applied to the City of Portland. *International Journal of Modern Physics* C, vol. 10, No. 5, 941-960.

Smith. M. J. and J. B. Wisten (1996). A Distributed Algorithm for the Dynamic Traffic Equilibrium Assignment Problem. *Transportation and Traffic Flow Theory*. J. P. Lesort (ed.). Proceedings of the 13th International Symposium on Transportation and Traffic Theory (ISTTT). Lyon, July 1996. pp.385-408.

Transportation Research Board (1998). Special Report 209: Highway Capacity Manual, 3rd ed. National Research Council.

Treiterer, J. and J.A. Myers (1974). The hysteresis phenomenon in traffic flow. 6th *International Symposium on Transportation and Traffic Theory*. D.J. Buckley (ed.), A.H. & A.W. Reed Pty Ltd., Artarmon, New South Wales.

Van Aerde, M.(1999) INTEGRATION© Release 2.20 for Windows: User's Guide.

Velan, S. M. (1997). Gap Acceptance of Permissive Movements at Signalised and Unsignalised Intersections. Master's Thesis. Queen's University. Kingston, Canada.

Velan, S., and M. Van Aerde (1998). The impact of driver and flow variability on capacity estimates of permissive movements. *Transportation Research*, Vol. 32A.

Velan, S. (2000). *The cell-transmission model: A new look*. Ph.D. thesis. Département d'informatique et de recherche opérationnelle, Université de Montréal, 300 p., 2000.

Whitham, G. B. (1990). Exact solutions for a discrete system arising in traffic flow. Proc. Royal Society London, A428:49-69.

Wu, J. H., Y. Chen, and M. Florian (1998). The continuous dynamic network loading problem: A mathematical formulation and solution method. *Transportation Research*, Vol. 32B, p. 173-87.

Ziliaskopoulos, A. and S. Lee (1997). A cell transmission based assignment-simulation model for integrated freeway/surface street systems. *Transportation Research Record*, 1997.

CURICULUM VITAE

EDUCATION

09/96 – 12/00 University of Montreal Montreal, Canada
Ph.D in Computer Science and Operations Research
- Operations Research Option

05/94 – 08/96 University of Toronto Toronto, Canada
Master of Applied Science (M.A.Sc.) in Civil Engineering
- Transportation Option

09/91 – 04/93 University of Toronto Toronto, Canada
09/88 – 04/90
09/86 – 04/87
Bachelor of Applied Science (B.A.Sc.) in Civil Engineering

AWARDS RECEIVED

09/98 – 08/00 University of Montreal Montreal, Canada
Post-Graduate Scholarship (PGS-B), Natural Sciences and Engineering Research Council of Canada (NSERC)

04/98 University of Montreal Montreal, Canada
Bursary , Centre for Research on Transportation (CRT)

09/96 – 08/98 University of Montreal Montreal, Canada
Research Assistantship under Professor M. Florian, CRT

04/95 – 08/96 University of Toronto Toronto, Canada
Research Assistantship under Professor G. Steuart, Joint Program in Transportation

04/93 University of Toronto Toronto, Canada
Bursary, Canadian Society for Civil Engineering (CSCE), Toronto Section

MEMBERSHIPS

01/96 – present
Student member, Institute of Transportation Engineers (ITE)

01/98 – present
Student member, Institute for Operations Research and the Management Sciences (INFORMS)

5280 BEACONSFIELD AVE. • MONTREAL QC • H3X 3R8
PHONE (514) 343-6111 X 8706 • FAX (514) 369-7121 • E-MAIL
MICHAELM@CRT.UMONTREAL.CA

04/98 – present
Friend, Commitee on Traffic Flow Theory and Characteristics (A3A11), and Committee on Tranportation Network Modelling (A1C05), Tranportation Research Board (TRB) (United States).

TEACHING ASSISTANTSHIPS

University of Montreal — Montreal, Canada
Introduction to C Language Programming

University of Toronto — Toronto, Canada
Transport 1 – Design
Traffic Engineering
Geomechanics

LANGUAGES

- Fluent in English, French, and Polish
- Rudimentary Spanish

ARTICLES ACCEPTED FOR PUBLICATION IN REFEREED JOURNALS

Comparison of Three Methods for Dynamic Network Loading
V. Astarita, M. Florian, K. Er-Rafia, M. Mahut, S. Velan

- To be published in Transportation Research Record
- To be presented by M. Mahut at the 80th Annual Meeting of the Transportation Research Board. Washington, D.C., January 7-11, 2001. (paper #01-0199)

ARTICLES SUBMITTED TO REFEREED JOURNALS

From Traffic Flow to Queueing Theory
M. Mahut

- Presented at the 8th Meeting of the Euro Working Group on Transportation (EWGT). Rome, Italy, September 11-14, 2000. Currently under consideration for a special issue of the European Journal of Operations Research (EJOR)

Behavioural Car Following Models
M. Mahut

- Submitted to Transportation Science. Published as report CRT-99-31, Centre for Research on Transportation, University of Montreal, 1999.

Analytical Modelling of Network Nodes in Traffic Models with Queue Spill-back
V. Adamo, V. Astarita, M. Florian, M. Mahut, S. Velan

- Submitted to Transportation Research Part C: Emerging Technologies

OTHER REFEREED CONTRIBUTIONS

The Development of a Mesoscopic Traffic Simulation Model

M. Mahut, M. Florian, N. Tremblay

- Presented at the 7th World Congress on Intelligent Transportation Systems, and published in the proceedings thereof (on compact disc). Turin, Italy, November 6-9, 2000.

Modeling the Spill-back of Congestion in Link-based Dynamic Network Loading Models: A Simulation Model with Applications

V. Adamo, V. Astarita, M. Florian, M. Mahut, J.H. Wu

- Presented at the 14th International Symposium on Transportation and Traffic Theory (ISTTT), and published in the proceedings thereof. Jerusalem, Israel, July 19-23, 1999. Elsevier, pp. 555-573.

Speed Maximizing Car Following Models based on Safe Stopping Rules

M. Mahut

- Presented at the 78th Annual Meeting of the Transportation Research Board. Washington, D.C., January 10-14, 1999. (paper #99-0351)

NON-REFEREED CONTRIBUTIONS

A Mesoscopic Approach to the Dynamic Network Loading Problem

M. Florian, M. Mahut, N. Tremblay

- Presented by M. Mahut at the spring meeting of the Institute for Operations Research and the Management Sciences (INFORMS). Salt Lake City, Utah, May 6-10, 2000.

Quelques observations sur la microsimulation du trafic avec Aimsun/2

M. Florian, M. Mahut

- Submitted to the Ministry of Transportation of Quebec in compliance with contract #1220-97-RC01. November, 1999.

Applications of the Aimsun/2 Micro-simulator in Montreal

M. Florian, M. Mahut, N. Tremblay

- Presented by M. Mahut at the spring INFORMS meeting. Montreal, Quebec, April 26-29, 1998.

A Parametric Analysis of Arterial Travel Time

M. Mahut

- Report #58, Joint Program in Transportation, University of Toronto, 1996. (Master's Thesis)