VEHICLE DISTRIBUTION AND ROUTING IN A LARGE AUTOMATED TRANSPORTATION NETWORK

by
Alan J. Pue

Dissertation submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
1981

ABSTRACT

Title of Dissertation: Vehicle Distribution and Routing in a Large Automated Transportation Network

Alan J. Pue, Doctor of Philosophy, 1981

Dissertation directed by: John Baras
Associate Professor
Electrical Engineering

The problem of efficiently routing vehicles from many origins to many destinations in a large Automated Guideway Transit (AGT) network is formulated into an optimization problem. First, a system using a vehicle-follower strategy is modeled at various levels of complexity by aggregating states over sections of guideway link. At the simplest level, the state variables are the link densities while a more complex model includes the specific vehicle-follower control dynamics by defining a link velocity as an additional state variable. Discrete vehicle simulations of a merge junction, diverge-merge junction, and a station are used to show the models adequately represent actual link density and delay. All models are nonlinear and nonconvex.

An optimal control problem is formulated by devising a performance index based on total system delay (time averaged travel time) subject to the dynamic flow constraints of the network. Duality theory is applied to decompose the overall network dynamics into vehicle type (origin-destination pair) subnetwork constraints that are decoupled in vehicle type state and control but coupled through the interconnection variables of total density and total control. The resulting structure of the subnetwork dynamics is then exploited to allow a distributed control computation where each node in the network only

needs to communicate with neighboring nodes to optimize the dual function objective. An upper level coordinating control, localized to each link, seeks to satisfy the interconnection constraint that the sum of individual vehicle type densities is equal to the total vehicle density on the link. As a result, all control computations can be performed in a completely decentralized manner where information exchange only occurs between physically adjacent wayside control computers. Convergence of the algorithm is proven.

Computational studies of a 4 station, 58 link network is used to demonstrate the efficacy of the proposed algorithm. It is shown and proven that no duality gap exists for the problem (convex cost and non-convex constraints). Moreover, several suboptimal control schemes with reduced computational requirements are presented and evaluated.

DEDICATION

This work is dedicated
to my parents.

ACKNOWLEDGMENT

I wish to thank my thesis advisor, Professor John Baras, for his guidance and suggestions.

I would also like to express appreciation to the Applied Physics Laboratory, Johns Hopkins University, for the opportunity to investigate this topic, and my supervisor, B. F. Hoffman, for his encouragement. I thank Janet Evans for help in preparation of the manuscript.

Finally, parts of this study were made possible by support of the Urban Mass Transportation Administration of the Department of Transportation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$a_{i,i+1}(k)$	velocity from link i to link i+1 at time k
$A_j(k)$	subnetwork system dynamics matrix
a_s	service acceleration limit
C	set of diverge links
C_j	set of diverge links for subnetwork j
$d_i(\cdot)$	cost on link i
D_i	length of link i
h	headway
J	number of vehicle types
j_s	service jerk limit
K	number of time steps
$\mathcal{L}$	set of links
$\mathcal{L}_j$	set of links for subnetwork j
L	vehicle length
$p_{i,j}(k)$	adjoint variable for link i, vehicle type j at time k
$q_i(\cdot,\cdot)$	flow function into link i
$q_m(i)$	maximum flow on link i
$r_j(k)$	trip requests for vehicle type j
T	integration step-size
t_f	final time
u	vector of routing and dispatch control variables
u_j	vector of routing and dispatch control variables for vehicle type j
$u_{i,j}(k)$	routing control variable for vehicle type j on link i at time k
v, v'	total routing control variables
v_j, v_j'	total routing control variables for subnetwork j

$v_i(k)$, $v_i'(k)$	total routing control on link i at time k
$v_{max}(i)$	maximum velocity on link i
x_j	vector of link densities for subnetwork j
$x_{i,j}(k)$	vehicle type j density on link i at time k
y	link densities
y_j	link densities for subnetwork j
$y_i(k)$	density on link i at time k
$y_m(i)$	density at maximum flow for link i
z	vector of interconnection veriables
z_j	vector of interconnection variables for subnetwork j
α_n	upper level step-size at iteration n
λ	Lagrange multipliers for total density
μ, μ'	Lagrange multipliers for total control

1. INTRODUCTION

Automated Guideway Transit (AGT) systems are a relatively recent development in transportation technology, intended to serve urban regions and major activity centers such as airports, shopping districts, and universities. They are characterized by automatically controlled vehicles (driverless) of small-to-moderate size, operating on dedicated guideway networks that have either on-line or off-line stations. In addition, AGT systems can vary in complexity according to vehicle size, system capacity, network density, and whether routes are fixed, selected at the time of trip request, or dynamically selected as vehicles progress through the network.

1.1 The Routing Problem

A large AGT network may be briefly described as a dense network composed of hundreds of miles of connected guideways, joining as many as a hundred stations (typically off-line), and containing possibly thousands of vehicles. Consequently, the problem of efficiently distributing and routing vehicles between many origins and destinations is of considerable importance in terms of customer acceptance, cost-effective operation, and energy use.

One approach to this problem is a reservation system (or synchronous-method) that selects an entire unimpeded route for each vehicle before it departs an origin station. A second approach, possibly more appropriate to large systems, dispatches vehicles on demand at origin stations with local wayside computers at intersections solving the merging and routing problems on-line.

Algorithms for this latter approach are developed in this thesis and are structured by the following situation. As each vehicle enters an intersection, the vehicle is assigned to a particular outgoing link based on its origin, destination and the state of the network. The vehicle is then assigned to follow a preceding vehicle and controlled according to specified merging and spacing policies. It is assumed that these latter problems (merging and spacing regulation) are of a local nature and can be solved independently of the routing problem. An additional problem is the distribution of empty vehicles from stations where demand is low to stations where demand is high and is included as part of the routing problem.

From a practicality standpoint, there is considerable advantage in solving the routing problem by local control. Suppose routing strategy were determined by a central control node in the network. Such a node would periodically obtain information concerning traffic congestion from all other nodes in the network and solve the current routing problem. For a network with many nodes and vehicles, the communication requirements would be costly. In addition, there can be serious problems when either the control node or any communication link fails. Consequently, from a economic cost and reliability viewpoint it is beneficial to design a distributed routing control algorithm. For example, each merge junction may communicate only with its nearest neighbors to determine the dynamic routing of vehicles throughout the network, the objective being the avoidance of vehicle bunching and excessive delay.

In formulating the routing problem it is first necessary to develop models that adequately represent the dynamic behavior of

traffic flow in the network so that algorithms for the management of vehicles may be designed. In principle, inter-vehicle dynamics are known to a fairly high degree of accuracy for an automated system and therefore, one could easily derive a microscopic traffic flow model based on discrete vehicle behavior. However, such a model would be of extremely high order and would lead to a complex and costly control algorithm. As a matter of judgment and practicality it is likely that an aggregate model using macroscopic variables such as density, flow, and average velocity will prove to be sufficient for the successful application of on-line control laws.

1.2 Objectives

In view of the above discussion, the basic objectives of this work are as follows. The first is to develop aggregate traffic flow models for an automated transit system. Next, an optimization problem is to be formulated by devising a performance index representing system delay. Minimization is subject to the dynamic constraints of the network (flow model) with control variables being the routing of vehicles through the network and the dispatching of empty vehicles from surplus areas to deficit areas. In particular, it is desirable to design distributed algorithms as an effort to minimize communication requirements.

Background literature for investigation of this problem is reviewed in the following section.

1.3 Previous Work

The fundamental routing problem may be viewed as a nonlinear multicommodity flow problem, a problem that has received wide attention in the nonlinear programming literature, although primarily for the static case. Recently, more research has been devoted to the decomposition and efficient solution of large problems for both the static and dynamic situations. In particular, applications have centered on traffic, power, and computer communication networks. For the specific application of automated transit systems we begin by reviewing the work that has been the most analytic in nature and then discuss some pertinent work in the related fields of traffic and computer communication networks. Finally, theoretical work in the area of large-scale dynamic systems most useful for this application is summarized. First, the modeling problem is reviewed.

1.3.1 Traffic Flow Models

Previous work in developing traffic flow models has concentrated on the problem of automobile traffic in congested urban streets and freeways. Many models are based on the original continuum model of Lighthill and Whitham [1] derived by using the analogy to continuity of fluid flow. Cunningham [2] has applied this model to the routing of vehicles in an AGT network, although, the effects of intervehicle dynamics are not included. These effects have been studied with respect to automobile traffic by either empirical derivation of dynamics [3] at high density or application of statistical mechanics at low to moderate densities [4]. An additional problem that has received less attention is the description of vehicle dynamic behavior

at intersections [5].

Fortunately, the modeling of an automated network does not contain the complicating factors of an urban traffic network such as passing, parked vehicles and human behavior. The interactions between vehicles in a automated system may be described according to a well-defined longitudinal control law (Garrard et.al., [6] and Pue, [7]) that is based on accurate measurements of vehicle position and velocity.

The longitudinal control laws that have received the most attention may be classified according to one of two approaches. The first, usually termed point following, assigns each vehicle to a moving cell, the cells being propagated along the guideway network at predetermined velocities and spacings. In this case, propulsion commands are generated so that each vehicle maintains its location within its assigned cell. The second approach, termed vehicle-following, is a control scheme which allows communication between successive vehicles such that the motion of a given vehicle is controlled in accordance with the motion of its neighbors. In particular, the strategy of greatest practical interest is where a vehicle follows the motion of its immediate predecessor; this approach is specifically considered in this thesis.

1.3.2 Automated Transit Network Studies

The analysis of automated transit networks has been exclusively limited to point-following systems or a model which makes no distinction. In [8], a steady-state point-following model is assumed and flows are assigned to each link based on minimization of a performance index given by the sum of fleet size cost, travel time cost, and

wait time cost. Tong and Morse [9] also study a point-following system but divide the system into cells of consecutive slots. The vehicle management strategy examined employs a fixed routing policy where each vehicle is assigned a particular route before leaving a station. The vehicle is then assigned to occupy the first cell leaving the station which has an open slot for a vehicle traveling the designated route. It is shown in [9] that the system may be properly phased so that as two distinct cells pass through a merge, they merge together, slot by slot, and may be locally merged without conflict under certain explicit conditions.

In [10], a point-following network is analyzed from the viewpoint of queueing theory for both local merge control and synchronous control methods (Sec. 1.1). The two methods are compared by calculating expected delay where for the local merge case, a station is modeled to be a M/M/1 queue and a merge is a M/M/1-Mq queue where Mq is the maximum queue length. For the synchronous case, the probability of securing a reservation is computed.

In [2], a model based on conservation of flow is used to determine a suboptimal control (i.e., routing variables) which minimizes the maximum value of the sum, over all links, of the densities squared. Simulation of the strategy revealed a distinct oscillatory behavior in the vehicle densities.

1.3.3 Traffic Network Studies

There is a large body of literature concerning the traffic assignment problem [11] for the static case. Several investigators have included dynamics for the problem of freeway ramp metering as for

example, in [12]. A few references which have considered the network problem are discussed below.

In [13], Kaya develops a traffic flow model based on conservation of flow with flow being a nonlinear function of density. An optimization problem is formulated that minimizes a performance index based on waiting time to get on a freeway, travel time, waiting queue length and vehicle density with control variables being the flows on to the freeway entrance ramps. It is noted that the problem is separable and nonlinear programming techniques for separable programs may be applied, however, no specific algorithms are given.

Merchant and Nemhauser ([14], [15]) formulate a dynamic traffic assignment problem using a discrete time, single destination model of vehicle flow where the number of vehicles leaving each section is a nonlinear function of the number of vehicles on that section and the conservation of flow constraint is employed. A performance index sums over all sections a nondecreasing cost that is a function of vehicle number. The performance index is minimized by using a piecewise linear approximation and then applying linear programming decomposition techniques.

1.3.4 Related Work

We now discuss several references that are related to the general problem of routing flow in networks using decentralized approaches. In particular, much work has been accomplished in the area of computer communication networks.

One of the first works to apply mathematical programming to computer communication networks is found in [16]. Cantor and Gerla

formulate the optimal routing of packets in terms of extremal flows and then solve a restricted master problem using the gradient projection method. A linear subproblem is then solved to determine if the solution is optimal and if not, what extremal flow should enter the basis. The approach is static where all flows are determined apriori.

An on-line computational algorithm is presented by Gallager [17] where the gradient of the total delay with respect to the routing variables is computed in a distributed manner. That is, the computation is accomplished at each node in the network using information only from adjacent nodes. The proposed algorithm seeks to equilibrate marginal delay, and is shown to be loop free.

Bertsekas ([18], [19]) expands on the work of Gallager by formulating the distributed computation approach in terms of a mathematical programming problem using a gradient projection method. It is shown that Gallager's algorithm is a special case of a class of algorithms using this method.

Another distributed approach is proposed by Meditch [20] who uses the goal coordination technique for large scale systems to decompose the routing problem. The conservation of flow equations are decoupled by breaking each link into an input flow and an output flow. The dual problem is then solved where an upper level controller seeks to satisfy the interconnection constraints (the input flow is equal to the output flow on each link).

1.3.5 Large-Scale Systems Theory

There has been much work related to the analysis and design of large systems as described in [21]. We will restrict attention to the deterministic optimal control of nonlinear systems and cite several of the references most pertinent to this problem.

A unifying discussion of techniques used for decomposition of large mathematical programs is given by Geoffrion [22] where various approaches to problem manipulation and solution are surveyed.

Most of the work on large-scale dynamic systems has adopted the viewpoint of low-order first level controllers which are coordinated by a second level central controller to achieve optimal performance for the overall system [23]. Specific algorithms are analyzed by Pearson [24] who investigates the role of duality and coordination in multilevel control.

1.4 Outline

The approach taken in this thesis is to develop accurate dynamic models of vehicle flow and formulate a performance index based on total, time averaged, travel time. The resulting optimization problem has convex cost and nonconvex constraints. Duality theory is applied to decouple the overall network dynamics into vehicle type (origin destination pair) subnetworks. The triangular structure of the subnetwork dynamics allows a distributed control computation to solve the dual problem while an upper level coordinating control, localized to each link, seeks to satisfy the interconnection constraint that the sum of the individual vehicle type densities is equal to the total vehicle density on the link.

In Section 2, an extended summary of the results of this work is given.

Section 3 provides a general description of an automated transportation system, the network layout, customer demand, associated control problems, performance measures, and motivation for the approach used to solve the routing problem.

The modeling of traffic flow for vehicle-follower systems is given in Section 4. The simplest model defines state as the vehicle density on a given guideway section while a more complex model includes the specific vehicle follower control law used between vehicles, and is represented through a link velocity, an additional state in the model. The validity of the models is tested with a discrete vehicle simulation and shown to well represent the actual system over a wide range of conditions.

In Section 5, the network control problem is formulated into an optimal control problem using the dynamic constraints derived in Section 4 and a performance index which averages travel time over all links in the network. The advantages of applying duality are then demonstrated.

Section 6 gives a description of the specific algorithm used for solution while Section 7 proves convergence. Section 8 contains computational results when the algorithm is applied to a 4 station, 58 link network. Certain aspects of convergence such as run time, state initialization, Lagrange multiplier initialization, step-size selection, control initialization, and problem duration are examined.

Section 9 proposes several suboptimal control strategies that approximate the optimal control algorithm in various ways. Simulations

of these strategies show significant savings in computational cost with little degrdation in performance.

Section 10 discusses practical issues concerning implementation such as algorithm initiation and termination criteria, and computer storage and speed requirements.

Finally, Section 11 contains conclusions and recommendations for further research.

2. SUMMARY

The basic objective of this work is to develop on-line control algorithms for the routing of vehicles through an automated transportation network. The first step is to derive and evaluate a set of macroscopic models analogous to traffic flow models, for the particular case of automated vehicles under vehicle-follower control. Next, an optimization problem is formulated and solved by distributed computation. Finally, the optimal algorithm as well as several suboptimal strategies are evaluated by computer simulation.

2.1 Flow Models

The flow models assume a control law where each vehicle controls its motions according to the motions of the immediately preceding vehicle, the control policy being a constant time headway of h seconds.

Two models of traffic flow for an infinite string of vehicles are derived. The first is based on a static velocity-spacing relationship, that is, vehicle spacing is given by hv + L where v is vehicle velocity and L is vehicle length. The second model includes the specific vehicle follower control dynamics by aggregating a velocity state over a guideway section. The former was chosen for algorithm design.

For both models, a flow function was developed to represent the behavior of flow when links are connected into junctions, that is, merges and diverges. The flow function was derived by observing a discrete vehicle simulation of a merge junction and comparing several possible models based on mathematical derivation and physical intuition. By a process of discovering deficiencies in the proposed

models and providing improvements, a final version that well represents behavior at junctions over a wide range of densities and bunchiness of input flow was selected. For example, at a headway of 3 sec the model compares favorably to actual vehicle density in a range of .01 veh/m to 0.1 veh/m where the guideway section length is 100 m and the maximum velocity is 15 m/sec. Thus, the lower bound corresponds to a single vehicle, while the upper bound corresponds to vehicles at a 2 m/sec velocity. Outside this range, the behavior of individual vehicles becomes significant and the averaging technique used for the model is invalid. This became evident when simulating a diverge and station.

The flow function has the form, $q_{i+1}(y_i, y_{i+1})$, where q_{i+1} is the flow into link i+1 and y_i, y_{i+1}, are the total vehicle densities on links i and i+1, respectively. The function depends on the physical characteristics of the guideway section through specification of a maximum velocity on each link. The function, q_{i+1}, is nonlinear.

The flow due to a particular vehicle type j is assumed to be the fractional portion of that vehicle type on a guideway section. If $x_{i,j}$ is the density of vehicle type j and y_i is the total density then the flow due to type j is

$$q_{i+1}(j) = (x_{i,j}/y_i) q_{i+1}(y_i, y_{i+1}) \tag{2.1}$$

where all densities are a function of time. Defining at time k, the velocity function, $a_{i,i+1}(k) = q_{i+1}(y_i, y_{i+1})/y_i$, the density of vehicle type j on link i is given in discrete time by

$$x_{i,j}(k+1) = x_{i,j}(k) + T[a_{i-1,1}(k)x_{i-1,j}(k) - a_{i,i+1}(k)x_{i,j}(k)]/D_i \qquad (2.2)$$

where T is the integration step size and D_i is section length. The above model is extended to include a merge and a diverge. For a diverge, the routing control variable, $0 \leq u_{i,j}(k) \leq 1$, is defined as the fraction of vehicle type j density routed onto one outgoing link. Thus, we define a total control $v_i = \sum_j u_{i,j} x_{i,j}$.

As a result, the dynamics associated with a particular vehicle type have the form

$$x_j(k+1) = [A_j(k) + \sum_{i \in C_j} u_{i,j}(k)B_{i,j}(k)]x_j(k) + r_j(k) \qquad (2.3)$$

where C_j is the set of diverge links for vehicle type j and $r_j(k)$ is the vector of dispatch variables at the origin station for type j. We associate a vehicle type with each origin-destination pair although a destination-only formulation could be easily accomodated. The matrices, $A_j(k)$, $B_{i,j}(k)$, are functions of total density and total control variables, but we write (2.3) in this way to emphasize the bilinear form of the dynamics with respect to subnetwork variables. Moreover, $A_j(k)$, $B_{i,j}(k)$ are triangular in structure, that is, in terms of subnetwork variables each vehicle-type density variable, $x_{i,j}(k+1)$, only depends on the immediately upstream type j densities and $x_{i,j}(k)$. Both of these properties are used to simplify the optimization problem which we now discuss.

2.2 Optimization Problem

The performance index used is the total time averaged travel time (or delay) encountered by all vehicles. Thus, if $\tau(y_i)$ is the travel time on link i the total cost is

$$\sum_{k=1}^{K} \sum_{i \in \mathcal{L}} TD_i \tau_i(y_i(k)) y_i(k)/t_f \qquad (2.4)$$

where K is the number of time steps, $\mathcal{L}$ is the set of links and t_f is the final time.

Minimization of (2.4) subject to the flow constraints of the network is accomplished by applying duality. That is, the interconnection constraints of total density and total control are appended to (2.4) via Lagrange multipliers. As a result, the problem constraint dynamics decomposes into J vehicle type subnetwork constraints that are decoupled with respect to individual vehicle type density and routing control variables. The interconnection variables of total link density and total diverge link control are treated as additional control variables that act to couple the subnetwork dynamic constraints. As explained below, the principal benefit of applying duality is to produce a dynamic constraint structure that allows a completely decentralized control computation.

Because of the triangular structure of the subnetwork dynamics and the additive form of the cost function, the effects of each control variable upon the dual function objective can be computed locally. To accomplish this, the state equations are integrated in a distributed manner from origin to destination where each link only needs to obtain information from adjacent upstream links.

Similarly, information concerning future downstream congestion is obtained by integrating a subnetwork adjoint equation from destination to origin and backwards in time. Because the state equation is triangular and linear in state, the adjoint equation is triangular and independent of state. Thus, the adjoint equations are also integrated in a distributed manner where each link obtains information from adjacent downstream links.

Using state and adjoint information the algorithm is structured so at the link level, routing, total control, and total density variables are computed to minimize the dual function objective. The upper level coordinating control on each link computes Lagrange multipliers via subgradient optimization to maximize the dual function and satisfy the interconnection constraints that total density be equal to the sum of individual vehicle type densities and total control be equal to the sum of vehicle type controls. Consequently, the control algorithm could be implemented such that wayside control computers only need to communicate with neighboring computers to solve the overall dual problem.

Convergence of the algorithm is proven and demonstrated by simulation of a 4 station, 58 link network. It is shown that no duality gap exists and therefore, solution of the dual problem corresponds to solution of the original problem.

Several suboptimal strategies, more appropriate for actual implementation, are proposed. It is shown for the network example studied that significant savings in computational cost is obtained with little sacrifice of performance. Because the computational cost for the optimal control increases dramatically as time scale is

lengthened, for practical implementation, it is recommended that the suboptimal strategies be further developed.

3. PROBLEM DESCRIPTION

3.1 System Types

Two basic types of AGT systems have received consideration for high capacity urban travel. A Personal Rapid Transit (PRT) system employs small vehicles for single party (one to four passengers) service. Such systems will typically be required to operate at headways significantly below those of conventional transit systems in order to attain satisfactory levels of capacity, thus dictating operating headways in the range of 0.5 to 3.0 seconds. An Advanced Group Rapid Transit (AGRT) system is characterized by 12 to 24 passenger vehicles and operates in the range of 3 to 5 second headways.

In a PRT system a party enters a station, makes its destination known and requests service. An empty vehicle is provided and the destination is encoded on the vehicle which then departs into main line traffic. The vehicle is appropriately routed through the network to the destination station where passengers deboard. A demand actuated system of this type leads to the following problems [25]:

1. Selecting a vehicle to respond to a request for service at a station, e.g., assigning a vehicle from station storage, calling in a passing empty vehicle from the main line, or waiting for the next inbound vehicle to arrive.
2. Disposing of an empty vehicle at a station after its passengers have departed, e.g., placing vehicle in storage, assigning vehicle to another passenger, or dispatching the empty vehicle to another station.

3. Selecting a route for each vehicle (occupied or empty) once it has been dispatched.

In an AGRT system additional questions arise due to the multiple party occupancy of each vehicle:

1. Once a vehicle has been assigned to one party, will it wait for other passengers and for how long?
2. Will service to other passengers be limited to those going to the same destination or to certain destinations that are "along the same route"?
3. Will the vehicle be permitted to stop at stations along the way to pick up other passengers or possibly diverted for pick-ups?
4. Should transfers be permitted?

To select algorithms for solution of the above problems, we may formulate the management of an automated transportation network into a control problem. This first requires definition of system inputs, states, controls, constraints (including dynamics) and performance indices. We begin by characterizing the components which comprise the transportation network in terms of network geometry and demand models, followed by exposition of the various control problems associated with automated transit systems.

3.2 Network Model

The network geometry is described in terms of nodal elements and links. Nodal elements maybe classified into five basic types [25]. First, stations are points where passengers board and deboard vehicles, and are characterized by the number of berths, berth configuration,

length of the deceleration lane, and length of the egress lane into mainline traffic. Intersection nodes are points where two lanes merge or a single lane diverges. A yard designates the location of the entrance to empty vehicle storage facilities and maintenance area. The final type of nodal element is a link characteristic breakpoint where certain link characteristics such as speed, grade, and radius of curvature change. The guideway links of the network are assumed to be constructed within the public right-of-way, i.e., urban streets and highway systems and are specified according to directionality, speed, curvature, and grade. A link is considered to be one-way although two links of opposite direction may parallel one another.

Thus, network layout consists of station locations, station configuration, link connectivity and geometry. For this work, these parameters are assumed to be given as dictated by known and projected population distribution in conjunction with economic cost-benefit studies. Hence, with respect to network management, only size and location of vehicle storage facilities are considered design variables in the control problem.

To formulate the control problem we must relate the fundamental system output, link flows, to the system demands and the constraints imposed by the above network geometry. In the following section, the system demand model is first described. The modeling of network geometry constraints in terms of link flows is considered in Section 4.

3.3 Demand Model

The trip request model may be specified by level of demand, pattern of demand, arrival frequency, and diurnal variation of demand. Network demand description therefore has several characteristics [25]. First, the number of trips between various stations varies throughout the day. For example, morning trips are generally towards employment centers while midday demand may be between shopping areas. There is also variation in total trip demand as a function of the time of day. For downtown urban systems there is heavy morning and evening use while at activity centers such as airports and universities, the patterns are quite different.

Variations also occur in the rate at which passengers arrive at a particular station. For some stations, arrivals may be by car or walking and by bus at other stations. Thus, in the latter case large groups will arrive with large interarrival times and in the former case demand will be characterized by small groups with short interarrival times. Finally, the size of a party making a trip is of importance, particularly in GRT systems where several parties use the same vehicle and the possibility arises of splitting parties between vehicles.

Consequently, the demand for service at each station will vary in a random manner throughout the day and from day to day. In distributing occupied and empty vehicles, the ability to predict demand is likely to be advantageous. For this purpose, it is useful to note that demand variations will likely fall into one of three categories:

1. Recognizable average trends in demand such as morning

and evening rush hours,

2. Short-term random fluctuations in demand characterized by commonly occuring brief surges or lulls,
3. An occasional unanticipated long-term change due to, for example, sporting or cultural events.

The first two categories may be termed routine while the last may be known in advance but the precise time and extent is often not known.

In an attempt to predict demand changes the following kinds of data would be available for use:

1. Cumulative past experience such as daily trip records over past weeks or months,
2. Current states of stations in the system, that is, queues of waiting passengers and trip requests received.

To quantify demand for analysis, many investigators define a demand matrix, where each element, d_{ij}, is equal to the average demand from station i to station j in trip requests per unit time. Passenger interarrival time is then exponential distribution with mean, d_{ij}. In addition, d_{ij} may be a function of time to account for temporal variations in demand.

We now describe the various control problems associated with an automated transit system.

3.4 Control Problems

There are four major levels of control for an automated transportation system operating under non-emergency or non-failure conditions. They are:

1. Vehicle control

2. Merging

3. Station control including empty vehicle dispatch

4. Routing.

These problems are discussed, in turn, below.

3.4.1 Vehicle Control

Vehicle control requires the regulation of longitudinal and lateral motions of the vehicle during normal operation. Longitudinal control involves the adjustment of vehicle velocities and spacings within a string of several vehicles, the selected spacing policy thus affecting overall network performance. Lateral control, on the other hand, is only related to passenger comfort and safety, and therefore, will not be discussed further.

Two generic approaches to longitudinal control for short-headway AGT systems have received consideration in previous studies. They are point-following and vehicle-following, already described in the Introduction.

For a vehicle-following system, a typical situation is depicted in Fig. 3.1.

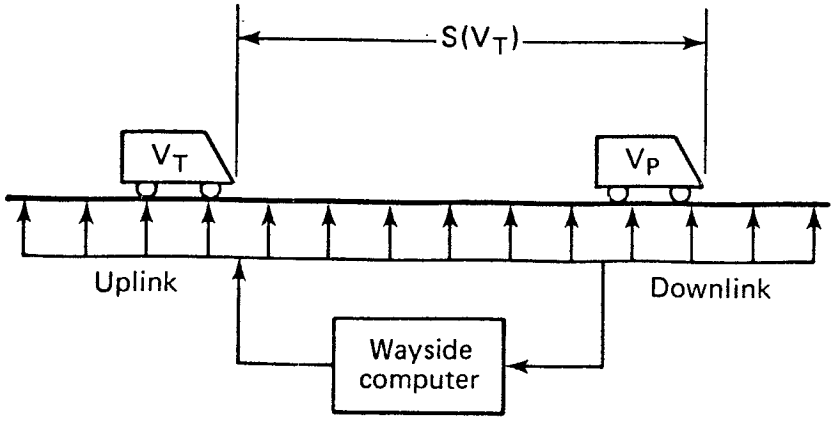

Fig. 3.1 Vehicle - follower control.

A wayside computer determines the position of each vehicle within its jurisdiction (i.e., guideway section) by position markers, such as induction coils imbedded in the guideway. Vehicle velocities are also downlinked through communication lines to the wayside computer. A vehicle-follower control law, generally based on the spacing and relative velocity between a preceding and trailing vehicle, generates acceleration commands to the trailing vehicle in order to maintain a specified spacing policy, $S(v_T)$. As discussed below most spacing policies are a function of the trailing vehicle velocity, v_T. When the spacing between vehicles is large, a trailing vehicle operates in an open-loop velocity command mode. When vehicle spacing crosses some threshold value, the vehicle must account for the presence of a preceding vehicle and a transition to the spacing policy, $S(v_T)$, is initiated.

Several spacing policies, $S(v_T)$, have been suggested [6]. A constant spacing policy, employed in point following, maintains a constant nose to nose spacing between vehicles. A constant time headway maintains a constant time between vehicles passing a fixed point on the guideway. A third spacing policy defines a K-factor as the ratio of vehicle separation (tail to nose) to emergency stopping distance. Thus, this represents a "brickwall" stopping criterion where if a vehicle is operating with a K-factor greater than one, the trailing vehicle may come to a stop without collision if the preceding vehicle becomes a brickwall. Both constant headway and constant K-factor policies may be used in either point-following or vehicle-following systems. A final policy designed for vehicle-following systems is a kinematic spacing [7] scheme where vehicle

spacing is based on a relative stopping distance. This policy arises in a vehicle following system because a trailing vehicle cannot anticipate future deceleration maneuvers of a preceding vehicle (due to downstream congestion). Thus, a trailing vehicle must maintain a spacing such that if the preceding vehicle suddenly brakes on limits to a stop, the trailing vehicle can respond without collision.

Many studies of vehicle-following have focused on the regulation of vehicle speeds and spacings during perturbations about nominal values as determined by one of the first three policies described above. However, in general, a control capability must exist to perform transient maneuvers such as overtaking a slower moving vehicle, switching from an open-loop velocity command mode to a closed-loop regulation mode, merging vehicles both on the main line and from stations, and generating gaps in vehicle flows during such merging operations. A control law based on a kinematic spacing policy is suitable for all of these maneuvers as reported in [7] and will be used to model discrete vehicle motion in the simulation work.

The next level of control is vehicle merging.

3.4.2 Merge Control

Many merging techniques have been studied in terms of the effect upon vehicle delay at a junction given various types of input flows. The algorithms which have been studied may be classfied into one of two general categories. First, synchronous methods are those where all vehicles are allocated to slots which move along the guideway so that no conflicts occur in the system, thus requiring a central

controller. Also note these methods would be associated with a point-following system. A second approach is to employ local control where at a given distance upstream from the merge junction, a merge controller commands vehicle actions to assure a successful merge of two vehicle strings. This may be accomplished in two ways: (1) movements of vehicles are controlled by the relative movements of other vehicles on the guideway (vehicle-following); or (2) vehicles are assigned slots so that merging occurs without conflict. Because a distributed control algorithm is sought the synchronous method will not be considered. The local strategy may be broken down into a number of schemes depending on how priorities are assigned. The results of studies comparing various schemes are now discussed.

The amount of literature concerning merge strategy using a vehicle follower approach is rather limited. The only work to address the problem analytically is by Athans [26] where he collapses the two incoming merge lanes into a single lane and then considers a string of "non-crashing" vehicles. A control law is designed using a quadratic performance criterion for the string of vehicles. For each possible merging sequence, the value of the associated performance index may be calculated and thus, the merging sequence with the lowest cost is selected. The major drawbacks of this approach is that (1) all vehicles in the string require information of every other vehicle in the string, (2) in a high density network the number of possible sequences is high and the required computation becomes excessive, and (3) the algorithm does not consider resulting delay.

A first come, first serve algorithm is simulated by Brown [27]

where individual vehicles through a merge are simulated using a linear vehicle-follower control at a 4.0 sec headway. This approach appears to be a good candidate for a merging algorithm because of its simplicity for implementation and the fact that it tends to treat all vehicles equally. This latter result has been shown by several investigators in the point following case discussed below. To implement first-come, first-serve a parallel data region upstream from the merge junction is selected such that a merge controller has knowledge of all vehicles within this region. Each vehicle entering the region is then assigned to follow the last vehicle to have entered the region whether it be a vehicle immediately in front or a vehicle on the alternate guideway.

There is more literature available concerning merge control using point-following, most likely because of its analytic tractability. The most basic formulation of the problem is given by Whitney [28] where incoming streams of vehicles are described by a binary word with a 1 representing an occupied slot and a 0 representing an empty slot. A terminal cost is computed by squaring the net number of slots moved by each vehicle, and summing over all vehicles moved. In [29], it is proved that a first-come, first-serve scheme minimizes the cost.

The most detailed investigation of merging for a point following system is contained in the thesis by Godfrey [30] where he evaluates six merging strategies based on queueing models and simulation. He also concluded the first-come, first-serve strategy to be consistently the best strategy by measuring merge effectiveness in terms of tail probabilities (i.e., Prob [Delay > N slots]).

In [29], Sakasita simulates Godfrey's six merging strategies, compares resulting queue lengths of each merge lane separately, and comes to the same conclusion concerning the desirability of first-come, first-serve. Some specific algorithms and problems which arise at intersections are considered in [31] and [32].

3.4.3 Station Control

The impact of station design upon network performance depends upon the station guideway configuration and the station operating policy ([33], [34], [35]). The configuration is the actual physical layout of the station including deceleration and accelerating ramps, number of berths, and layout of the docks. The station operating policies involve the management of vehicles once they have entered the station, the unloading and loading of passengers, and the merging of vehicles onto the mainline. It will be assumed that a station design has been performed to efficiently handle expected demand at that location; the only vehicle management design variable being the handling of empty vehicles and the dispatching of occupied vehicles.

To aid in station modeling we note the events which occur [34] when a vehicle enters a station:

1. Time to switch off guideway
2. Deceleration time
3. Move to dock from deceleration ramp
4. Open doors
5. Unload time
6. Load time

7. Close doors

8. Dispatch queueing time

9. Move from dock to acceleration ramp

10. Acceleration time

11. Switch onto main guideway.

The above sequence may be altered when, for example, a vehicle unloads and then advances to the first unoccupied berth to await passengers for loading. The loading and unloading times are random variables and have been modeled [34] using a log-normal distribution. However, this may also be represented as a deterministic minimum dwell time which is normally expected to be sufficient for all loading and unloading of passengers.

The most important aspect concerning stations is the possibility that there is no room for an arriving vehicle and it must be rejected. The vehicle would then be routed to the nearest station or looped around until a space is free. Alternatively, we may allow "backups" onto the main guideway until space is free.

Finally, in a demand actuated AGT system an essential feature of vehicle management is to provide empty vehicles at stations to serve trip requests [36]. Two questions which naturally arise are where to obtain the empty vehicles and how to efficiently dispatch the vehicle to the station. The first question deals with the arrangement of vehicle storage facilities while the latter is associated with vehicle routing.

The possibilities for vehicle storage facilities are as follows [36]:

1. Each station can have its own storage

2. Network divided into districts, each district having a common storage

3. Moving storage with vehicles circulating through a district ready to be called upon.

Many factors including economic cost and real estate enter into the selection of an appropriate arrangement for vehicle storage facilities. This selection process would be aided by comparing network performance in terms of figures of merit (discussed below) for a given vehicle management strategy and various facility configurations.

The final level of control is that of vehicle routing, the major problem of interest in this work. A review of literature was given in the Introduction. We now describe the general approach to the problem.

3.4.4 Routing

In the previous sections we have described the local control problems of vehicle control, merging, and station control. We now consider the combined problem of network control, illustrated by the network section in Fig. 3.2. Passengers arrive at the station loading dock and board a vehicle that has either just unloaded arriving passengers or has been withdrawn from vehicle storage. The vehicle merges into mainline traffic and is controlled by a series of local wayside computers that communicate with the vehicle through sensors imbedded in the guideway. Each wayside computer has jurisdiction over a region of the guideway network and passes control to adjacent wayside computers as the vehicle passes through jurisdictions.

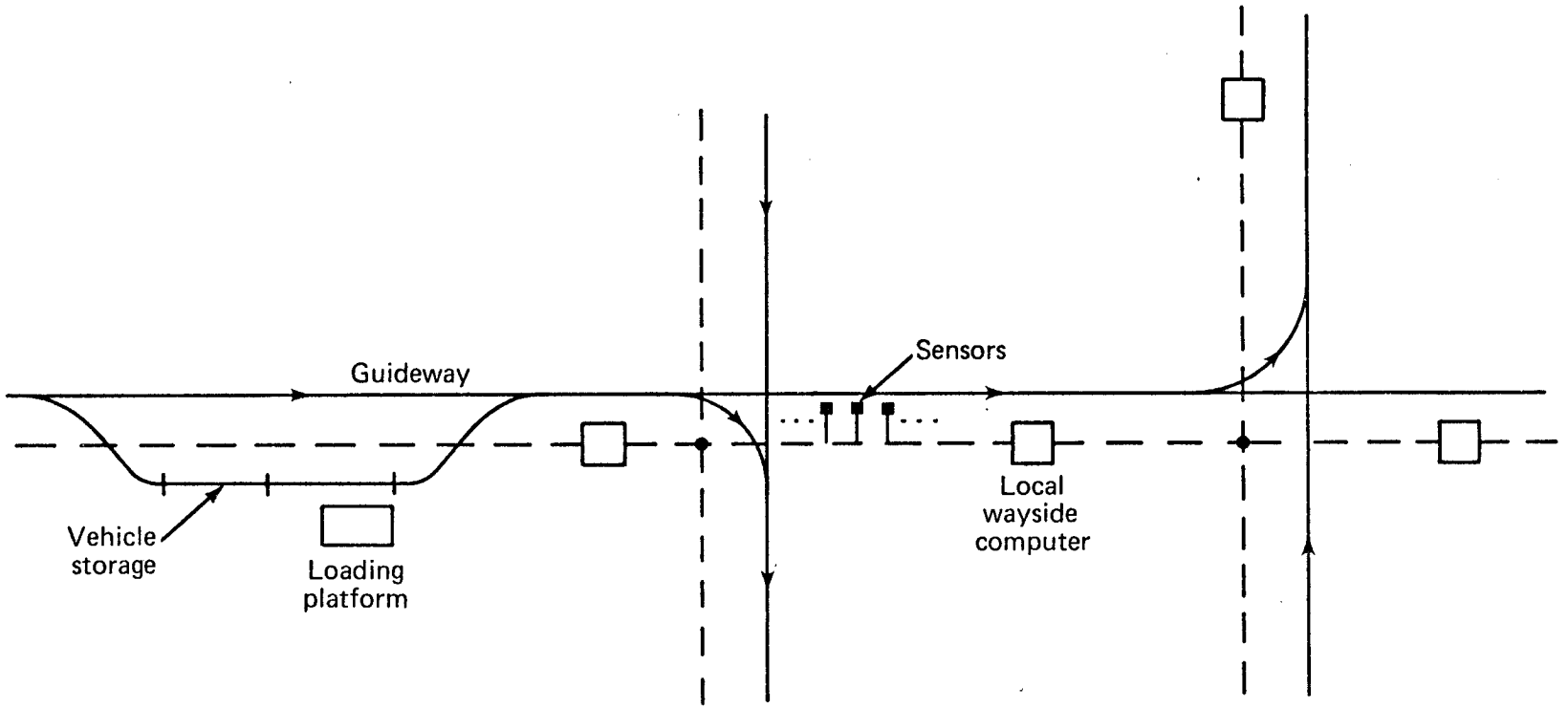

Fig. 3.2 Network section.

As we have already indicated, in a large network it may be desirable to make routing decisions on-line and locally. That is, as each vehicle approaches a diverge link the local wayside computer makes the decision as to what outgoing link the vehicle should travel. In addition, the local wayside computer makes this decision based on information obtained from the adjacent wayside computers. Thus, using this approach we utilize the communication links already in place for the vehicle control problem and no additional links are required.

It might be suggested that because we have a completely automated system, we only need information at stations. That is, all behavior within the system is deterministic and all control policies are initiated in response to trip demands. However, any practical system will have occasional failures, requiring a re-start within the network and therefore, an implementation such as that in Fig. 3.2 is needed.

The above discussion implies a multi-level structure to the control of an automated transportation network where the levels of control are interacting. The effects of local vehicle control, merge control, and distributed routing must be assessed in terms of network performance. To quantify this performance several figures of merit may be employed as we now describe.

3.4.5 Figures of Merit

In describing figures of merit for a transportation network we first note the distinction between a system optimizing index and a user optimizing index. System optimization involves assigning

a cost to each section based on system variables such as flow or delay and then summing over all sections in the guideway network. User optimization assigns costs associated with origin to destination pairs.

System optimizing performance measures would include:

1. Total Delay - average all link delays where delay is defined as either (1) time to traverse a link, or (2) extra time to traverse link due to congestion, and is weighted by the number of vehicles experiencing delay.
2. Total Service [12] - compute the product of the number of vehicles on each link and the average velocity on that link; sum over all links and integrate over a time interval T; the result is the total distance moved by vehicles in the network during time interval T.
3. Energy Use - vehicle accelerations, number of empty vehicles traveling on network, fleet size (i.e., total number of vehicles required to fulfill a given level of demand).

User optimizing performance measures would include:

1. Average origin to destination delay where delay is the time beyond the time required for an unimpeded trip (including wait time)
2. Maximum origin to destination delay
3. Dependability - variance in origin to destination delay at a given time of day.

To complete a description of the control problem, we conclude with a discussion of the information that would be available through sensor measurement.

3.4.6 Sensors

The fundamental information needed for control is position and velocity. A multitude of systems have been conceived to measure these quantities, but at this time the most technically feasible type of sensor appears to be one which is imbedded in the guideway to measure vehicle position. As a vehicle passses over a sensor (e.g., inductive loop) this event is recorded to determine vehicle position with respect to a reference. The spacing of these sensors thus determines the accuracy of the position measurement. Vehicle velocity may be measured by a tachometer on-board the vehicle or estimated from the position measurements. Using these measurements, vehicle control computation is performed either at wayside, on-board or with an appropriate allocation between wayside and vehicle. The necessary communication links are generally through the guideway by inductive coupling.

Other possible sensing elements would be:

1. Measure demand at stations,
2. Measure time delay to traverse a link by clock aboard vehicle,
3. Measure number of vehicles on a link with up-down counter.

3.5 Problem Scope

Section 3 has been devoted to an overview of the normal operational control problems involved with the design of an automated transportation system. This is done to place in context the more limited scope of the present research work, although it is intended that the results of this work may be suitably modified and

extended to encompass associated problems for purposes of comparison and performance trade-off studies.

Specifically, a vehicle-follower control policy will be considered primarily because of its wider applicability to traffic problems, and the general lack of studies in this area for automated systems. We will also consider a PRT system with single party vehicle occupancy because of its greater simplicity, a useful asset for an initial study. The network geometry is assumed to be a general connection of links, diverges and merges; note this includes intersections by an appropriate combination of diverges and merges. Finally, station configuration is assumed to be a single, off-line spur where vehicles are both queued for arrival and departure, and stored. A detailed model of the station is given in Section 4.2.3.

4. TRAFFIC FLOW MODELS

The major component describing the behavior and characteristics of a transportation network is vehicle flow. This flow is determined by the operational policies (e.g., intervehicle spacing control scheme), station geometries, and link connectivities (merges and diverges) discussed above. The accurate modeling of vehicle flow and the attendent description of network dynamics is critical to a successful control law design. Therefore, this section discusses in detail, the derivation of vehicle-follower traffic models and the simulation of these models to evaluate their effectiveness in representing actual traffic flow. We begin with the basic flow model drawn from analogy to fluid flow and then derive a specific vehicle-follower model at various levels of complexity.

4.1 Fundamental Link Flow Model

In modeling the flow of traffic on a given guideway link, there are three quantities of interest: vehicle density (vehicles/unit length), flow (vehicles/sec), and velocity (m/sec). Each quantity is a function of time, t, and position, s, along the guideway link. The fundamental relationship between flow and density is derived by considering conservation of flow across an element ds (Lighthill and Whitham, [1]). The resulting continuity equation is

$$\frac{\partial x}{\partial t} + \frac{\partial q}{\partial s} = 0$$

where $x(s,t)$ is vehicle density and $q(s,t)$ is vehicle flow.

Discrete forms (Tabak, [37]) of the above model are obtained by considering guideway sections of length D_i with no more than one entrance link and one exit link as shown in Fig. 4.1.

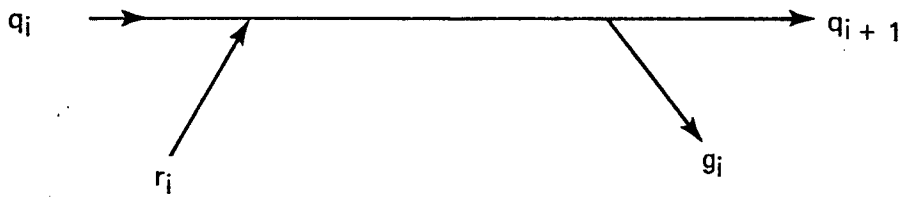

Fig. 4.1 Guideway section.

The continuity equation, discretized in space, becomes

$$\frac{dx_i}{dt} = [q_i - q_{i+1} + r_i - g_i]/D_i$$

The modeling problem is two-fold. First, we need to represent what occurs when we concatenate links together, and into junctions. That is, the flow, q_i, into link i will be a function of the state of link i and link i-1. To derive an aggregate model, we also need to determine this function without the detailed knowledge of every vehicle's position and velocity. Second, we must establish the relationship between the fundamental quantities of density, flow, and velocity for a string of discrete vehicles. We first consider the latter problem.

For vehicles operating under vehicle-following, a steady state spacing policy is employed such as a constant time headway. For example, using constant headway a string of vehicles with velocity v is spaced by hv+L where h is the desired headway in seconds and

L is vehicle length. Consequently, the vehicle density, x(t), is given by

$$x(t) = 1/(hv(t) + L) \tag{4.1}$$

and the corresponding flow is

$$q(t) = x(t)v(t) = v(t)/(hv(t) + L) \tag{4.2}$$

Through (1) and (2) we may determine the vehicle flow in terms of the vehicle density, or

$$q(t) = (1 - x(t)L)/h \tag{4.3}$$

Note that (4.3) neglects the vehicle-follower control dynamics, that is, some control law on vehicle position, velocity, and acceleration is used to regulate to the desired spacing, hv, giving a dynamic transfer between vehicle spacing and vehicle position rather than the static relationship assumed in (4.1). Models that include vehicle-follower control dynamics are derived in Section 4.3. First, we will derive a simpler model that describes flow connecting adjacent links and assumes the static relationship, (4.1).

4.2 Vehicle-Follower Network Model

The basic network elements to be modeled are a link, a merge, a diverge and a station. We will assume a constant headway control policy of h sec, a maximum allowable velocity $v_{max}(i)$ on the link i and a vehicle length, L.

4.2.1 Link Model

A guideway link section, i, is represented in Fig. 4.2. The rate of change of density on link i is given by the flow into link i minus the flow out of link i divided by its length, D_i. Thus, letting x_i denote the density on link i we have

$$\dot{x}_i = [q_i - q_{i+1}]/D_i \quad . \tag{4.4}$$

For simplicity of notation the time subscript, t, is dropped wherever it is apparent that quantities are functions of time (e.g., density, velocity, flow).

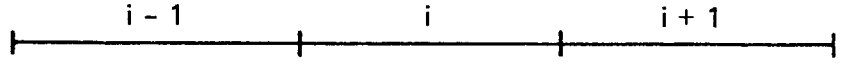

Fig. 4.2 Link section.

We now propose that the flow q_{i+1} is given by

$$q_{i+1}(x_i, x_{i+1}) = f_1(x_i) f_2(x_{i+1}) \tag{4.5}$$

where

$$f_1(x_i) = \begin{cases} x_i/y_m(i) & x_i \le y_m(i) \\ 1 & y_m(i) \le x_i \le 1/L \end{cases}$$

$$f_2(x_i) = \begin{cases} q_{max}(i) & x_i \le y_m(i) \\ (1 - x_i L)/h & x_i \ge y_m(i) \end{cases}$$

$$y_m(i) = 1/(hv_{max}(i) + L)$$

$$q_{max}(i) = y_m(i)v_{max}(i) \quad .$$

The quantity, y_m, is the density at maximum flow, which occurs at the maximum velocity for a constant headway system. It also represents the threshold density value at which vehicle following is initiated. That is, when densities are below y_m it is assumed that vehicles are operating in an open-loop velocity command mode and are therefore, decoupled. The functions f_1, f_2 are represented graphically in Fig. 4.3. Note the flow function, (4.5), depends on the physical characteristics of links i and i+1 through specification of $v_{max}(i)$, which is typically at a lower value for curved links.

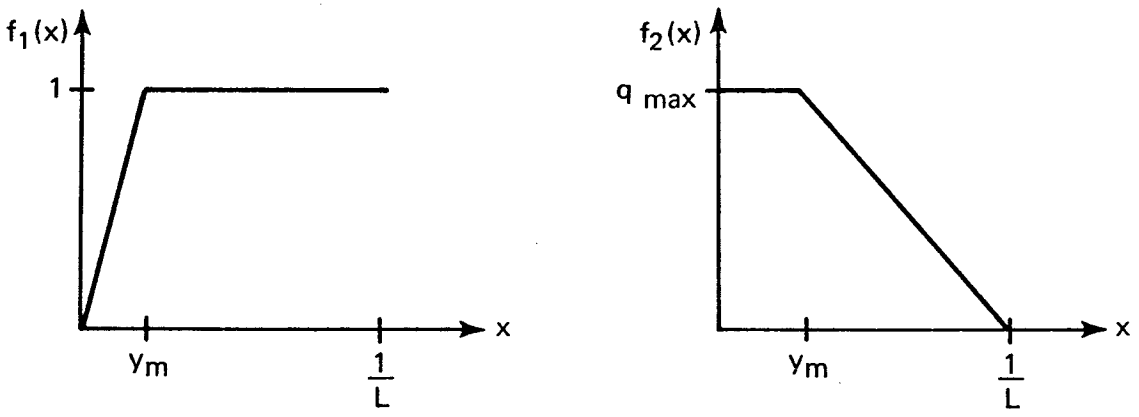

Fig. 4.3 Flow functions.

The rationale of the above model is that when the density on link i-1 exceeds y_m then the input flow is restricted by the flow on i, that is, there are enough vehicles on i-1 to fulfill the flow allowed into link i as determined by the density on link i. When

the density on link i-1 is below y_m the flow is $x_{i-1}v_{max}$ for $x_i \leq y_m$, and is reduced as x_i exceeds y_m.

The simplest model which we will term Model 1 is therefore given by

$$\dot{x}_i = [q_i(x_{i-1},x_i) - q_{i+1}(x_i,x_{i+1})]/D_i \tag{4.6}$$

Note that link capacity constraints are implicit in the model. To evaluate the performance of a network control algorithm it is important that delay be accurately represented for each link. The delay on link i may be computed as a function of density by solving (4.1) for vehicle velocity when $x_i > y_m(i)$. As a result we have

$$\tau_i = \begin{cases} hx_iD_i/(1-x_iL) & x_i > y_m(i) \\ D_i/v_{max}(i) & x_i \leq y_m(i) \end{cases} , \tag{4.7}$$

The models for the merge, diverge, and station are straightforward extensions of the link flow model.

4.2.2 Merge Model

A merge junction be modeled by two links joining into a common merge region where vehicle assignment and merging takes place. This is shown schematically in Fig. 4.4, the two actual merge region links being collasped into a single link.

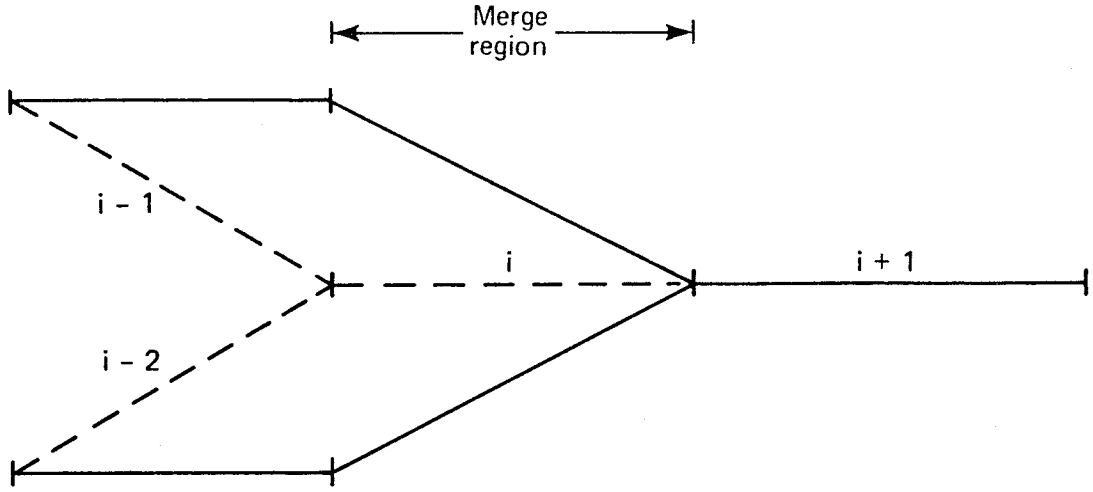

Fig. 4.4 Merge model.

The density on link i may be modeled by summing the flows determined by the i-1, i boundary and the i-2, i boundary. As a result, we have

$$\dot{x}_i = [q_i(x_{i-1},x_i) + q_i(x_{i-2},x_i) - q_{i+1}(x_i,x_{i+1})]/D_i \quad . \quad (4.8)$$

4.2.3 Diverge Model

A diverge is represented by the geometry illustrated below in Fig. 4.5.

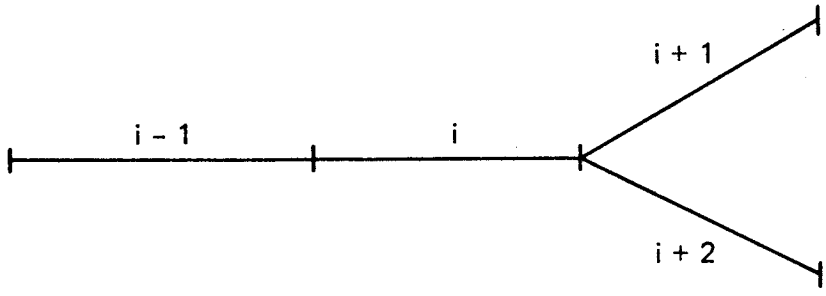

Fig. 4.5 Diverge model.

We define a control variable, u_i as the fraction of vehicles on link i to be routed onto link i+1; hence, the fraction routed onto i+2 is $1-u_i$. The corresponding flows are $q_{i+1}(u_i x_i, x_{i+1})$ and $q_{i+2}((1-u_i)x_i, x_{i+2})$. Again, extending the basic link model we have

$$\dot{x}_i = [q_i(x_{i+1},x_i) - q_{i+1}(u_i x_i, x_{i+1}) - q_{i+2}((1-u_i)x_i, x_{i+2})]/D_i \quad (4.9)$$

4.2.4 Station Model

The impact of station design upon network performance depends upon the station guideway configuration and the station operating policy (Sirbu [34]). The configuration is the actual physical layout of the station including deceleration and accelerating ramps, number of berths, and layout of the docks. The station operating policies involve the management of vehicles once they have entered the station, the unloading and loading of passengers, and the merging of vehicles onto the mainline.

The station model considered here consists of an off-line spur containing a vehicle arrival queue that also serves as an empty vehicle storage area for arriving trip requests. For simplicity we assume that passengers disembark and load at a single platform, once loaded each vehicle enters an egress lane queue and then merges into mainline traffic. The model may be extended to include multiple loading berths. The model is illustrated in Fig. 4.6.

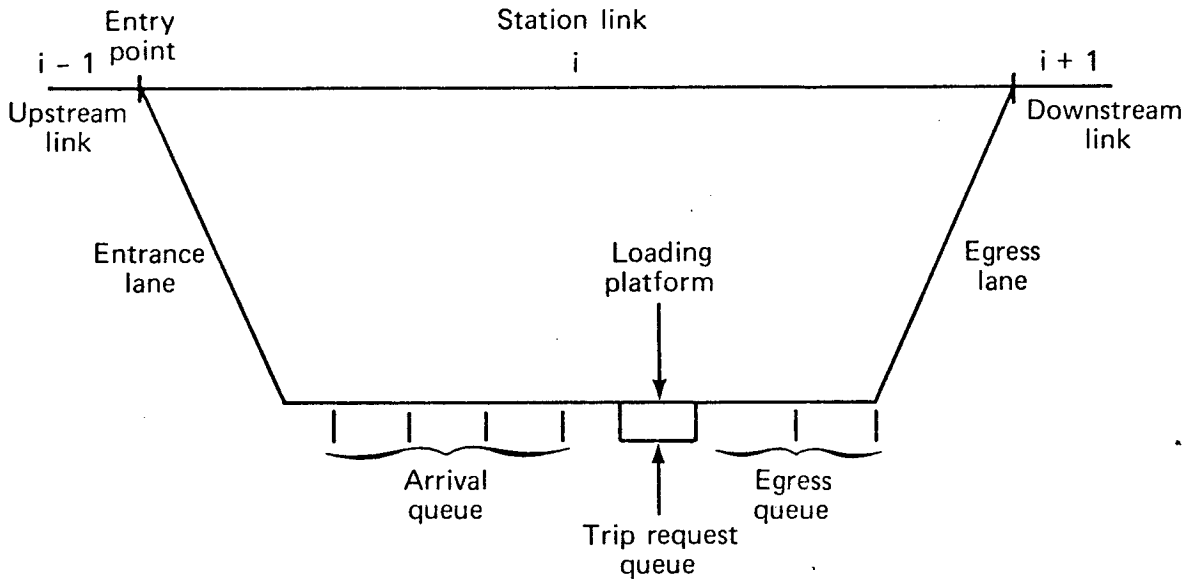

Fig. 4.6 Station model.

Each station is associated with a link i where

q_a = allowable arrival flow

n = number of stored vehicles (stored vehicles = number of vehicles at loading platform plus number in the arrival queue)

n_{max} = maximum number of vehicles that can be stored at station i

r = new trip request rate (trips/unit time)

w = number of waiting trips

t_d = minimum dwell time per vehicle at station platform (total unload and load time, designed to be fixed for the majority of circumstances).

$$u_d = \begin{cases} 1 \text{ if trip is dispatched} \\ 0 \text{ if } n=0,\ x_e = x^e_{max}, \text{ or otherwise} \end{cases}$$

x_e = vehicle density on egress lane

D_e = length of station egress queue

D_i^e = length of egress merge section

x_{max}^e = maximum vehicle density allowed on vehicle egress lane

a_s = service deceleration rate.

The allowable arrival rate, q_a, is defined in a manner analogous to the previous models for link flow. That is, for low vehicle densities the flow is constrained by some maximum rate and falls off to zero when vehicle storage is filled. As vehicle storage fills, the distance to decelerate to a stop on the station entry lane decreases as shown in Fig. 4.7.

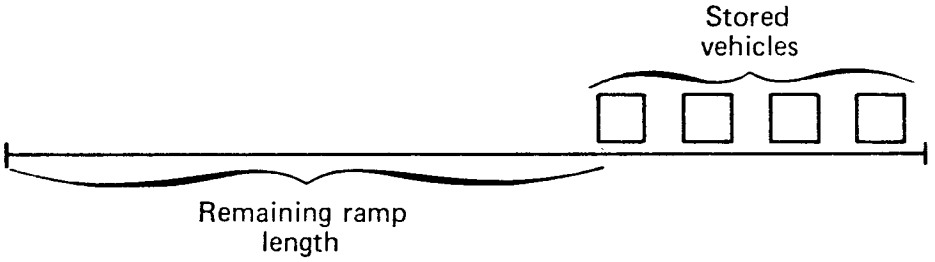

Fig. 4.7 Station entrance ramp.

It will be assumed that a vehicle may begin decelerating on the main line as it enters the station. Consequently, the allowable flow at the station entry point is determined by the velocity a vehicle must have to successfully stop at the end of the arrival queue. This velocity is given by

$$v_a = (2a_s(n_{max}-n)L)^{\frac{1}{2}} \tag{4.10}$$

and the corresponding allowable flow is

$$q_a = \frac{v_a}{hv_a + L} \quad . \tag{4.11}$$

The equation for the density of vehicles entering the station from upstream link, i-1, is given by

$$\dot{x}_{i-1} = [q_{i-1}(x_{i-2}, x_{i-1}) - f_1(x_{i-1})q_a]/D_{i-1} \tag{4.12}$$

where $f_1(x_{i-1})$ is defined in (4.5) and q_a is defined in (4.10), (4.11). Note we have assumed that all vehicles destined for station i are diverted into the station. Hence, backups onto the mainline are allowed when station capacity is exceeded. This is a policy decision that would be determined by judgement and network analysis. The above model (4.12) may be easily modified to treat the upstream link, i-1, as a diverge link and thus allow station bypasses.

The accumulation of stored vehicles is given by

$$\dot{n} = f_1(x_{i-1})q_a - u_d/t_d \tag{4.13}$$

where u_d determines if a vehicle is being dispatched and u_d/t_d is the flow of vehicles into the egress lane. Note that t_d is a minimum dwell time (or $1/t_d$ is a maximum flow). Longer dwell times are represented by setting $u_d = 0$ if a vehicle is idle at the dock berth.

To describe egress lane dynamics, it is necessary to consider multiple vehicle types, that is, vehicles destined for stations other than station i traveling on link i. As a vehicle departs

from the loading berth it enters the egress lane queue where it is assigned to follow a vehicle on the mainline (link i). Thereafter, it merges into mainline traffic according to the same control policy as a mainline merge junction. Thus, the egress lane may be modeled in two parts. The first consists of the injection of vehicles from the dock berth into the egress queue. The second is the flow of vehicles into the merge section, collapsed onto the station link as in the merge model. This is illustrated in Fig. 4.8.

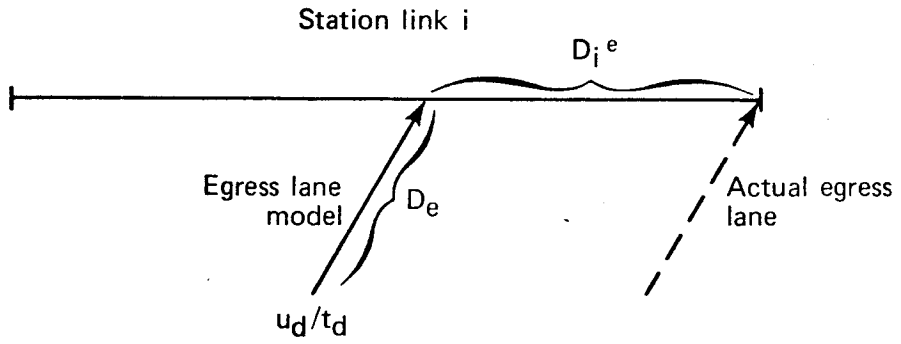

Fig. 4.8 Egress lane model.

The flow into the egress lane is given by u_d/t_d while the flow out of the egress lane depends on the density of link i so that the egress lane density is determined by

$$\dot{x}_e = [u_d/t_d - q_i(x_e, x_i)]/D_e \tag{4.14}$$

The density of egressing vehicles merging into link i on section D_i^e, denoted by x_{ie}, is given by

$$\dot{x}_{ie} = [q_i(x_e, x_{ie}) - q_i(x_{ie}, x_{i+1})]/D_i^e \qquad (4.15)$$

where i+1 is the downstream link and i is the mainline station link. If station bypasses are allowed, (4.15) would be modified to account for vehicles entering from the upstream station link.

Finally, the queue of waiting trips is given by

$$\dot{w} = r - u_d/t_d \quad . \qquad (4.16)$$

The delay on link i is computed using (4.7). The delay on the arrival link is computed by

$$\tau_a = nt_d + v_a/a_s \quad .$$

The delay on the egress lane is computed in a different manner from previous cases by taking into account that vehicles are accelerating from a stop under congested conditions. (Note on the arrival lane that vehicles always come to a stop, unimpeded by traffic). We first compute the maximum density of vehicles in the egressing region assuming they accelerate from a stop and are operating under vehicle following. The velocity a vehicle attains after a distance, s, with constant acceleration is

$$v(s) = (2a_s s)^{\frac{1}{2}}$$

and the corresponding vehicle density is

$$x(s) = 1/(hv(s) + L) \quad .$$

We set $v(D_i^e) = v_{max}$. Thus, the maximum density of vehicles allowed in the egress region is

$$x_{max}^e = \frac{1}{D_i^e} \int_o^{D_i^e} x(s)ds \quad .$$

We now assume that the delay in the egress region is determined by the density, x_i, since egressing vehicles must merge with vehicles on the station link, i. For densities below x_{max}^e it is assumed that vehicles can egress with acceleration a_s (i.e., unimpeded). For densities above x_{max}^e, we compute the vehicle velocity corresponding to a density, D_i^e. Consequently, the delay is computed as

$$\tau_e = \begin{cases} 2D_i^e/v_{max} + \delta & x_i \leq x_{max}^e \\ v_e/a_e + \delta & x_i > x_{max}^e \end{cases}$$

where

δ = fixed delay for vehicle to advance from loading dock to egress queue

$v_e = ((1/x_i)-L)/h$

$a_e = v_e^2/(2D_i^e)$.

4.2.5 Multiple Destinations

The above equations do not account for the various destinations of the vehicles on a section, a factor that must be included in order

to develop a routing strategy. Consequently, we may modify the above equations by attributing the flow of a particular vehicle type to the fractional portion of that vehicle type on a guideway section. Thus, if we define

$$\begin{aligned} x_{i,j} &= \text{density of vehicle type } j \text{ on link } i \\ y_i &= \sum_j x_{ij} = \text{total density on link } i \end{aligned} \tag{4.17}$$

then the flow due to vehicle type j leaving link i is

$$q_{i+1}(j) = (x_{i,j}/y_i)q_{i+1}(y_i, y_{i+1}) \tag{4.18}$$

At this point, we will express the dynamic equations for the network in discrete time, being consistent with the formulation in the following sections. Using (4.18) we define the velocity function at time k as $a_{i,i+1}(k) = q_{i+1}(y_i(k), y_{i+1}(k))/y_i(k)$. The equation for a link is then

$$x_{i,j}(k+1) = x_{i,j}(k) + T[a_{i-1,i}(k)x_{i-1,j}(k) - a_{i,i+1}(k)x_{i,j}(k)]/D_i \tag{4.19}$$

and a merge is

$$x_{i,j}(k+1) = x_{i,j}(k) + T[a_{i-1,i}(k)x_{i-1,j}(k) + a_{i-2,i}(k)x_{i-2,j}(k) - a_{i,i+1}(k)x_{i,j}(k)]/D_i \tag{4.20}$$

For a diverge, we define the total control variables

$$v_{i,i+1} = v_i = \sum_j u_{i,j} x_{i,j} \tag{4.21}$$

$$v_{i,i+2} = v_i' = \sum_j (1-u_{i,j}) x_{i,j} \tag{4.22}$$

and associated flows $a_{i,i+1} = q_{i+1}(v_i, y_{i+1})/v_i$, $a_{i,i+2} = q_{i+2}(v_i', y_{i+2})/v_i'$. The corresponding diverge equation is

$$x_{i,j}(k+1) = x_{i,j}(k) + T[a_{i-1,i}(k) x_{i-1,j}(k) - (u_{i,j}(k) a_{i,i+1}(k) + (1-u_{i,j}(k)) a_{i,i+2}(k)) x_{i,j}(k)]/D_i \tag{4.23}$$

A complete listing of all equations for the network is given in Appendix A. Note the above dynamics have a relatively simple form if the total density variables, y_i, and total control variables, v_i, are treated as independent variables and not a function of state. That is, the dynamics are bilinear in the state and the control, and triangular in structure. This structure will be exploited in the problem formulation discussed in Section 5.

The next section derives a more detailed model based on specific vehicle follower control dynamics.

4.3 Vehicle Follower Control Model

A vehicle follower control law will generally base acceleration commands on the relative motion between a preceding and trailing vehicle. For example, a control law designed by Pue [7] is approximately

$$a_c = \frac{dv_i}{dt} = - ghv_i + g\left(\frac{a_s}{2j_s} + \frac{v_{max}}{2a_s} - h\right)(v_{i+1} - v_i)$$

$$+ g\frac{v_i}{2a_s}(v_{i+1} - v_i) + g(S-L) \qquad (4.24)$$

where a_c = acceleration command

v_i = velocity of the i^{th} vehicle

v_{i+1} = velocity of vehicle preceding the i^{th} vehicle

S = yehicle spacing

h = headway

g = controller gain

a_s = service acceleration limit (1.5m/s)

j_s = service jerk limit (2.0 m/s)

v_{max} = maximum line speed (15 m/s).

The control law in (4.24) neglects several nonlinearities due to a vehicle being on a jerk or acceleration limit. In addition, if we neglect vehicle propulsion dynamics then vehicle acceleration is $a_c = dv_i/dt$.

Now consider a vehicle string to form a continuum with vehicle velocity and density a continuous function of time and position along a guideway link. Then substituting $1/x(s,t)$ for vehicle spacing, (4.24) becomes

$$\frac{dv(s,t)}{dt} = c_1 v(s,t) + (c_2 + c_3 v(s,t))\,[v(s+1/x(s,t),t) - v(s,t)]$$

$$+ c_4((1/x(s,t)) - L) \qquad (4.25)$$

where the gains, c_1, c_2, c_3, c_4 are given by the corresponding quantities in (4.24). For a congested link we may approximate the

"preceding" vehicle velocity, $v(s+1/x(s,t),t)$ by

$$v(x+1/x(s,t),t) \simeq v(s,t) + (\partial v(s,t)/\partial s)/x(s,t) \quad . \tag{4.26}$$

Substitution of (4.26) into (4.25) gives

$$dv(s,t)/dt = c_1 v(s,t) + (c_2+c_3 v(s,t))\,(\partial v(s,t)/\partial s)/x(s,t)$$

$$+ c_4((1/x(s,t)) - L) \quad . \tag{4.27}$$

Using $dv(s,t)/dt = \partial v(s,t)/\partial t + v(s,t)\,(\partial v(s,t)/\partial s)$ with $x = x(s,t)$ and $v = v(s,t)$, (4.27) becomes

$$\partial v/\partial t = c_1 v + ((c_2+c_3 v)/x - v)\,(\partial v/\partial s) + c_4((1/x) - L) \quad . \tag{4.28}$$

If we now discretize in space and time over a guideway section of length, D_i, and time interval T then

$$v_i(k+1) = v_i(k) + T\{[-v_i(k) + (c_2+c_3 v_i(k)/x_i(k)][v_{i+1}(k)-v_i(k)]/D_i$$

$$+ c_4((1/x_i(k)) - L) + c_1 v_i(k)\} \quad . \tag{4.29}$$

where $v_i(k)$ and $x_i(k)$ are the velocity and density of link i at time, k.

The model (4.29) appears similar to that used by Payne [12] for traffic flow. In Payne's model a desired velocity that is a function of density is used while controller dynamics is modeled

by an average driver reaction time. Note the above derivation may be applied to any longitudinal control law.

As noted earlier, a major problem in modeling the transportation network is to describe the flow connecting adjacent links. As seen in (4.19), the above model provides some interconnection through v_{i+1}, but as found in simulation, results in too much "averaging" and tends to smooth out bunchy input flows. The models proposed in Section 4.2 appear to mitigate this problem when we combine the above with the flow functions defined in (4.5). As a result, we have the model, termed Model 2,

$$v_i(k+1) = v_i(k) + gT\{-hv_i(k) + (1/x_i(k)) - L$$

$$+ [v_{i+1}(k) - v_i(k)][-v_i(k) + (5.3-h + .33v_i(k))]/D_i\}$$

$$x_i(k+1) = x_i(k) + T[q_i(k) - q_{i+1}(k)]/D_i$$

$$q_i(k) = f_1(x_{i-1})v_i(k)/(hv_i(k) + L)$$

$$v_i(k+1) = v_{max} \text{ if } x_i(k) < y_m(i) \qquad (4.30)$$

where the gains 5.3, .33 are obtained by substituting the assumed values of jerk and acceleration limits into (4.24). The flow, $q_i(k)$ is determined as in Model 1 except we use (4.2) to compute the allowable flow into link i rather than $f_2(x_i)$ as defined in Model 1. The delay may be computed as in Model 1 or by

$$\tau_i = D_i / v_i \quad . \tag{4.31}$$

The above appears to be the more accurate as shown by the simulation results of the next section.

4.4 Computational Results

To evaluate the effectiveness of the above proposed models, each network element was simulated using discrete vehicles operating under the vehicle-follower control law designed by Pue [7]. On each link the vehicle density is computed as a function of time by dividing the number of vehicles on the link at a given time by the link length. The other major quantity of interest, link delay, is measured by computing link travel time for each vehicle as it leaves the link, giving a piecewise constant function of time. The resulting values are compared to those predicted by the models.

4.4.1 Merge Simulation

The first geometry considered for verification of the traffic models is a merge junction with upstream and downstream links as shown in Fig. 4.9.

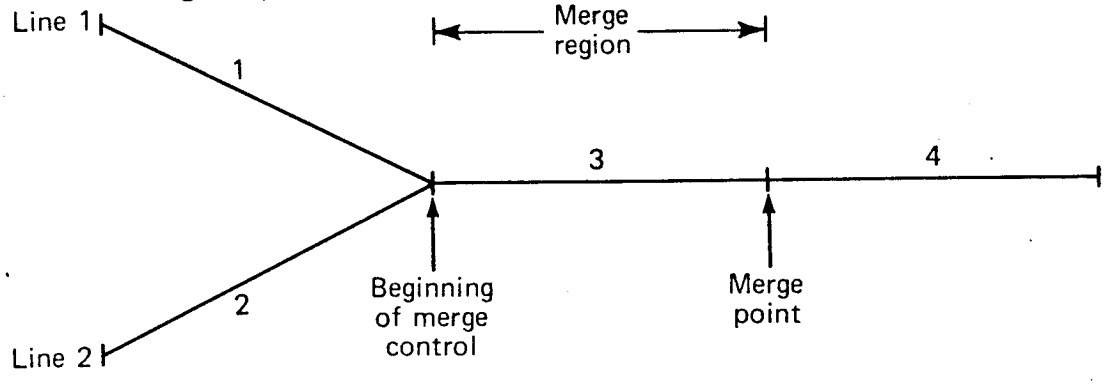

Fig. 4.9 Merge simulation geometry.

Individual vehicles enter the simulation from the left. When a vehicle enters the merge region it is assigned to follow the last vehicle to have entered the merge region on either line 1 or line 2 (Brown, [27]). The links are labeled 1 and 2 for the upstream links, 3 for the two merge links combined into a single link, and 4 is the downstream link.

The vehicle input distribution is assumed a shifted exponential in headway and is given by

$$F(h) = \begin{cases} 0 & 0 \leq h < h_m \\ F_o + (1-F_o)\,[1-\exp(-h-h_m)/h^*)], & h_m \leq h < \infty \end{cases} \qquad (4.32)$$

where h_m is the minimum allowable headway and F_o is the percentage of vehicles entering at minimum headway. The mean headway is given by

$$\bar{h} = h_m + h^* (1-F_o) \quad .$$

For a given simulation run, h is specified according to a mean traffic density, $h_m/\bar{h}$, and F_o which regulates the degree of bunchiness in the flow. That is, a large F_o will generate long strings of vehicles at h_m with large gaps in-between while h^* is adjusted in order to set $\bar{h}$. In all simulation runs, h_m = 3.0 sec.

To compare the proposed models against the simulation, the density and delay on each link of the actual system (i.e., discrete vehicle simulation) was measured. The model equations were inte-

grated in parallel, with inputs being the actual vehicle densities on sections 1, 2 and 4. The density and delay on section 3 was computed via the model and plotted against the density and delay of the actual system.

Models 1 and 2 were generated by a process of simulating a series of models based on physical considerations, discovering various flaws through examination of the results, and subsequently improving the models.

The equations for the selected models are now given followed by the simulation results.

Model 1:

$$\dot{x} = [q_3(x_1,x_3) + q_3(x_2,x_3) - q_4(x_3,x_4)]/D_3 \tag{4.33}$$

with q_3, q_4 as defined previously

Model 2:

$$\dot{v}_3 = g\left(-hv_3 + \frac{1}{x_3} - L\right) + \frac{v_4-v_3}{D_3}\left(-v_3 + \frac{1}{x_3}(5.3 - h + .33v_3)\right) \tag{4.34a}$$

$$\dot{x}_3 = [q_3 - q_4]/D_3 \tag{4.34b}$$

$$q_3 = (f_1(x_1) + f_1(x_2))v_3/(hv_3 + L) \tag{4.34c}$$

$$q_4 = f_1(x_3)v_4/(hv_4 + L) \tag{4.34d}$$

$$v_3 = v_{max} \text{ if } x_3 < y_m \tag{4.34e}$$

The equation for $v_4(t)$ is analogous to $v_3(t)$ with $v_5(t) = v_{max}$ and $x_4(t)$ = actual density as inputs. That is, after passing through the merge, it is assumed vehicles accelerate to v_{max}.

Several other variations of these models were considered in the simulation work. For example, flows could be averaged according to either an arithmetic or geometric mean of densities in adjacent links. Other models considered included summing the two input flows and then limiting by the maximum flow on a link. However, all of these variations performed poorly when compared to models 1 and 2 given above.

A variety of simulation test cases have made over a wide range of traffic densities and bunchiness of input flows for a minimum operating headway of h_m = 3 sec. An example is given for models 1 (long-dashed lines) and 2 (short dashed lines) in Fig. 4.10 where the traffic density on each input line is .5($\bar{h}$ = 6 sec) and 50 percent of the vehicles are at the minimum headway, a moderately bunchy situation. Both models compare favorably with the actual system with model 2 performing slightly better at higher densities. On the plot of delay, the curves denoted method 1 and method 2 correspond to the use of (4.7) and (4.31), respectively, in computing delay for model 2. As can be seen, method 2 is more accurate. Time delayed versions of these models show that each model is shifted to the right and peaks match more closely.

Based on these and other simulation results, model 1 is selected to be the analytic model used for design while a time delayed model 2 can be used for simulation of an actual network. The accuracy of the models deteriorate for densities above 0.1 veh/m, although

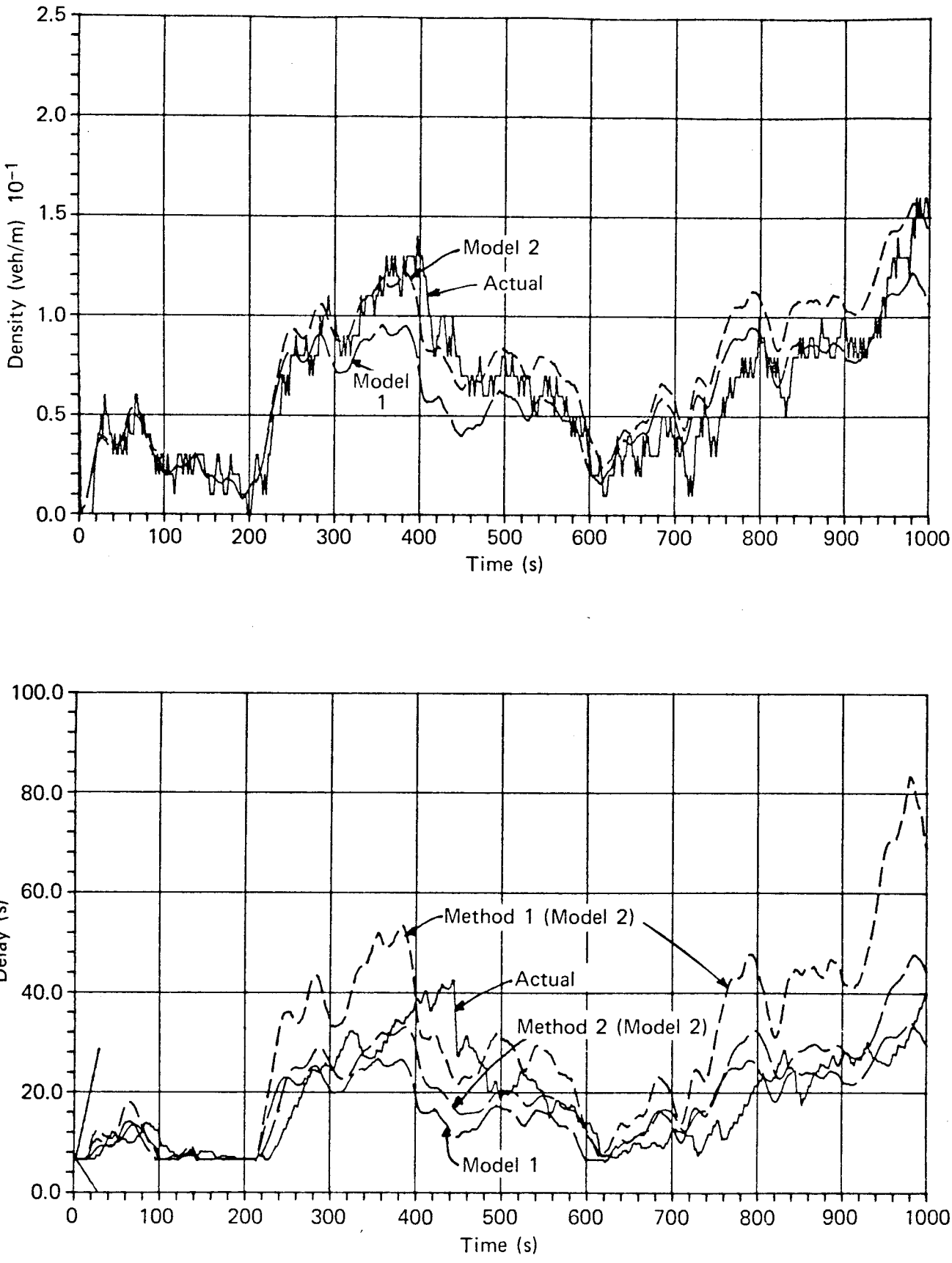

Fig. 4.10 Merge model simulation results.

this is a rather high density corresponding to a vehicle velocity of approximately 2 m/s. Modeling accuracy also degrades at very low densities as seen in the following section.

4.4.2 Diverge-Merge Simulation

To evaluate the effectiveness of models 1 and 2 when control is involved, a diverge-merge was simulated based on the representation illustrated in Fig. 4.11.

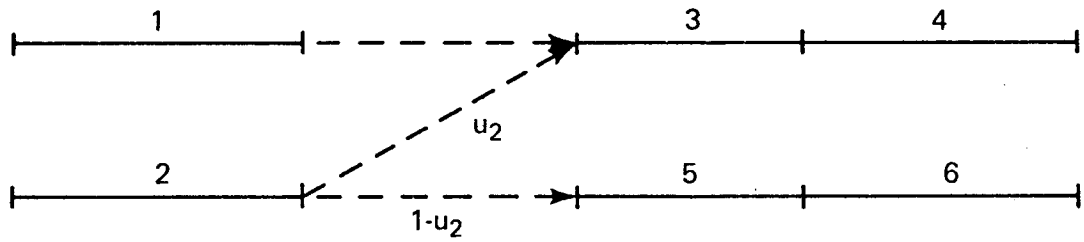

Fig. 4.11 Diverge-merge simulation model.

The model equations assuming one vehicle type (j=1) are given by

$$\dot{x}_3 = [q_3(x_1,x_3) + q_3(u_2x_2,x_3) - q_4(x_3,x_4)]/D_3$$

$$\dot{x}_5 = [q_5((1-u_2)x_2,x_5) - q_6(x_5,x_6)]/D_5 \quad .$$

The flows, q_i, are determined by either a static function of density (Model 1) or a function of link velocity (Model 2). As in the merge simulation, vehicles enter from the left via a truncated exponential distribution in headway and vehicles merge on link 3 according to a first-come, first-serve strategy.

Again, several test cases have been run for a variety of traffic densities and bunchy flows. The merge link 3 density and delay showed very similar results to the pure merge. An example of link 5 density is given in Fig. 4.12. Here, input traffic densities are .5 on link 1 and .9 on link 2, 25 percent of incoming vehicles are at the 3 sec minimum headway and the control variable u_2 is equal to .5. The resulting density on link 5 is very low with only 1 or 2 vehicles on the link at any given time, thus producing the spikes in density and delay in Fig. 4.12. Note that both models cannot respond to these single vehicle spikes and tend to give an average value for vehicle density. Vehicle delay (not shown) is at the minimum value for both the simulation and the model.

4.4.3 Station Simulation

The model described in Section 4.2.4 was simulated in parallel with a discrete vehicle simulation of the station. One result was that the model tended to smooth out the injection of vehicles from the station onto the main guideway. Because bunchiness of vehicle flows could be important with respect to congestion, several modifications to the model were considered in order to represent this effect.

The major modification was a redefinition of $f_1(x_e)$ to include hysteresis for $x_e < x^e_{max}$. That is

$$f_1(x_e) = \begin{cases} x_e/x^e_{max} & \text{for } u_j = 0 \\ I/(D^e_i\, x^e_{max}) & \text{for } u_j = 1 \end{cases}$$

$j=1, \ldots, J$ and I is an appropriate integer corresponding to

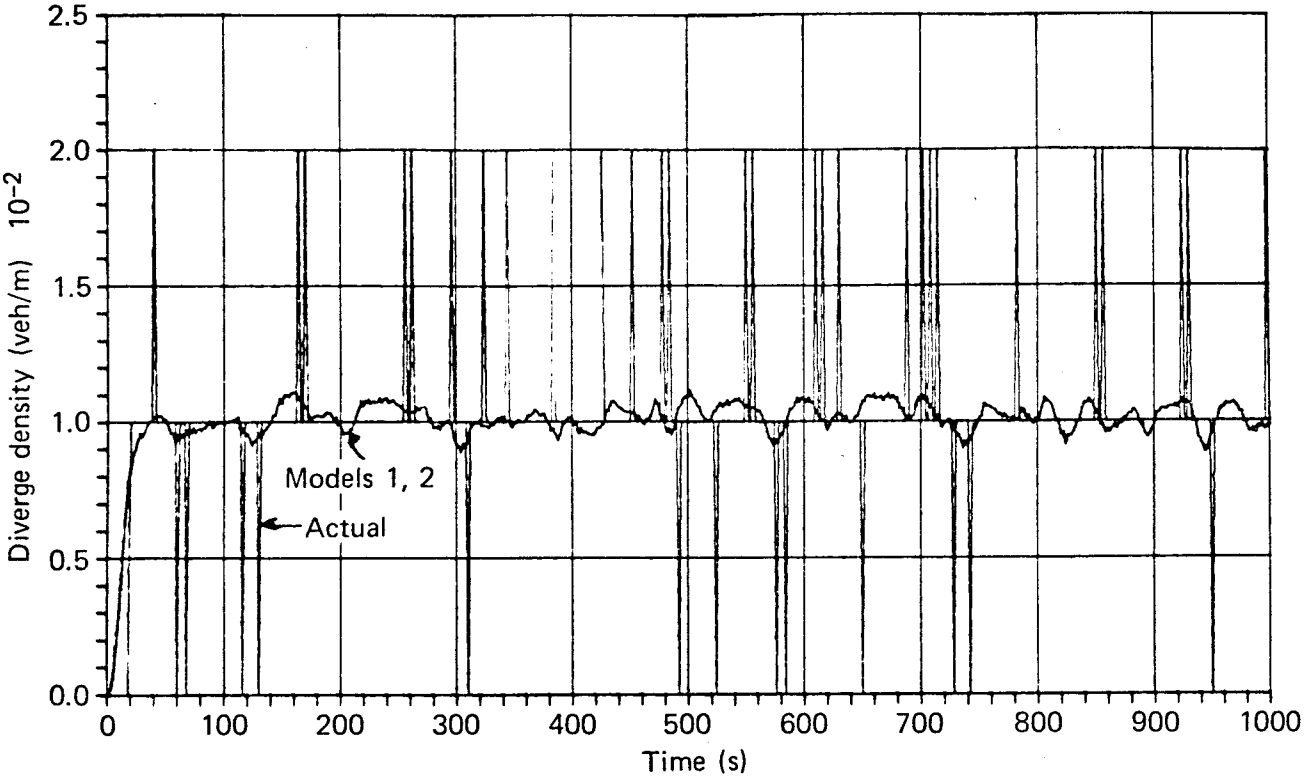

Fig. 4.12 Diverge-merge simulation results - link 5.

quantization of $x_e D_i^e$ to discrete vehicles. This model considerably complicates an analytic model but could be easily incorporated into a simulation model.

A second refinement is used to account for the fact that only one vehicle can pass through the station entry point at a time. Thus, if the arrival queue is backed up into the mainline, vehicles cannot pass the station. This phenomenon would also occur on diverge links, that is, if the flow is blocked to one outgoing link, the flow is also cut-off to the other outgoing link. As a result, the flow, q_i^d, out of a diverge link, i, could be represented by

$$q_i^d = f_2(x_{i+1}) f_2(x_{i+2}) / (q_{max}(i+1) q_{max}(i+2))^{\frac{1}{2}} \quad . \qquad (4.35)$$

Again, this modification could be useful for a simulation model but the increase in complexity for the analytic model is not justified by the increase in performance.

A variety of simulation test cases have been made to investigate the effectiveness of the suggested models. One example showing arrival lane density and delay is given in Fig. 4.13. Here, the input traffic density is .90 and 50 percent of the vehicles are at minimum headway. The percent of upstream vehicles diverted into the station is 20 and the station dock dwell time is 15 sec. The lead in the arrival lane delay is due to the fact that the model computes the delay to a vehicle entering the arrival lane while the simulation of discrete vehicles computes the delay to vehicles leaving the arrival lane. As in the merge model, the station model tends to be less representative at densities above 0.1 veh/m. A

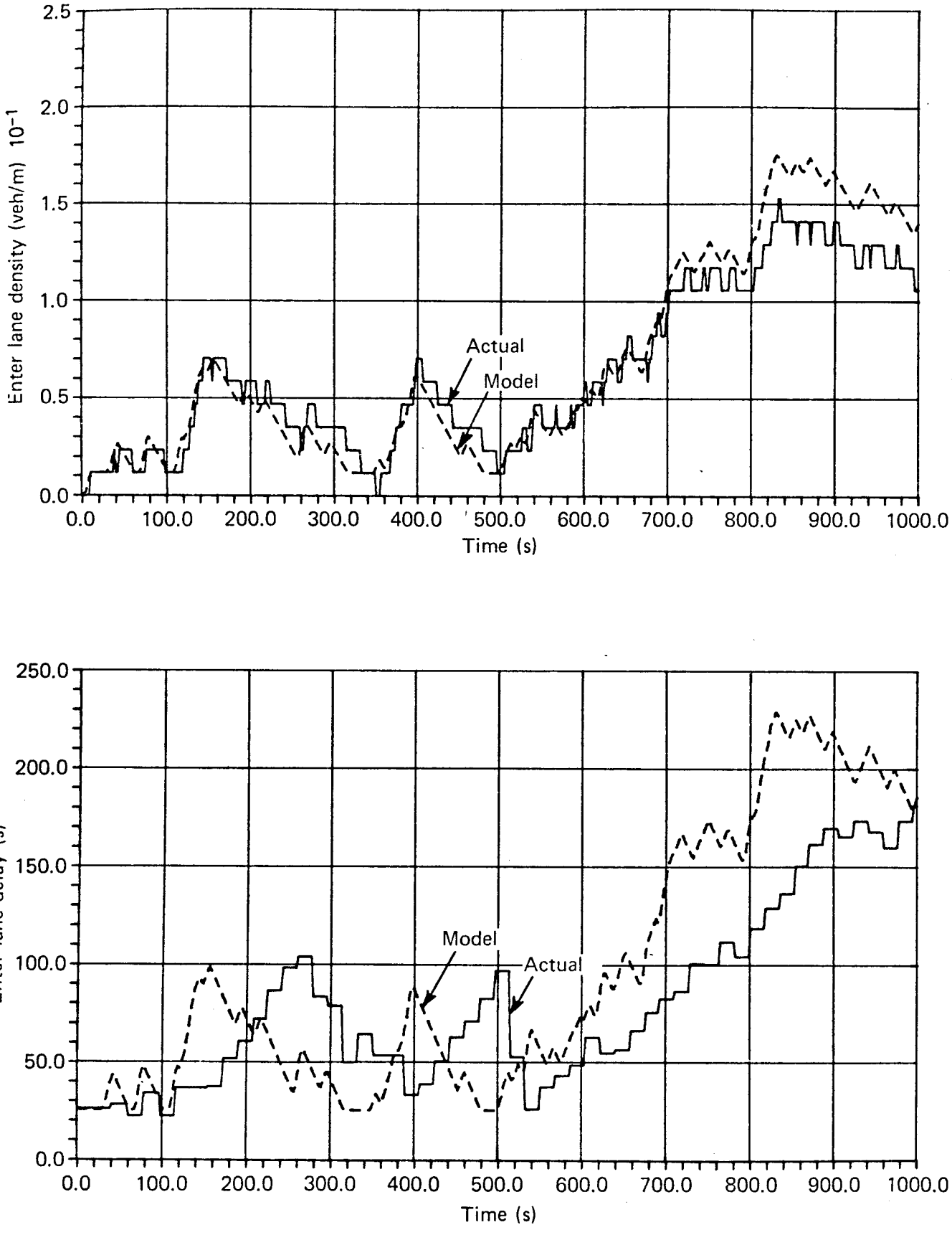

Fig. 4.13 Station simulation - arrival lane.

plot of vehicle density and delay on the egress lane is given in Fig. 4.14. When delay exceeds the minimum delay, the model over predicts the actual delay by approximately 20 percent. This is because the continuum model assumes an interference between mainline and egressing vehicles whenever there are vehicles present on these links and they exceed the threshold densities computed in 4.2.3. On the other hand, when individual vehicles are modeled the precise relative positioning of vehicles on the mainline and egress links determines the incurred delays as the vehicles maneuver. The discrepancy between the model and actual system due to this effect becomes significant at low densities as revealed in Fig. 4.14.

4.5 Conclusions

Analytic models for the basic elements of an AGT network have been developed and compared to discrete vehicle simulations of a diverge, merge and station. In particular, a nonlinear function of vehicle densities in adjacent links is used to compute vehicle flow across link boundaries. This function was introduced to help preserve the representation of bunchy input flows at merge junctions, otherwise flows tended to smooth out to average values.

Simulations of the various network elements demonstrate that the models well represent discrete vehicle behavior for moderate densities. In a 3 second headway system this corresponds to a density range of 0.01 to 0.1 veh/m. Outside this range, individual vehicle positioning and dynamics become significant and the continuum model is degraded in representing the true system. Moreover, if many links are connected in a long string, it is expected that smoothing of flow would occur in the model. Thus, final verification

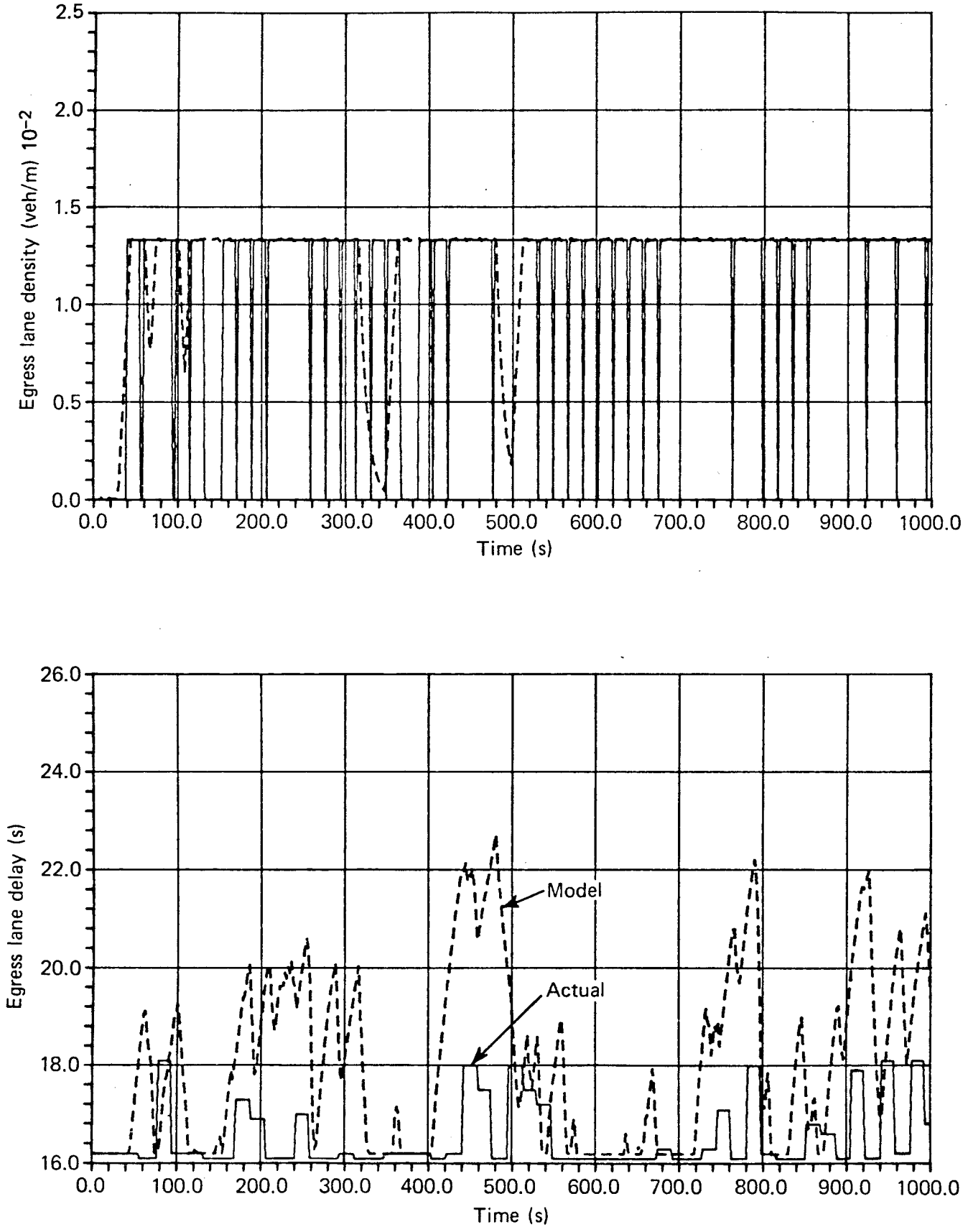

Fig. 4.14 Station simulation - egress lane.

of the proposed models would require a network simulation to investigate the importance of this effect upon the study of network operation.

5. OPTIMAL CONTROL FORMULATION

We now combine the model of traffic flow constraints and station constraints given in Section 4 with a performance index to formulate the routing problem into an optimal control problem. We then demonstrate that duality may be applied to decouple the overall network flow constraints into simpler subnetwork constraints that allow a distributed control computation.

As in multicommodity flow problems a vehicle type will be associated with a particular origin-destination pair. For each O-D pair we apriori select a set of links which is "appropriate" for that pair, that is, exclusion of loops and any extraordinarily long paths that would be unacceptable to a customer. For practical network configurations that have been proposed [38] this does not pose any serious problem. Thus, associated with each O-D pair is a subnetwork which we index by j. Letting J be the total number of subnetworks or O-D pairs, we define for j=1, ..., J;

$\mathcal{L}_j$ = set of links in subnetwork j

C_j = set of diverge links in subnetwork j,

To define a performance index we use (4.7), the delay (travel time) per vehicle on each link i. The instantaneous cost to vehicle type j on link i is $\tau_i(y_i)x_{i,j}D_i$ and the total cost for all vehicles on link i is $\tau_i(y_i)y_iD_i$. Summing over all links in the network and summing over K time intervals the total time average delay is

$$\sum_{k=1}^{K} \sum_{i \in \mathcal{L}} TD_i \tau_i(y_i(k))y_i(k)/t_f \qquad (5.1)$$

where $\mathcal{L}$ is the set of network links and t_f is the final time. If we define $d_i(y_i(k)) = TD_i\tau_i(y_i(k))y_i(k)/t_f$ the performance measure is written as

$$\sum_{k=1}^{K} \sum_{i\in\mathcal{L}} d_i(y_i(k)) \quad . \tag{5.2}$$

We include a cost on passengers waiting at stations by appending the number of waiting passengers of each vehicle type to the cost function. This cost is designed to force the dispatch of empties from stations with excess vehicles. Thus, the performance index is

$$I(x,y) = \sum_{k=1}^{K} \left[\sum_{i\in\mathcal{L}} d_i(y_i(k)) + T \sum_{j=1}^{J} x_{1,j}(k) \right] \tag{5.3}$$

where the index 1 in $x_{1,j}(k)$ refers to the trip request queue for vehicle type j.

The above cost is a system performance measure because it sums the total trip time to all vehicles as they pass through the system. Hence, it is possible some vehicle may incur a large delay while others have a very short delay when the sum is a minimum. Note, however, during uncongested conditions where $\tau_i(y_i(k))$ is a constant travel time, (5.3) will favor shortest path routes thus reflecting some degree of preference to the user. Moreover, the apriori selection of possible subnetwork paths reflects user costs.

We now define the vectors

u = routing and dispatch control variables

u_j = routing and dispatch control variables for vehicle type j

x_j = vehicle type j link densities

y = link densities

v = total routing control variables

v' = alternate total routing control variables

z = $[y,v,v']$ = vector of interconnection variables

z_j = $[y_j,v_j,v_j']$ = vector of interconnection variables for links in subnetwork, j.

Using the above notation the problem statement is

$$\min_u I(x,y) \text{ such that } 0 \leq u \leq 1 \tag{5.4}$$

subject to the dynamic constraints

$$x_j(k+1) = F(x_j(k), u_j(k), z_j(k)) \quad j=1, \ldots, J \quad . \tag{5.5}$$

and the interconnection constraints

$$y_i(k) = \sum_j x_{i,j}(k) \tag{5.6}$$

$$v_i(k) = \sum_i u_{i,j}(k) x_{ij}(k) \tag{5.7}$$

$$v_i'(k) = \sum_j (1-u_{i,j}(k)) x_{ij}(k) \quad . \tag{5.8}$$

As noted in Section 4, the dynamic flow constraints have a relatively simple form if the interconnection variables are considered as independent variables rather than a function of state. This suggests a problem manipulation where the subproblems consist of

individual vehicle type and the total density variables are treated as additional control variables.

Such an approach corresponds to the goal coordination method of optimal control where we dualize with respect to the interconnection constraints. As a result the problem becomes

$$\underset{\lambda,\mu,\mu'}{\text{Max}} \quad \Phi(\lambda,\mu,\mu')$$

where

$$\Phi(\lambda,\mu,\mu') = \min \sum_{k=1}^{K} \Big[\sum_{i\in\mathcal{L}} d_i(y_i(k)) + T \sum_{j=1}^{J} x_{l,j}(k)$$

$$+ \sum_{i\in\mathcal{L}} \lambda_i(k)(y_i(k) - \sum_j x_{i,j}(k))$$

$$+ \sum_{i\in C} \{\mu_i(k)(v_i(k) - \sum_j u_{i,j}(k)x_{i,j}(k))$$

$$+ \mu_i'(k)(v_i'(k) - \sum_j (1-u_{i,j}(k))x_{i,j}(k))\}] \qquad (5.9)$$

$$\text{with } x\in X(u,z),\ u\in U,\ z\in Z \quad .$$

The constraint sets are defined as

$$X(u,z) = \{x:\ x_j(k+1) = F(u_j(k), x_j(k), z_j(k)),\ j=1, \ldots, J\}$$

$$U = \{u:\ 0 \le u \le 1;\ \text{first-come, first-serve priority for dispatch controls}\}$$

$$Z = \{z:\ y_m(i) \le z_i \le 1/L,\ i\in\mathcal{L}\} \quad .$$

The algorithm used to solve this problem is given in the following section. However, we will first demonstrate how the dual function (5.9) can be computed in a distributed manner.

First, in view of (4.19), (4.20), and (4.23) the dynamics may be written in the form

$$x_j(k+1) = [F_j(k) + \sum_{i \in C_j} u_{i,j}(k) B_{i,j}(k)] x_j(k) + r_j(k)$$

$$= A_j(k) x_j(k) + r_j(k) \quad . \tag{5.10}$$

where $F_j(k)$, $B_{i,j}(k)$ are functions of the interconnection variables, $z_j(k)$. The dual function objective may also be written as

$$\phi(u,z) = \sum_{k=1}^{K} \Big[\sum_{j=1}^{J} \alpha_j^T(k) x_j(k) + \sum_{i \in \mathcal{L}} f_i(y_i(k)) + \mu^T(k) v(k)$$

$$+ \mu'^T(k) v'(k) \Big] \tag{5.11}$$

where

$$\alpha_{i,j}(k) = \begin{cases} -\lambda_i(k) & i \in \mathcal{L}_j,\ i \in C_j \\ -\lambda_i(k) - \mu_i(k) u_{i,j}(k) - \mu_i'(k)(1-u_{ij}(k)) & i \in C_j \\ T & i = 1 \end{cases}$$

$$f_i(y_i(k)) = d_i(y_i(k)) + \lambda_i(k) y_i(k)$$

As explained in Section 7, (5.11) is minimized by application of the Principle of Optimality (or Dynamic Programming). That is, at each time step k, the cost-to-go is minimized with respect to the control variables u(k), z(k). Satisfaction of the Dynamic Programming recursion guarantees an optimal solution, namely, if

$$V_k = \min_{u(k),z(k)} [\psi_k + V_{k+1}] \text{ for } k=1, \cdots, K$$

where

$$\psi_k = \sum_{j=1}^{J} \alpha_j^T(k)x_j(k) + \sum_{i\varepsilon\mathcal{L}} f_i(y_i(k)) + \mu^T(k)v(k)$$

$$+ \mu'^T(k)v'(k).$$

then u(k), z(k) for k=1, $\cdots$, K, are optimal.

As a result, an essential feature of the optimization algorithm is to minimize ϕ at the k^{th} stage for k=1, $\cdots$, K where the k^{th} stage optimization problem is

$$\min_{u(k),\, z(k)} \quad \phi_k \tag{5.12}$$

such that $x\varepsilon X(u,z)$, $u\varepsilon U$, $z\varepsilon Z$ and

$$\phi_k = \sum_{\ell=k}^{K} (\Sigma\alpha_j{}^T(\ell)x_j(\ell)) + \sum_{i\varepsilon\mathcal{L}} f_i(y_i(k))$$

$$+ \mu^T(k)v(k) + \mu'^T(k)v'(k). \tag{5.13}$$

The only difficulty in solving (5.12) is to determine the dependence of the first term in (5.13) upon u(k), z(k).

Because of the triangular structure of $A_j(k)$ (5.12) can be solved in a distributed manner by defining the adjoint variables

$$p_j(k) = A_j^T(k)p_j(k+1) - \alpha_j(k) \tag{5.14}$$

$$p_j(K+1) = 0 \quad .$$

A listing of the adjoint equations for each link type is given in Appendix B. Use of (5.14) is explained below.

Expanding the first term of (5.13) in time we have

$$\sum_j \sum_{\ell=k}^{K} \alpha_j^T(\ell)x_j(\ell) = \sum_j [\alpha_j^T(k)x_j(k) + \alpha_j^T(k+1)x_j(k+1) + \cdots + \alpha_j^T(K)x_j(K)]. \tag{5.15}$$

Recursively using (5.10), the summation (5.15) may be expressed in terms of $x_j(k)$ or

$$\begin{aligned}\sum_j \sum_{\ell=k}^{K} \alpha_j^T(\ell)x_j(\ell) = \sum_j \{&\alpha_j^T(k)x_j(k) + \alpha_j^T(k+1)[A_j(k)x_j(k)+r_j(k)] \\ &+ \alpha_j^T(k+2)[A_j(k+1)[A_j(k)x_j(k)+r_j(k)] + r_j(k+1)] \\ &\vdots \\ &+ \alpha_j^T(K)[A_j(K-1) \cdots A_j(k)x_j(k) \\ &\quad + A_j(K-1) \cdots A_j(k+1)r_j(k) \\ &\quad + A_j(K-1) \cdots A_j(k+2)r_j(k+1) \\ &\quad \vdots \\ &\quad + r_j(K-1)]\}.\end{aligned} \tag{5.16}$$

All terms involving $r_j(\ell)$ for $\ell \geq k+1$ are not involved in the minimization and can therefore be neglected. If we separate the terms associated with $x_j(k)$ and $r_j(k)$ we then have

$$\sum_j \sum_{\ell=k}^{K} \alpha_j^T(\ell)x_j(\ell) = \sum_j \{\alpha_j^T(k)x_j(k)+[\alpha_j^T(k+1)+\alpha_j^T(k+2)A_j(k+1) + \cdots$$

$$+ \alpha_j^T(K)A_j(K-1) \cdots A_j(k+1)][A_j(k)x_j(k)+r_j(k)]\}$$

$$+ C \qquad (5.17)$$

where C is not a function of u(k), z(k). Using (5.14) the summation (5.17) is given by

$$\sum_j \sum_{\ell=k}^{K} \alpha_j^T(\ell)x_j(\ell) = \sum_j \{\alpha_j^T(k)x_j(k)-p_j^T(k+1)[A_j(k)x_j(k)+r_j(k)]\}$$

$$+ C. \qquad (5.18)$$

The summation in (5.13) can now be expressed in terms of the adjoint variables so that (5.13) becomes

$$\phi_k = \sum_j([\alpha_j{}^T(k) - p_j{}^T(k+1)A_j(k)]x_j(k) - p_j{}^T(k+1)r_j(k))$$

$$+ \sum_{i\epsilon\mathcal{L}} f_i(y_i(k)) + \mu^T(k)v(k) + \mu'^T(k)v'(k) + C \quad . \qquad (5.19)$$

As a result, for link i the dependence of ϕ_k on $y_i(k)$ can be written as

$$\phi(y_i(k)) = f_i(y_i(k))$$

$$+ T \sum_{j=1}^{J} \{a_{i-1,i}(k)x_{i-1,j}(k)[p_{i-1,j}(k+1)/D_{i-1}-p_{i,j}(k+1)/D_i]$$

$$+ a_{i,i+1}(k)x_{i,j}(k)[p_{i,j}(k+1)/D_i-p_{i+1,j}(k+1)/D_{i+1}]\} \tag{5.20}$$

The minimization of (5.20) with respect to $y_i(k)$ is considerably simplified if the change in value of the objective function along a coordinate direction, y_i, only depends on y_i. This may be accomplished by a simple modification of the flow model.

For adjacent links i, i+1 when $y_i > y_m(i)$ and $y_{i+1} > y_m(i+1)$, that is the densities are above the threshold values for vehicle following the velocity function is given by

$$a_{i,i+1}(k) = (1-y_{i+1}L)/(hy_i) \quad . \tag{5.21}$$

The simplification is achieved if $a_{i,i+1}(k)$ is only a function of either y_i, or y_{i+1}, or if it is an additive function of y_i and y_{i+1}. We take the latter approach by determining a function that approximates (5.21) and has the form

$$a_{i,i+1}(k) = g_1(y_i) + g_2(y_{i+1}) \quad .$$

If we expand (5.21) into a Taylor series about $y_m(i)$ and $y_m(i+1)$ and drop cross derivative terms the result is

$$a_{i,i+1}(k) = q_m(i+1)/y_i + L(y_m(i+1)-y_{i+1})/(hy_m(i)) \qquad (5.22)$$

giving the desired form. A similar approximation is used for the station arrival velocity (Appendix A).

Simulations of the modified model (5.22) versus the original model (5.21) have shown essentially identical performance over the density range for which the original model is valid. An example showing a comparison with (5.22) and Model 2 is given in Fig. 5.1. A plot of the original Model 1 would precisely overlay the revised Model 1 plot in Fig. 5.1.

Because of the form of the velocity function (5.22) we can independently minimize ϕ_k for each link density. That is, as a function of $y_i(k)$, ϕ_k has the form

$$\phi(y_i(k)) = f_i(y_i(k)) + c_1 y_i(k) + c_2/y_i(k) \qquad (5.23)$$

where c_1, c_2 are not functions of u(k), z(k). Note that c_1 and c_2 are computed using information local to link i.

For a diverge link the optimization problem becomes more complex than a link because the subnetwork control variables, $u_{i,j}(k)$, cannot be decoupled from $z_i(k) = [v_i(k), v_i'(k), y_{i+1}(k), y_{i+2}(k)]$. The value of ϕ_k due to these terms has the form

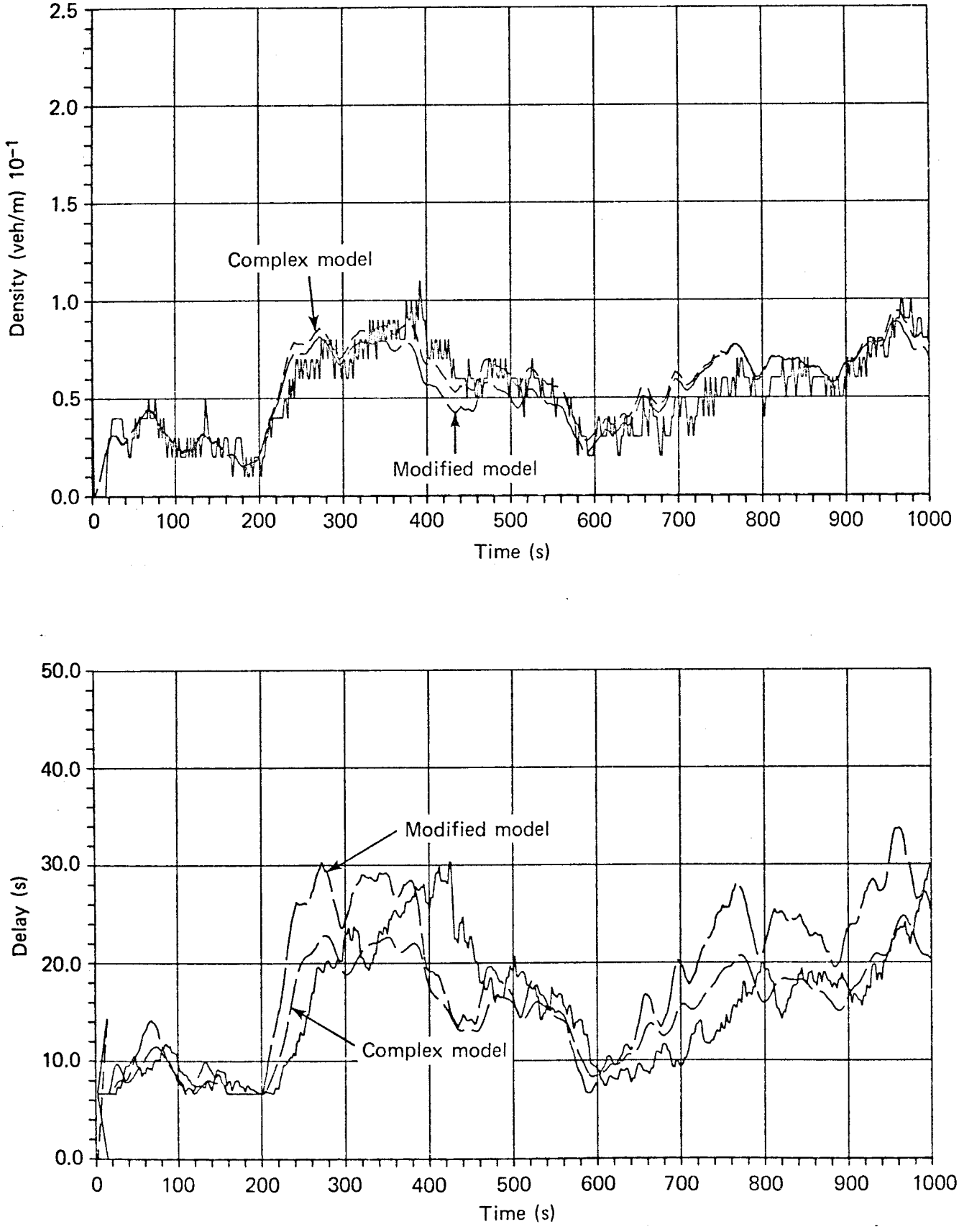

Fig. 5.1 Modified model, complex model comparison.

$$\phi_k(z_i(k)) = f_{i+1}(y_{i+1}(k)) + f_{i+2}(y_{i+2}(k)) + d_i(v_i(k) + v_i'(k))/2$$

$$+ a_{i,i+1}(k)\sum_j u_{i,j}(k)[-p_{i+1,j}(k+1)/D_{i+2} + p_{i,j}(k+1)/D_i]x_{i,j}(k)$$

$$+ a_{i,i+2}(k)\sum_j (1-u_{i,j}(k))[-p_{i+2,j}(k+1)/D_{i+2} + p_{i,j}(k+1)/D_i]x_{i,j}(k)$$

$$+ a_{i+1,i+3}(k)\sum_j [p_{i+1,j}(k+1)/D_{i+1} - p_{i+3,j}(k+1)/D_{i+3}]x_{i+1,j}(k)$$

$$+ a_{i+2,i+4}(k)\sum_j [p_{i+2,j}(k+1)/D_{i+2} - p_{i+4,j}(k+1)/D_{i+4}]x_{i+2,j}(k)$$

$$+ (\mu_i'(k) - \mu_i(k))\sum_j u_{i,j}(k)x_{i,j}(k) + \mu_i(k)v_i(k) + \mu_i'(k)v_i'(k) \quad . \tag{5.24}$$

where link i+3 is downstream from link i+1 and link i+4 is downstream from link i+2. Again, because of the additive form of the velocity function, (5.24) can be written as

$$\phi_k(z_i(k)) = f_{i+1}(y_{i+1}(k)) + f_{i+2}(y_{i+2}(k)) + d_i(v_i(k) + v_i'(k))/2$$

$$+ \sum_j u_{i,j}(k)[c_1 + c_2 y_{i+1}(k) + c_3 y_{i+2}(k) + c_4/v_i(k) + c_5/v_i'(k)]$$

$$+ c_6 v_i'(k) + c_7 y_{i+2}(k) + c_8/y_{i+1}(k) + c_9/y_{i+2}(k)$$

$$+ \mu_i(k)v_i(k) + \mu_i'(k)v_i'(k) \tag{5.25}$$

where c_ℓ, $\ell=1, \ldots, 9$ are not functions of $u(k)$, $z(k)$. The algorithm

for minimizing (5.25) with respect to $z_i(k)$ is given in Appendix C. Again, only information local to link i is required. Note the presence of the term, $d_i(v_i(k) + v_{\hat{i}}'(k))$. This is included to prevent occurrence of a duality gap, as discussed in Section 7.

6. ALGORITHM DESCRIPTION

The basic algorithm consists of a subnetwork optimization and an upper level coordinating control [23]. As illustrated in Fig. 6.1, the upper level seeks to maximize the dual function by finding Lagrange Multipliers (λ,μ) via subgradient optimization ([39] [40]). The lower level minimization problem requires computation of routing and dispatch control variables u, total control variables v, and total density variables, y. As shown in Section 5 this can be accomplished in a distributed manner using information local to each link. Each subnetwork updates state and adjoint variables by integrating the state equation from origin to destination and the adjoint equation from destination to origin.

The specific steps of the algorithm are:

1. Initialize state variables, Lagrange multipliers and adjoint variables
2. Compute $\Phi(\lambda_o,\mu_o)$ by minimizing the cost-to-go at each time step ℓ as follows:

2a. $$\min_{\substack{x(\ell+1)\in X \\ u(\ell)\in U \\ z(\ell)\in Z}} \sum_{k=\ell}^{K} \phi_k \quad \text{for } \ell=1, \ldots, K$$

Each link may be independently optimized except for diverge links which are optimized by the algorithm given in Appendix C.

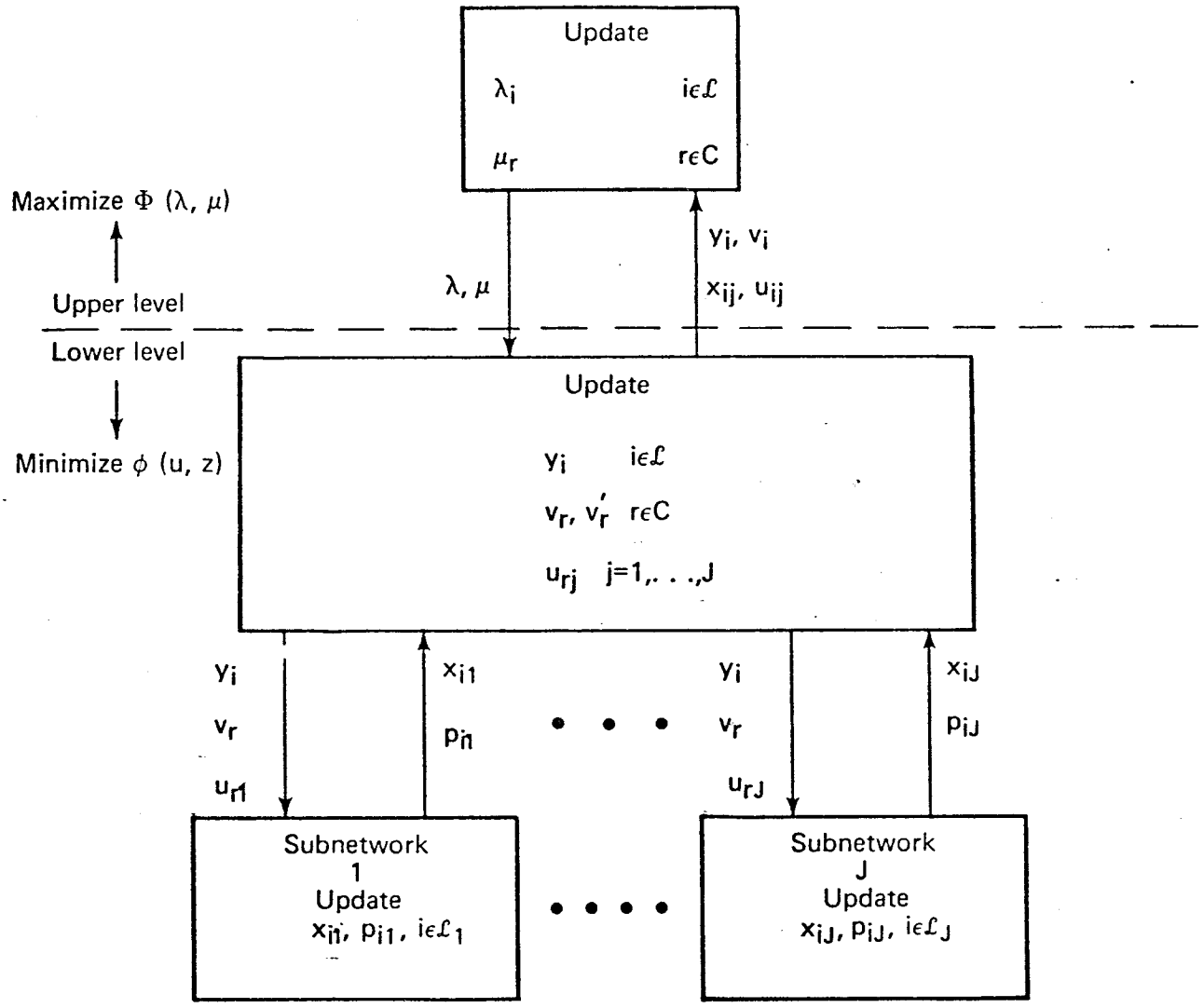

Fig. 6.1 Algorithm structure.

2b. If a stationary point is found (i.e., no change in variables) stop. At the n^{th} iteration the values provided to the upper level are v^n, v'^n, y^n, u^n.

2c. Otherwise, compute new adjoint and state variables and return to 2a.

3. Compute subgradients of Φ and update multipliers

$$\lambda_i^{n+1}(k) = \lambda_i^n(k) + \alpha_n(y_i^n(k) - \sum_j x_{i,j}(k))$$

$$v_i^{n+1}(k) = v_i^{n+1}(k) + \alpha_n(v_i^n(k) - \sum_j u_{i,j}^n(k)x_{i,j}^n(k))$$

$$v_i'^{n+1}(k) = v_i^{n+1}(k) + \alpha_n(v_i'^n(k) - \sum_j (1-u_{i,j}^n(k))x_{i,j}^n(k))$$

4. If interconnection constraints are satisfied, stop, otherwise, return to step 2.

The step-size rule that determines α_n is given in Section 7.

7. CONVERGENCE ANALYSIS

The analysis of algorithm convergence is divided into three sections. First, we prove overall algorithm convergence, that is, the dual function is maximized via subgradient optimization. To compute the subgradient for upper level iteration, we next show the lower level algorithm converges to a minimum of the dual function objective. Finally, it is shown that no duality gap exists for the problem and the subgradient magnitude does not necessarily go to zero.

7.1 Upper Level Convergence

To maximize the dual function, subgradient optimization is applied ([39], [40], [41]). The proof given below is a modification of one given by Poljak [40] where the subgradient algorithm has access to all information. To implement a distributed control, we apply the step-size rule first given by Ermol'ev [39].

7.1.1 Theorem 1

The algorithm consists of constructing the sequence

$$\lambda^{n+1} = \lambda^n + \alpha_n \gamma_n \bar{\lambda}^n \tag{7.1}$$

where $\bar{\lambda}^n$ is a subgradient to the function, $\phi(\lambda)$, at the point, λ^n. The step length parameters, α_n, γ_n are subject to the conditions

$$\lim_{n\to\infty} \alpha_n = 0 \tag{7.2}$$

$$\sum_{n=0}^{\infty} \alpha_n = \infty \tag{7.3}$$

$$0 < \varepsilon_1 \leq \gamma_n \leq \varepsilon_2 / ||\bar{\lambda}^n|| \ . \tag{7.4}$$

Then, for arbitrary λ^o, the sequence, $\{\lambda^n\}$, contains a subsequence $\{\lambda^{n_k}\}$, $k \varepsilon K$ for which

$$\lim_{k \to \infty} \phi(\lambda^{n_k}) = \phi^* = \max_{\lambda} \phi(\lambda) \ .$$

Proof:

Select an arbitrary $\beta < \phi^*$ and define $S_\beta = \{\lambda : \phi(\lambda) \geq \beta\}$. Suppose that $\lambda^n \notin S_\beta$ for all n. It follows that $S_\beta \subset \{\lambda : \phi(\lambda) \geq \phi(\lambda^n)\}$ and $\phi(\lambda_o) > \phi(\lambda^n)$ for all $\lambda_o \varepsilon S_\beta$. Because $\phi(\lambda)$ is concave we also have

$$\phi(\lambda_o) - \phi(\lambda^n) \leq (\bar{\lambda}^n, \lambda_o) - (\bar{\lambda}^n, \lambda^n) \ . \tag{7.5}$$

where $(\cdot,\cdot)$ denotes inner product. As a result,

$$\begin{aligned} ||\ \lambda^{n+1} - \lambda_o||^2 &= ||\ \lambda^n + \alpha_n \gamma_n \bar{\lambda}^n - \lambda_o||^2 \\ &= ||\ \lambda^n - \lambda_o||^2 + \alpha_n^2 \gamma_n^2 ||\bar{\lambda}^n||^2 \\ &\quad + 2\alpha_n \gamma_n (\bar{\lambda}^n, \bar{\lambda}^n - \lambda_o) \\ &= ||\ \lambda^n - \lambda_o\ ||^2 + \alpha_n^2 \gamma_n^2 ||\ \bar{\lambda}^n||^2 \\ &\quad - 2\alpha_n \gamma_n [(\ \bar{\lambda}^n, \lambda_o) - (\bar{\lambda}^n, \lambda^n)] \end{aligned}$$

$$\leq || \lambda_n - \lambda_o ||^2 + \alpha_n^2 \gamma_n^2 || \bar{\lambda}^n ||^2$$

$$- 2\alpha_n \gamma_n [\phi(\lambda_o) - \phi(\lambda^n)]$$

$$\leq || \lambda^n - \lambda_o ||^2 + \alpha_n^2 \varepsilon_2^2 - 2\alpha_n \varepsilon_1 \varepsilon_3 \qquad (7.6)$$

where the last two steps follow from (7.5) and (7.4), and the fact

$$\phi(\lambda_o) - \phi(\lambda^n) \geq \varepsilon_3 > 0,$$

Now, sum (7.6) from n=N to n=N + m to obtain

$$||\lambda^{N+m+1} - \lambda_o||^2 \leq ||\lambda^N - \lambda_o||^2 + \sum_{n=N}^{N+m} \alpha_n (\alpha_n \varepsilon_2^2 - 2\varepsilon_1 \varepsilon_3) .$$

Finally, choose N such that $\alpha_n \leq \varepsilon_1 \varepsilon_3 / \varepsilon_2^2 = \varepsilon / \varepsilon_2^2$ to obtain

$$0 \leq ||\lambda^N - \lambda_o||^2 - \varepsilon \sum_{n=N}^{N+m} \alpha_n . \qquad (7.7)$$

Because the summation, $\Sigma \alpha_n$, diverges (7.7) is impossible and thus, there exists $\lambda^{n_k} \in S_{\beta_k}$. If we let $\beta_k \to \phi^*$ we obtain the desired subsequence.

Note that the above step-size selection rule requires the subgradient magnitude be upper bounded, otherwise $\gamma_n = 0$. This property is satisfied in our application since x,y,v are bounded between 0 and 1/L, and therefore, the interconnection errors used for subgradient computation are bounded in magnitude.

7.2 Lower Level Convergence

The minimization problem required to compute the dual function can be written as

$$\min_{z \in Z} F(z) \tag{7.8}$$

where the vector z consists of

$$x_{i,j}(k) \quad ; \quad i \in \mathcal{L}_j$$
$$u_{i,j}(k) \quad ; \quad i \in C_j$$
$$y_i(k) \quad ; \quad i \in \mathcal{L}$$
$$v_i(k) \quad ; \quad i \in C$$
$$v_i'(k) \quad ; \quad i \in C$$

for $j = 1, \ldots, J$ and $k = 1, \ldots, K$.

The set $Z = X \times U \times Y \times V$ where

$$X = \{x | x = G(u,y,v),\ u \in U,\ y \in Y,\ v \in V\}$$
$$U = \{u |\ 0 \leq u_{i,j}(k) \leq 1\}$$
$$Y = \{y |\ 0 \leq y_i(k) \leq 1/L\}$$
$$V = \{v,v' |\ 0 \leq v_i(k),\ v_i'(k) \leq 1/L\}$$
$$F(z) = f(z) + \lambda^T g(z) .$$

The dispatch control variables are included in the set, U. This is done to assure the continuity of F although physically, these variables can only have a value of zero or one. However, this poses no problem since F is linear in the dispatch variables.

According to Theorem 6.3.4 of [42] if z^* is a solution to (7.8) then $g(z^*)$ qualifies as a subgradient. Thus, it is necessary to show that the algorithm gives an absolute minimum of F(z), otherwise the subgradient computation may not be valid.

To assure an absolute minimum, we apply Bellman's Principle of Optimality [43]. That is, at each time step, k, we define the recursion,

$$V_k = \min_{z_k} [F_k(z_k) + V_{k+1}] \tag{7.9}$$

where z_k is the portion of z at time step k and $F_k(z_k)$ is the cost at the k^{th} stage.

The proposed algorithm is to compute y_k, v_k, u_k at a given x_k for k=1, ..., K. Using z_k we then compute z_{k+1}. The process is continued and repeated until a stationary condition satisfying the above recurrence relation (7.9) is reached. According to the Principle of Optimality, z_k is then an optimal control. Thus, we need to prove that a stationary point is, indeed, obtain via the proposed algorithm.

A point, z, is stationary if there is no z_k such that $F(z_k)<F(z)$, $z_k \epsilon Z$, k=1, ..., K. We define the sets

$$\Omega = \{z: z \text{ is a stationary point}\}$$

$$Z_o = \{z: z\epsilon Z,\ F(z) \leq F(z_o)\}.$$

Note F(z) is continuous, Z is compact, and Z_o is compact by continuity of F(z).

7.2.1 Theorem 2

Let $A_k: Z \to 2^Z$ be a point to set map at time k. We define the algorithm

$$z_{n+1} \in A_k(z_n) \ , \quad \text{if } z_{n+1} \in \Omega, \text{ stop}$$

where $A_k(z_n)$ is the minimization at time step k defined by

$$A_k(z_o,\xi_k) = \{z: z = z_o + \sum_i \bar{\alpha}_i \xi_k^i \text{ for some } \bar{\alpha}_i \in L \text{ and}$$

$$F(z) \leq F(z_o + \sum_i \alpha_i \xi_k^i)\}$$

where $L = \{\alpha_i | z_o + \sum_i \alpha_i \xi_k^i \in Z\}$

ξ_k^i = vector with one in position corresponding i^{th} variable in z_k and zeros in the remaining positions.

Then every convergent subsequence of $\{z_n\}$ has a limit in Ω.

The proof is a straight forward modification of the proof corresponding to convergence of the coordinate search algorithm. Several proofs are described in [42]. The modification applies compactness of Z_o, the fact that A_k is closed, and continuity of F.

7.3 Duality Gap

When applying duality to nonconvex optimization problems it is generally expected that duality gaps will appear, that is, the cost function optimum value is greater than the dual function optimum value. However, for this application we can show that the duality gap is zero despite nonconvexity of the constraints.

In [44], a duality gap estimate is derived for the following problem (Theorem 3, p. 365):

$$\underset{x\in X}{\text{Min}} \sum_{i=1}^{n} f_i(x_i)$$

$$\text{such that} \sum_{i=1}^{n} g_i^j(x_i) \leq c^j \quad j=1,\cdots,k.$$

where f_i, g_j, are functions on some Euclidian space, $E_i = X$. The Shapley-Folkman theorem is applied to obtain the estimate

$$GAP \leq \sum_{i\in J} \alpha(f_i) \tag{7.10}$$

where $\alpha(f_i) = \sup_{x\in X}[f_i(x) - f_i^{**}(x)]$,

f_i^{**} = largest convex function bounding $f_i(x)$ from below,

J is a subset of $\{0, \cdots, n\}$ with $k+1$ elements,

(GAP = duality gap for convex g_i^j, see [44] for details).

In our formulation, X is the set of dynamic constraints, $f_i(x)$ corresponds to $d_i(y_i(k))$ and $\Sigma\, g_i^j(x_i) = 0$ correspond to the interconnection constraints. Thus, we have replaced E_i with a nonconvex constraint set and replaced the inequality constraints with equality constraints. However, these modifications do not change the results given in [44]. Because $d_i(y_i(k))$ is a convex function on X, and the interconnection constraints are convex, use of (7.10) shows there is no duality gap, a fact confirmed by the computational results given in Section 8.

A condition given in [44] as part of the duality gap estimate is that

$$[f_i(x_i) + \sum_{j=1}^{k} g_j(x_i)] \;/\; ||x_i|| \to \infty$$

$$\text{as } ||x_i|| \to \infty.$$

This condition is included to assure the lower-semicontinuity of the perturbation function ([44], Lemma 1, p. 360). In our application, this condition is satisfied with respect to the total density variables since delay approaches infinity as density approaches 1/L. To satisfy this condition with respect to total control variables, we may rewrite the delay on diverge link i as

$$d_i(y_i(k)) = d_i(y_i(k))/2 + d_i(v_i(k) + v_i'(k))/2$$

and thus, not change the cost function. However, the diverge link algorithm (Appendix D) must be suitably modified to account for nonconvexity in the total control variables.

It is interesting to note that if we had written the cost function as

$$\sum_k \sum_i \sum_j d_i(y_i(k)) x_{i,j}(k), \tag{7.11}$$

the same algorithm may be applied to solve the dual problem but a duality gap would appear because of the nonconvexity of $x_{ij}(k)$ as a function of interconnection variables. In fact, (7.11) was simulated and a large duality gap did occur.

The final fundamental question concerning convergence is magnitude of the subgradient, or satisfaction of the interconnection constraints. In the next section,we show the magnitude does not necessarily approach zero.

7.4 Subgradient Magnitude

Under certain conditions, the subgradient does not approach zero but the interconnection constraints can be satisfied without affecting the result.

This situation arises with respect to total density variables in the region $0 \leq y_i(k) \leq y_m(i)$, that is, the total density does not influence the network dynamics but the dual function objective for link i has the form (dropping i subscripts)

$$\phi(y) = cy + \lambda(y - x_o) \tag{7.12}$$

where c, x_o do not depend on y. Thus, the minimum is given by

$$\Phi(\lambda) = \min_y \phi(y) = \begin{cases} -\lambda x_o & c + \lambda \geq 0 \\ (c+\lambda)y_m - \lambda x_o & c + \lambda < 0 \end{cases} \tag{7.13}$$

Consequently, the dual function is linear in λ and non-differenciable at the maximum. The maximum is found from (7.13) to be

$$\max_\lambda \Phi(\lambda) = \Phi(-c) = cx_o. \tag{7.14}$$

At $\lambda = -c$, y can have any value without affecting the cost or dual functions, but note that (7.14) is the cost function for link i.

A similar situation arises in the case of total control interconnection variables. In the region, $0 \leq v_i(k) \leq y_m(i)$ the dual function objective has the form

$$\phi(v) = c/y_m + \mu(v - x_o) \tag{7.15}$$

where c, x_o do not depend on v.

The minimum is given by

$$\min_v \phi(v) = \begin{cases} c/y_m - \mu x_o & \mu \geq 0 \\ \\ c/y_m + \mu(y_m - x_o) & \mu < 0 \end{cases} \tag{7.16}$$

When $x \leq y_m$, we have,

$$\max_\mu \Phi(\mu) = \Phi(0) = c. \tag{7.17}$$

At the optimum, v can have any value without affecting (7.17) and thus, we set $v = x_o$, although the value is immaterial with respect to both cost and dual function values.

8. COMPUTATIONAL STUDY

The algorithm described in Section 6 was simulated for the 4 station, 58 link network, structured by one given in [25] and shown in Fig. 8.1. The traffic flow model for the network contains 54 total density variables (y) and 28 total control variables (v, v'). For the simulation results given here, four vehicle types corresponding to O-D pairs (1,3), (3,1), (2,4), (4,2) were considered. Each subnetwork was chosen to exclude loops so that for example, the only O-D pair (1, 3) subnetwork diverge links are 5 and 15 with all possible paths being the minimum distance path to station 3 (Fig. 8.2). The other vehicle type subnetworks are similarly defined as shown in Figs. 8.3 - 8.5. There is a total of 8 routing control variables.

For a 10 time step problem we have 40 dispatch control variables, 540 density interconnection variables, 280 control interconnection, and 80 routing control variables, giving a total of 940 variables for the subproblem optimization.

8.1 Network Parameters

The simulation was run at a 5 sec. headway, a typical value for a high capacity AGT system. Values for other parameters were:

vehicle length = 3 m,

service acceleration limit = 1.5 m/sec^2,

integration step size = 3 sec,

dock berth dwell time = 6 sec.

The network link characteristics, length and maximum velocity are given below in Table 8.1

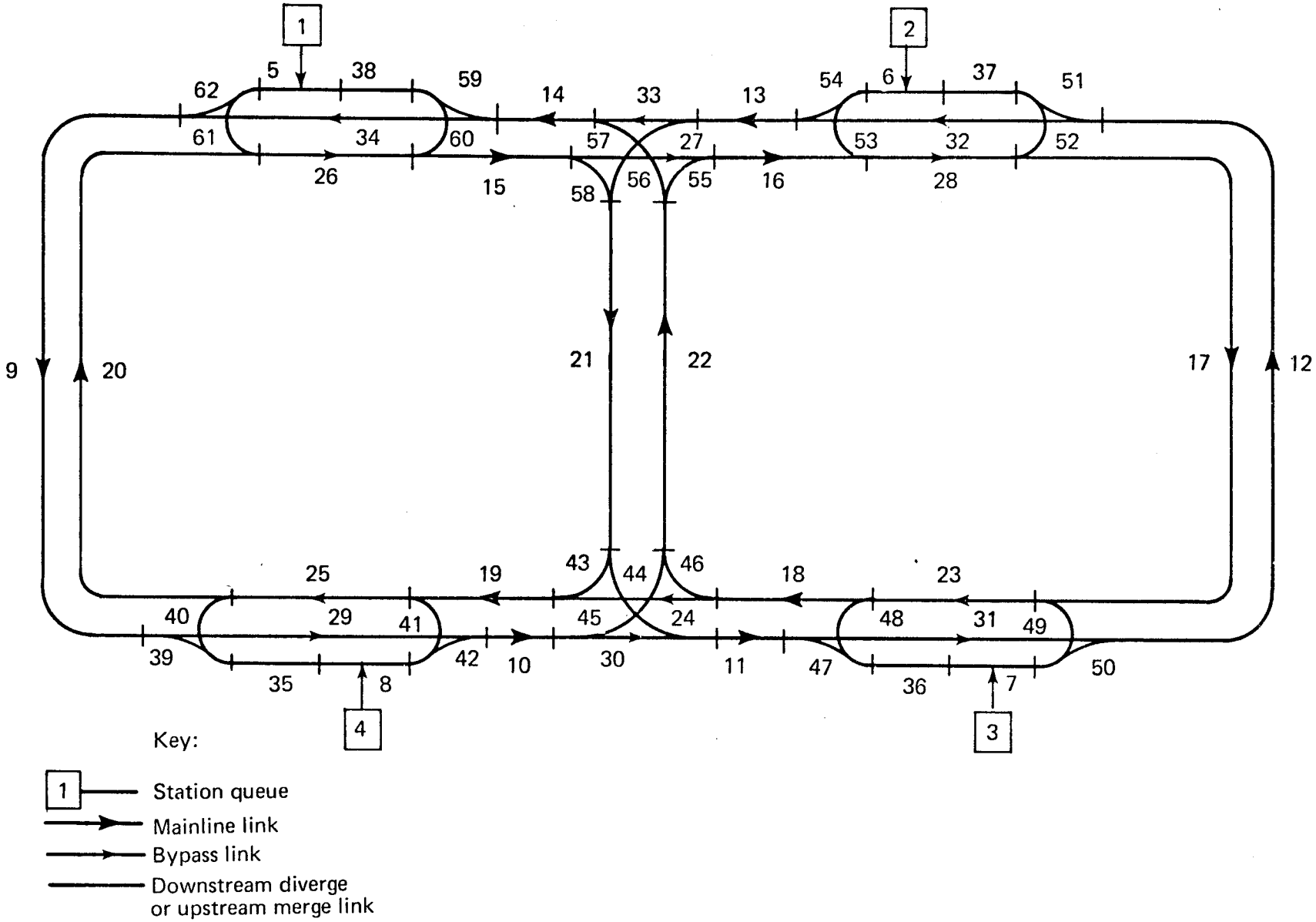

Fig. 8.1 Simulated network.

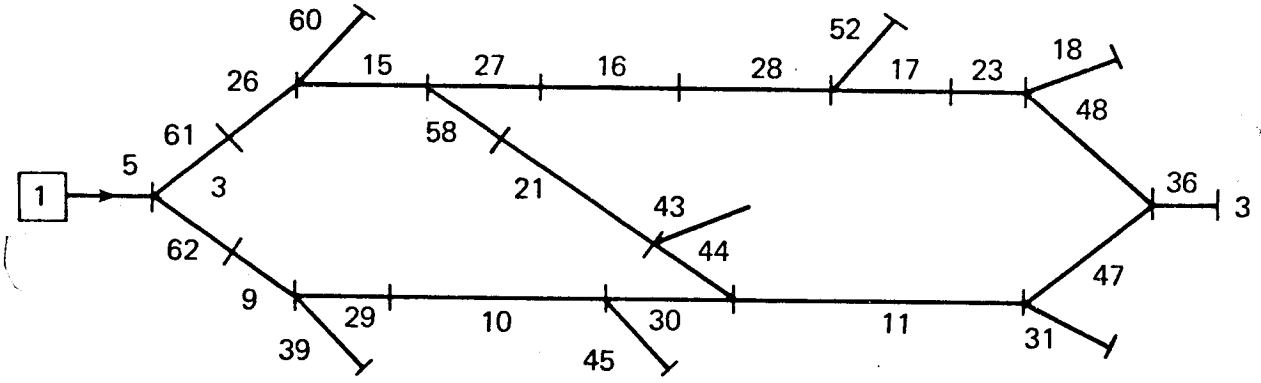

Fig. 8.2 Subnetwork 1 (O-D pair 1, 3).

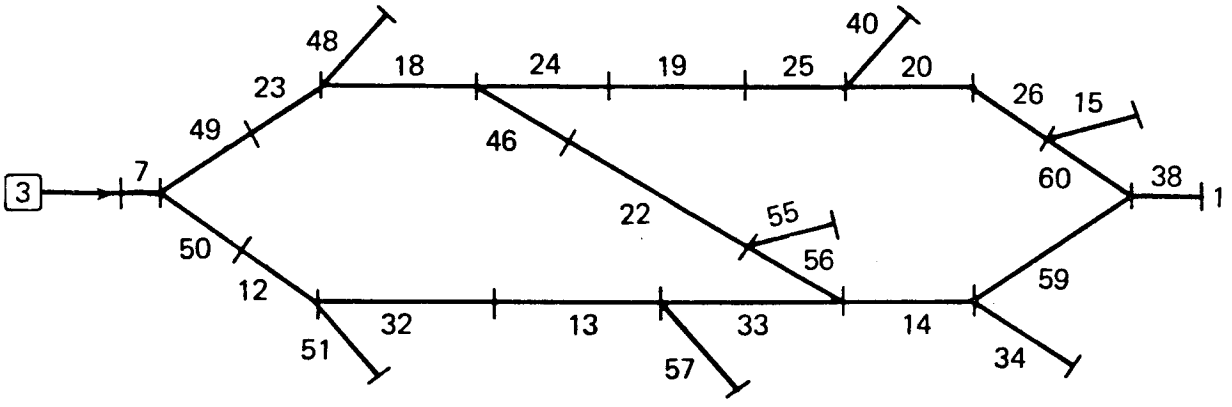

Fig. 8.3 Subnetwork 2 (O-D pair 3, 1).

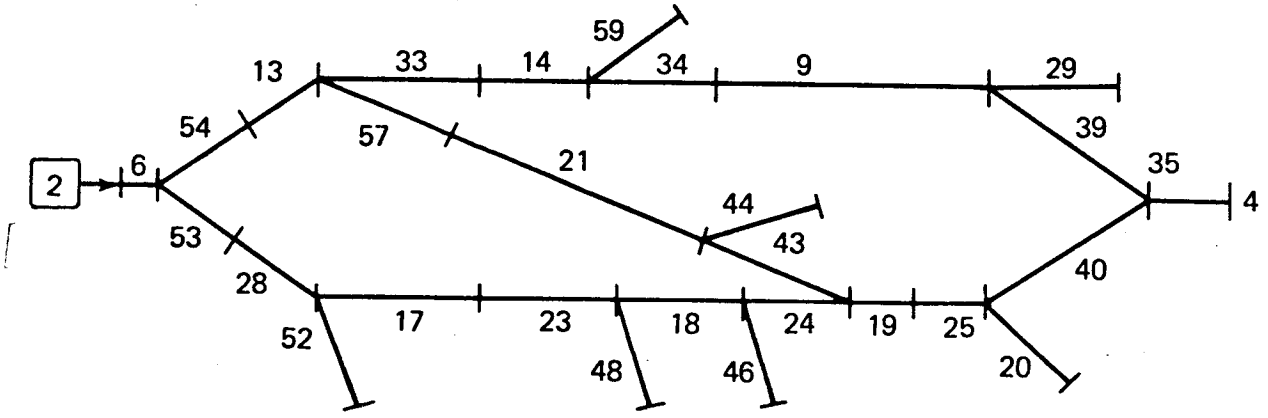

Fig. 8.4 Subnetwork 3 (O-D pair 2, 4).

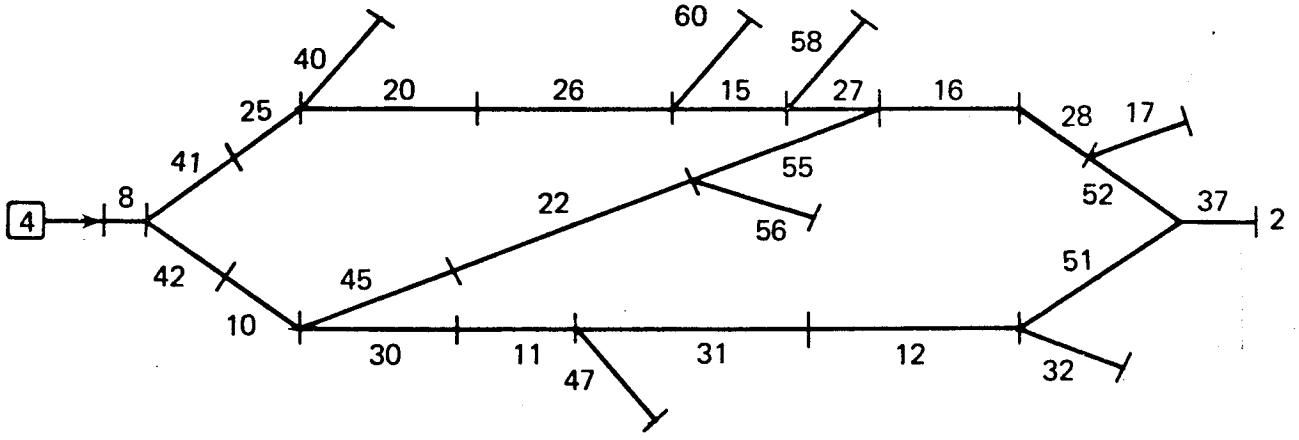

Fig. 8.5 Subnetwork 4 (O-D pair 4, 2).

Link Type	Number	Length (m)	V_{max} (m/s)
Station	1-4	--	—
Egress Queue	5-8	L	15
Mainline	9-22	350	15
Bypass	23-34	150	15
Arrival Queue	35-38	--	15
Merge or Diverge	39-62	75	7

TABLE 8.1 NETWORK LINK CHARACTERISTICS

The demand matrix used is given in Table 8.2 in terms of average times for an exponentially distributed interarrival time.

From Station	To Station 1	2	3	4	Total
1	—	20	5	45	3.9
2	40	--	23	180	13.3
3	26	60	--	51	13.3
4	72	30	360	--	30
Total	12.9	10	4.3	21.2	

TABLE 8.2 AVERAGE INTERARRIVAL TIMES (SEC)

8.2 Computational Parameters

Computational studies were performed to investigate convergence properties of the algorithm. Convergence and computational cost were studied as a function of initial control, step-size selection, Lagrange multiplier initialization, problem duration, initial state, and accuracy of lower level solution. The subproblems were allowed to converge but the upper level was terminated after a given number of iterations,

although convergence is apparent.

The upper level step-size rule employed was:

$$\alpha_n = \begin{cases} \alpha_o & , \quad n \leq n' \\ \alpha_o/(n - n'), & n > n' \end{cases} \tag{8.1}$$

where n is the iteration number and α_o is a constant. For all simulation runs, $n' = 10$.

Two methods for initializing Lagrange multipliers were investigated. The first, termed method 1, is to offset the total density delay cost on each link with the cost due to the multiplier. That is, determine $\lambda_i(k)_o$ such that $d_i(y_i(k)) + \lambda_i(k)\, y_i(k)$ is a minimum for the initial $y_i(k)$. This gives for $y_i(k) > y_m(i)$,

$$\lambda_i(k)_o = - hD_i^2\, y_i(k)(2 - y_i(k)L)/(t_f(1 - y_i(k)L)^2) \tag{8.2a}$$

and for $y_i(k) \leq y_m(i)$,

$$\lambda_i(k)_o = - D_i^2/(t_f\, v_{max}(i)). \tag{8.2b}$$

The multipliers associated with arrival queue total densities are initialized to zero and the total control multipliers $\mu_i(k)$, $\mu_i'(k)$ are initialized to zero.

The second method, termed method 2, is to initialize all multipliers to zero.

Two sets of initial control variables were investigated. They are given below in Table 8.3 with the associated link number and vehicle type. Each set gives the control variable value at all time steps where a 1 indicates diversion to the lower numbered link. The dispatch control variables are initially set so as to dispatch a waiting trip if one exists.

	Link No.	5	15	7	18	6	13	8	10
	Veh. Type	1	1	2	2	3	3	4	4
Set	1	1	1	1	0	0	1	1	1
	2	0	1	0	1	0	1	0	1

TABLE 8.3 INITIAL CONTROL VARIABLES

The accuracy of the lower level optimization was investigated according to two techniques. The first was to halt a link optimization if the resulting values of the variables had not changed when compared to the previous iteration. Thus, as the lower level iterations progress, only those links that have changes in associated variable values are reoptimized. This approach was compared to fully optimizing all links at every lower level iteration. The difference in dual function value was typically less than 1% but cost savings were significant. Thus, the approximation was used for all subsequent simulation runs although it was found that an instability can result if α_o is too large. The approximation is used only for the first 15 iterations.

The second accuracy technique involves computation of the total density interconnection variables. We have noted that the function of y to be optimized is nonconvex. The optimization approach is a linear search taken at uniform steps between $y_m(i)$ and $1/L$ ($y_{max} < 1/L$ was used to save computational cost). A parabolic fit is determined using the minimum and two adjacent points. The minimum of the parabolic fit is used as the solution. The parameter that was varied to investigate the accuracy of this approach was the number of increments, N, used in the initial linear search.

The final parameter investigated was initial state. By method 1, each vehicle type was initialized to a density of .015 veh/m on each link while method 2 increases the initial density to .020 veh/m.

The simulation runs with associated parameter values are listed in Table 8.4. The values of control variable initialization, state variable initialization, method of Lagrange multiplier initialization, and lower level accuracy are indicated by a 1 or 2 as defined by the above discussion. Problem duration is indicated by the number of time steps, each being 3 sec. The results are discussed below.

8.3 Numerical Results

The results of the simulation runs described in Table 8.4 are summarized in Table 8.5. In each case, the initial, final, and maximum dual functions are given because in some cases a maximum dual function value exists prior to the final iteration. In addition, initial and final cost function values are given as well as total run time on an IBM 3033. Note the run times correspond to sequential processing while an implementation would contain a high degree of parallelism.

8.3.1 Lower Level Accuracy

Runs 1, 2, and 3 compare the effects of changing, N, the number of increments in the linear search to optimize with respect to total density interconnection variables. The differences are most apparent in the values of the initial dual function. As expected, increasing the accuracy gives a smaller dual function value. However, the cost savings for N = 5 (i.e., lower run time) versus the degradation in

Table 8.4

Simulation run descriptions

Run	Step size	Initial control variable	Initial state variable	Lower level accuracy (N)	Upper level iterations	Initial lagrange multipliers	Problem duration (time steps)
1	2000	1	1	10	20	1	10
2	2000	1	1	20	20	1	10
3	2000	1	1	5	20	1	10
4	1000	1	1	5	20	1	10
5	4000	1	1	5	20	1	10
6	2000	1	1	5	10	1	10
7	2000	1	1	5	40	1	10
8	2000	1	1	5	5	1	10
9	2000	2	1	5	20	1	10
10	2000	1	1	5	20	2	10
11	500	1	1	5	20	1	20
12	2000	1	2	5	20	1	10

Table 8.5

Numerical results

Run	Dual function			Cost function		Run time (sec)
	Initial	Final	Maximum	Initial	Final	
1	8158	8864	8868	9641	8908	15.23
2	8157	8872	8872	9641	8918	19.88
3	8171	8876	8879	9641	8908	10.38
4	8171	8854	8854	9641	8913	10.27
5	8171	–	–	9641	9062	–
6	8171	8819	8831	9641	8917	5.60
7	8171	8892	8892	9641	8908	20.05
8	8171	8744	8744	9641	8917	3.25
9	7990	8877	8882	9784	8910	9.55
10	2867	8141	8141	9641	8979	11.37
11	6395	7875	8081	9110	8393	40.54
12	15480	16806	16923	18068	16928	12.34

performance suggests that N = 5 be the value of choice for the remainder of the study. Note that N = 5 corresponds to computation of 6 points, twice the minimum number required for a parabolic fit. Typically, only 2 or 3 lower level iterations were needed for convergence.

8.3.2 Step-Size Selection

Runs 3, 4, and 5 show the effects of adjusting α_o. The differences between 3 and 4 are small with the larger step-size, $\alpha_o = 2000$, performing slightly better. However, increasing the step-size to 4000 created an instability because of the approximation discussed in section 8.2 where the lower level iteration is prematurely terminated, that is, the dual function is not accurately computed and the subgradient computed at the upper level is not actually a subgradient. Removing this approximation removed the instability.

8.3.3 Upper Level Convergence

Runs 3, 6, 7, and 8 compare results when 20, 10, 40, and 5 upper level iterations are used. The run times are approximately proportional to the number of upper level iterations. The results for the cost and dual functions are plotted in Fig. 8.6. The cost function is reduced rapidly. After 2 iterations (not shown in Table 8.5) the cost function has a value of 8936 and the dual function is to within 1 percent of the final cost function value (dashed line) after 6 iterations. The total interconnection errors for the total density and total control variables are plotted in Fig. 8.7. The errors are defined by

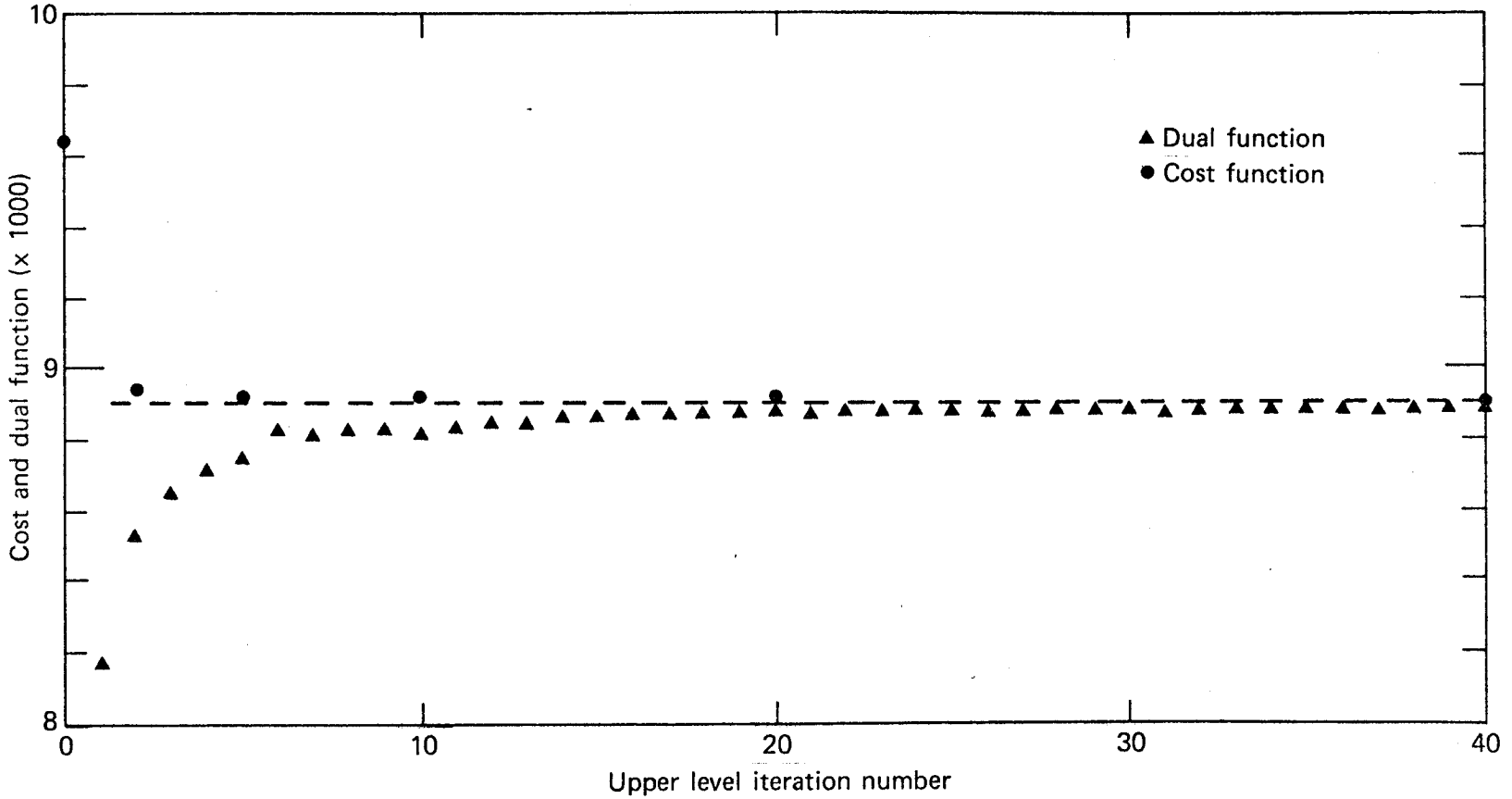

Fig. 8.6 Cost and dual functions - Run 7.

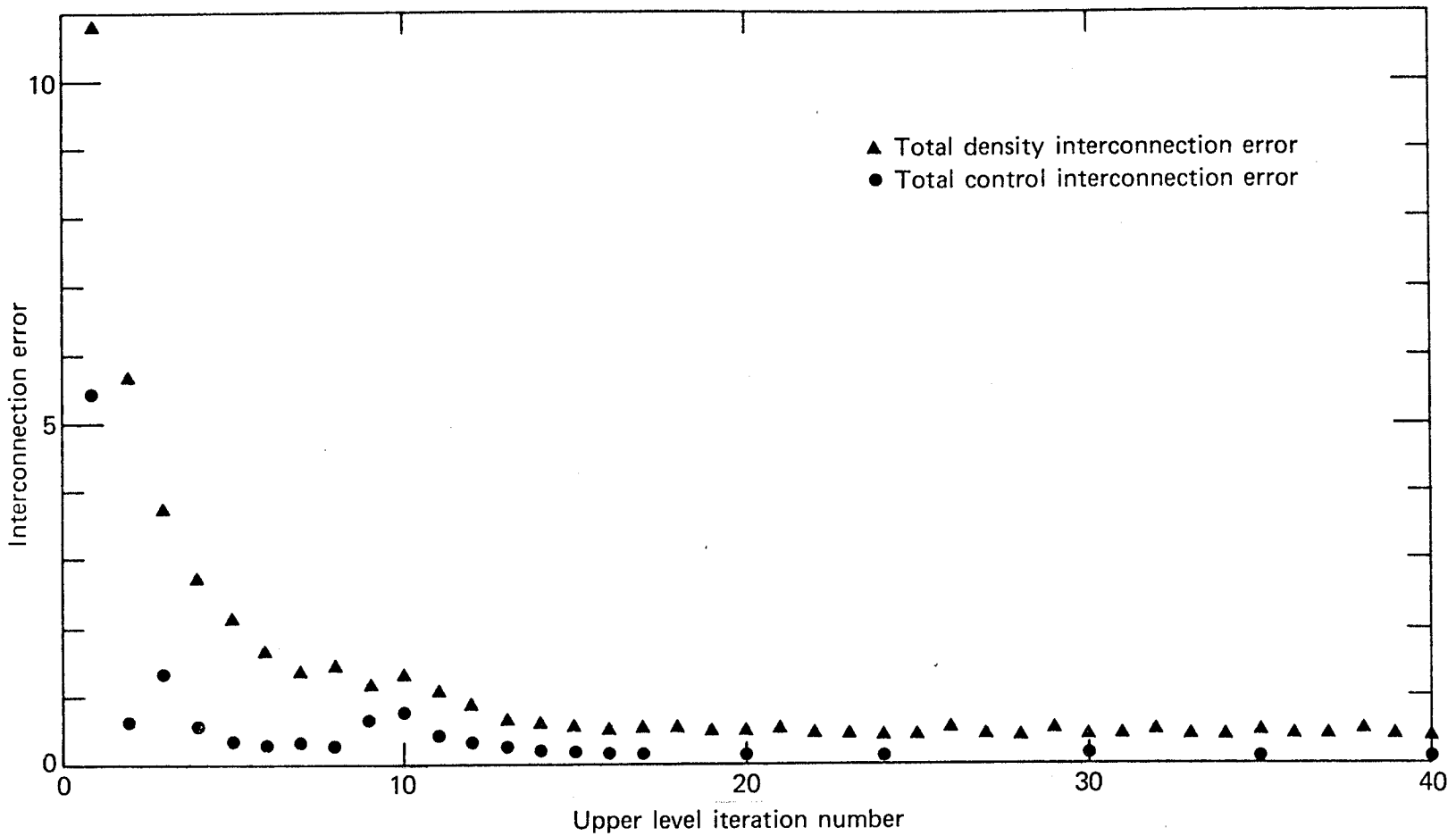

Fig. 8.7 Interconnection error - Run 7.

$$y_\varepsilon = \sum_i \sum_k T\left|y_i(k) - \sum_j x_{i,j}(k)\right|$$

$$v_\varepsilon = \sum_{i\varepsilon C} \sum_k \{T\left|v_i(k) - \sum_j u_{i,j}(k)\, x_{i,j}(k)\right|$$

$$+ T\left|v_i'(k) - \sum_j (1 - u_{i,j}(k))\, x_{i,j}(k)\right|\}$$

The difference in final cost between Run 3 (20 iterations) and Run 6 (10 iterations) is due to the value of the type 2 dispatch control variable at time step 1.

8.3.4 Control Variable Initialization

Run 9 shows the effect of initializing the control variables to the second set given in Table 8.3. The initial cost is slightly higher but all other aspects of convergence are essentially unchanged. However, the final control variables are quite different, indicating either the non-uniqueness of a solution or that several control variables do not significantly affect the cost.

8.3.5 Lagrange Multiplier Initialization

In Run 10, all multipliers are set to zero, in contrast to the previous runs which used method 1. Clearly, our initial approximation (method 1) places the dual function closer to the solution after the first iteration and is therefore, the method of choice.

8.3.6 Problem Duration

In Run 11, the number of time steps is increased to 20. The first observation is that the initial cost decreases as the time increases. This is because vehicles are arriving at stations at a

faster rate than they are departing stations. Thus, as time increases, the time averaged delay decreases. Comparing Run 11 to Run 3 the run time is quadrupled while the number of time steps is doubled. Moreover, the dual function is further from solution in Run 11. It was also found that the step-size, α_o, had to be decreased, otherwise convergence was very poor. That is, the dual function became negatively large before increasing.

8.3.7 Initial State

In Run 12, the initial state for each vehicle type is increased to 0.02 veh/m. Convergence is similar to Run 3 with a slight increase in run time.

9. SUBOPTIMAL CONTROL STRATEGIES

To develop suboptimal strategies we first outline the greatest contributors to computational complexity in the optimal control. These are:

1. The computation of the interconnection variables at the lower optimization level involves a linear search that requires many evaluations of the cost. This is because the cost function as a function of an interconnection variable is of the general form

   $$c_1 y^2/(1 - yL) + c_2/y + c_3 y, \qquad (9.1)$$

   a nonconvex function for certain values of c_1, c_2, c_3. If c_2 were zero, the minimum could be easily computed analytically. Otherwise, the roots of a fourth order polynomial must be determined.

2. The diverge link optimization algorithm is complex because of the nonconvexity of the problem resulting from coupling between routing control variables and interconnection variables.

3. The lower level optimization requires several iterations through the network. This must be repeated many times because the upper level does not generally converge quickly.

In devising suboptimal strategies it may also be useful to keep in mind the relative ease of solving the following associated problem (Appendix D):

$$\underset{u}{\text{Min}} \sum_{k} \lambda^T(k)\, x(k)$$

such that $x(k+1) = [A(k) + \sum_i u_i(k)\, B_i(k)]\, x(k) + r(k)$

where $A(k)$, $B_i(k)$ satisfy the constraints of a transportation network.

The three following strategies are proposed to simplify the computational burden of the optimal control. There is no significant savings in storage requirements.

9.1 Suboptimal I

This strategy is characterized by making the following modifications to the optimal strategy.

1. To compute the y interconnection variables at the lower level a linear least squares fit is made to approximate c_2/y when determining the optimal y. However, the original flow function c_2/y is retained when integrating state and adjoint equations. As a result, a closed form solution for y can be found. That is, the cost function as a function of y has the form

   $$c_1 y^2/(1 - yL) + c_2 y \qquad (9.2)$$

   The optimal y is found by solving the quadratic equation

   $$(c_2 L - c_1)\, Ly^2 + 2(c_1 - c_2 L)y + c_2 = 0$$

   where L is vehicle length and c_1, c_2 are computed at each time step for each link and are functions of the current state and adjoint variables. Note it must be checked if the solution falls within the density variable range and that $c_2 L - c_1 \geq 0$.

2. The diverge link algorithm is simplified by using the density variables that were computed at the previous iteration. That is, the routing control variables are computed based on v_i, v_i', y_{i+1}, y_{i+2}, computed at the previous iteration. The y_{i+1}, y_{i+2} variables are re-computed using the new routing control variables while v_i, v_i' are computed at the upper level using the newly computed state and routing control variables.

3. The adjoint variables are computed at the upper level and the lower level is optimized using these adjoint variables without iteration. Thus, at the lower level we solve the K problems for k = 1, ···, K

$$\underset{u(k),\, y(k)}{\text{Min}} \sum_i [d_i(y_i(k)) + \lambda_i(k)\, y_i(k) - \sum_{\ell=k}^{K} \lambda_i(\ell) \sum_j x_{i,j}(\ell)] \tag{9.3}$$

rather than

$$\underset{u,y}{\text{Min}} \sum_k \sum_i [d_i(y_i(k)) + \lambda_i(k)(y_i(k) - \sum_j x_{i,j}(k))].$$

Thus, the lower level problem gives an approximate dual function and approximate subgradient.

9.1.1 Suboptimal I Algorithm

The algorithm for Suboptimal I at the n^{th} iteration is:

Upper Level Computation:

$$\lambda_i^n(k) = \lambda_i^{n-1}(k) + \alpha^n [y_i^{n-1} - \sum x_{i,j}^{n-1}(k)]$$

$$v_i^n(k) = \sum_j u_{i,}^{n-1}(k)\, x_{i,j}^{n-1}(k)$$

$$v'^{n}_{i}(k) = \sum_{j} (1 - u^{n-1}_{i,j}(k))\, x^{n-1}_{i,j}(k)$$

compute adjoint variables $p^{n}_{i,j}(k)$ as a function of u^{n-1}, y^{n-1}, v^{n}, v'^{n}.

Lower Level Computation:

Solve (9.3) where

$$x_j(k+1) = [A_j(k) + \sum_{i \in C_j} u_{i,j}(k)\, B^{n}_{i,j}(k)]\, x_j(k) + r_j(k)$$

and $A_j(k)$ is a linear function of $y_i(k)$, $i \in L_j$

$B^{n}_{i,j}(k)$ is evaluated at v^{n}_{i}, v'^{n}_{i}, y^{n-1}_{i+1}, y^{n-1}_{i+2}

when computing u^{n}. The variables y^{n}_{i+1}, y^{n}_{i+2} are computed at u^{n}.

9.2 Suboptimal II

This control strategy is derived as a further simplification of Suboptimal I. That is, the constraint dynamics use interconnection variables computed at the previous iteration while the upper level computation is identical to Suboptimal I. The lower level problem is:

$$\underset{u,y}{\text{Min}} \sum_{k} \sum_{i} [d_i(y_i(k)) + \lambda^{n}_{i}(k)\, (y_i(k) - \sum x_{i,j}(k))]$$

s.t. $$x_j(k+1) = [A^{n-1}_{j}(k) + \sum_{i \in C_j} u_{i,j}(k)\, B^{n-1}_{i,j}(k)]\, x_j(k) + r_j(k)$$

where $A^{n-1}_{j}(k)$, $B^{n-1}_{i,j}(k)$ are evaluated at previous values, y^{n-1}, v^{n}, v'^{n}.

Note the minimum at the lower level is the desired minimum rather than the K step approximation in Suboptimal I. This is because the dynamics in Suboptimal II are only a function of time and the results in Appendix D apply.

9.3 Suboptimal III

The final simplification is to adjust all interconnection variables at the upper level thereby eliminating use of the duality concept. The lower level problem corresponds to the simplified problem in Appendix D. The algorithm is:

Upper Level:

$$y_i^n(k) = \Sigma\, x_{i,j}^{n-1}(k)$$

$$v_i^n(k) = \Sigma_j\, u_{i,j}^{n-1}(k)\, x_{i,j}^{n-1}(k)$$

$$v_i'^n(k) = \Sigma_j\, (1 - u_{i,j}^{n-1}(k))\, x_{i,j}^{n-1}(k)$$

Lower Level:

$$\underset{u}{\text{Min}} \; \Sigma_k \Sigma_i \Sigma_j\, d_i(y_i(k))\, x_{i,j}(k)$$

$$\text{s.t. } x_j(k+1) = [A_j^n(k) + \underset{i \in C_j}{\Sigma} u_{i,j}(k)\, B_{i,j}^n(k)]\, x_j(k) + r_j(k).$$

9.4 Computational Results

The three proposed suboptimal strategies were simulated for the network and conditions described in Section 8. Specifically, Run 3 in Table 8.4 was repeated for the three suboptimal schemes. The results are given below in Table 9.1.

	Final Dual	Final Cost	Run Time (sec)
Optimal	8876	8908	10.38
Suboptimal I	8804	8918	3.91
Suboptimal II	8694	8913	3.71
Suboptimal III	--	8941	2.39

TABLE 9.1 SUBOPTIMAL COMPUTATIONAL RESULTS

Each of the suboptimal strategies have significantly smaller computational time than the optimal strategy. Although the dual functions for I and II (not actually dual functions but approximations) are not as large as that attained in the optimal algorithm, the final cost functions have nearly the same value. In Section 10, it is noted that an implementation would not include a dual function computation but the upper level iteration number may be determined by apriori analysis. Thus, the significance of Table 9.1 is that the suboptimal strategies nearly attain the optimal cost function value in the same number of iterations as the optimal algorithm. In suboptimal I it was found that the gap between the dual and cost functions at the 20^{th} iteration can be eliminated if the step-size, α_o, is increased to 4000.

Only suboptimal III has significantly worse performance than the optimal. Moreover, suboptimal III does not converge but oscillates between the values 8940.88 and 8940.44.

Most importantly, because there is no iteration at the lower level, run times for the suboptimal strategies are directly proportional to the number of time steps in the problem (i.e., doubling the time steps doubles the run time) versus the dramatic increase seen in the optimal algorithm.

10. CONTROL IMPLEMENTATION

In this section we address the question of how to implement the algorithms given in previous sections. Many of the specific questions must be resolved through apriori analysis of a particular network. That is, many of the approximations, suboptimal designs, computer configurations, software designs, and communication requirements are highly network dependent. Here, we outline some general approaches and possibilities that should be considered. In particular, algorithm initiation criteria, stopping criteria, and computer storage and speed requirements are examined.

First, we note the overall intent of this work is to develop on-line control algorithms. However, for certain networks it may be possible to devise routing tables using the given algorithms as part of an apriori analysis. That is, a nominal routing strategy might be developed for a nominal set of demands with relatively simple rules for adaptation to off-nominal cases. The adequacy of this type of approach must be evaluated for a specific network. In the following sections, the on-line approach is discussed.

10.1 Algorithm Initiation

Under non-failure conditions the operation of the automated transit network is essentially deterministic except for unexpected changes in demand. Thus, as long as the trip demands correspond to the predicted demands, a nominal set of routing control variables computed apriori may be used. However, as actual demand departs from the predicted demand, the nominal control may degrade in

performance such that on-line control computation should be invoked. Thus, the question of how to determine when the control should be re-optimized, arises.

A simple technique is to continuously run the algorithm using current demand and future predicted demands. Thus, known changes in demand are taken into account.

A second technique, intended to reduce communication requirements, is to initiate control computations only when it is needed. This may be accomplished by computing the changes in cost function value associated with each subnetwork due to changes in demand. The value of the cost function due to vehicle type j dispatch, $u_j(k)$, has the form

$$c(u_j(k)) = u_j(k)[p_{1,j}(k+1) - p_{2,j}(k+1) + \bar{p}_n(k+1)] + C \quad (10.1)$$

where subscript 1 represents the trip request queue, subscript 2 represents the egress queue, $\bar{p}_n$ is the adjoint variable associated with the arrival queue of vehicle types with origin station being the destination station of j, and C is a constant not depending on $u_j(k)$. Thus, each station may compute the change in cost due to a dispatch of vehicle types departing that station. At each time step, if $u_j(k)$ differs from the nominal $u_j^o(k)$ then the state and adjoint equations must be integrated to account for the future effects of $u_j(k)$.

Whenever a change in demand causes the cumulative change in cost to cross a specified threshold, then a particular station can initiate a new control computation. As the algorithm progresses from the station that initiates the re-optimization, it must be determined whether subnetworks emanating from other stations need to be

re-optimized. Again, there are several options as to how to accomplish this task. As the total densities on common links change, this information eventually reaches other stations through transmission of adjoint variables. Thus, other stations may initiate re-optimization of associated subnetworks according to the same criterion, namely, when change in cost function passes a specified threshold. A second approach would be to re-optimize all portions of the network affected by the initiating subnetwork. Because communication links are distributed this must be accomplished by a flag being passed between adjacent links, eventually reaching the origin station of each affected subnetwork.

Thus, the general approach we have outlined is that a decision whether or not to initiate re-optimization is made at the subnetwork level. The effects of re-optimization eventually reaches the other subnetworks which in turn, make a decision. For a specific network, synchronization and time delay problems must be investigated prior to implementation.

10.2 Stopping Criteria

The stopping criterion typically used when applying duality is that the error between the dual and cost function values be within a specified level. For the distributed implementation described here, this is clearly impossible and an alternate criterion is needed. As in algorithm initiation, decisions must be made at either the subnetwork or link level.

One possible approach is to select a fixed number of upper level iterations and accept the result. A number that is found to be satis-

factory over a wide range of conditions would be determined via apriori analysis.

Another approach is to decide at the link level whether or not to continue upper level iterations based on satisfaction of the interconnections constraints. The need to continue iteration must be transmitted through the subnetwork and network.

These approaches can also be applied to solution of the lower level problems where the criterion used is the change in each link solution from the previous iteration.

10.3 Computer Storage Requirements

The arrangement of computational facilities is also a network dependent variable although it can be expected that a single wayside computer would handle a small portion of the network involving several links. To estimate the storage requirements for algorithm implementation we consider the requirements for variables associated with each link. Note this may underestimate actual requirements because state and adjoint variables transmitted from adjacent links must be stored prior to computation. On the other hand, a single computer would typically be responsible for several links and therefore, intermediate storage for these links is not required. Thus, we only give the single link requirements with a rough approximation of total requirements being the sum over all individual links.

The storage requirement for a non-diverge link is estimated as follows. Let J be the number of vehicle types that pass through a given link and K be the number of time steps in the optimization time interval. Then, for each link the variables that must be stored are:

1. J vehicle type density variables (computed as needed at each time step),
2. K total density variables - stored for adjoint integration and other computations,
3. JK adjoint variables - stored for forward control computation,
4. K Lagrange multiplier's - stored for adjoint integration,
5. $\approx$10 miscellaneous variables such as headway, integration time step, subgradient step-size, vehicle length, etc.,
6. storage space for intermediate computation.

Neglecting space needed for (5) and (6) the storage requirement is approximately $J + K(J + 2)$. Thus, the major contributor to storage requirement is the JK term, that is, the adjoint variables.

The requirement for a diverge link is increased due to the additional variables involved. Namely, a maximum of JK routing control variables (the variables for some vehicle types are fixed in time according to subnetwork definition), 2K total control variables, and 2K additional Lagrange multiplier variables. Thus, the total requirement is approximately, $J + K(2J + 6)$.

10.4 Speed Requirements

Speed requirements must be determined for a specific network. That is, the delays, due to computation, that still maintain a specified level of network performance places requirements on wayside computer speed.

The suboptimal strategy run times given in Section 9 appear plausible for implementation. That is, the time for 20 upper level iterations is approximately 10 percent of control time history. Although these times correspond to a computer that is considerably faster than the typical mini-computer, the following caveats should be taken into account.

First, computations for the simulated algorithm are performed sequentially while an implementation would take advantage of parallel processing. Although the degree of parallelism depends upon the precise sequencing and communication of information for a given network, the possibilities for parallel computation are the overall subnetwork problem and computation of variables for each link. Only state and adjoint information must be processed sequentially, being passed through each subnetwork from origin to destination and destination to origin.

Second, the coding used for this simulation was not optimized in all respects.

Third, it is possible the upper level could be terminated sooner than 20 iterations.

Finally, the simulation contains many "bookkeeping chores" that can be "handwired" into an actual implementation.

11. CONCLUSIONS AND FURTHER RESEARCH

The major result of this work is the development of a routing control algorithm with potential for on-line application. The algorithm is based on a traffic flow model derived for vehicle-follower systems and a cost function representing total, time averaged, travel time. The traffic flow model is a first attempt at formulating and applying a macroscopic model for vehicle-follower automated systems. Furthermore, the model has been compared to a number of alternatives and tested with discrete vehicle simulations of several network elements.

The most practical feature of the algorithm is the distributed computational structure. By applying duality, the optimization problem is decomposed into parallel subnetwork problems. Communication links needed for solution of the subnetwork problems correspond to communication links in place for vehicle spacing and velocity regulation, that is, connection between adjacent guideway sections. The control coordinating subnetwork problems is also localized to each link. Thus, routing control is completely decentralized, as is the vehicle-follower control strategy.

Development of the suboptimal strategies enhances the possibilities for on-line implementation of a routing algorithm although design refinements and evaluation of these approaches should involve consideration of a specific network.

Suggested areas of research for extension of this work are,

(1) further development of the suboptimal strategies including analytic investigation of performance degradation,

(2) extended formulation of the problem to include systems based on multiple party vehicle occupancy and intermediate stops,

(3) comparison to simpler but more heuristic approaches,

(4) testing of the flow model in a discrete vehicle simulation of a network, and performance evaluation of the algorithm when used with a discrete vehicle network,

(5) improvement of the model to include discrete vehicles, that is, each link can be modeled as a queue of discrete vehicles with motion through the queue being represented by an aggregate flow model,

(6) reformulation of the problem where each vehicle type is associated with a destination rather than an origin-destination pair.

The last point (6) can produce significant savings in communication and storage requirements although increased complexity of the subnetworks will create longer delays, a trade-off that needs to be investigated.

APPENDIX A

SUBNETWORK DYNAMIC EQUATIONS

The dynamic equations of the network are derived in Section 4 and verified with the aid of computer simulation. It is assumed the subnetwork link may be classified as one of the following types:

1. waiting passenger queue
2. egress queue - diverge
3. egress queue - link
4. link
5. diverge
6. downstream diverge
7. alternate downstream diverge
8. merge
9. station arrival queue - merge
10. station arrival queue - link

Note the above list does not include for example, a link that is both a merge and a diverge. This is because the list only covers those links in the subnetworks being studied (Section 8). The list, however, can be expanded to include any link type. Also note that there may be network link types that no subnetwork contains. This is the case for the network in Section 8 which, in fact, contains many merge-diverge links.

The state equations for type j are written in discrete time with the j subscripts dropped for notational convenience. The integration step-size is T and s_j indicates the waiting queue for trip type, j.

1. waiting trip queue

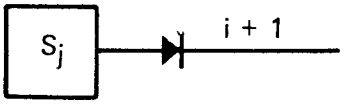

Fig. A.1 Waiting trip queue.

$$x_i(k+1) = x_i(k) + r(k) - u_d(k)$$

where $r(k)$ = number of new trip requests of type j at time k,

$u_d(k)$ = dispatch of vehicle type j at time k

2. egress queue - diverge and downstream links

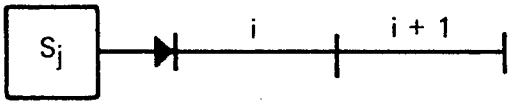

Fig. A.2 Egress queue - link.

$$x_i(k+1) = u_d(k)$$

$$x_{i+1}(k+1) = x_{i+1}(k) + u_i(k)x_i(k)/D_{i+1} - Ta_{i+1,i+3}(k)x_{i+1}(k)/D_{i+1}$$

$$x_{i+2}(k+1) = x_{i+2}(k) + (1-u_i(k))x_i(k)/D_{i+2}$$

$$- Ta_{i+2,i+4}(k)x_{i+2}(k)/D_{i+2}$$

3. egress queue - link

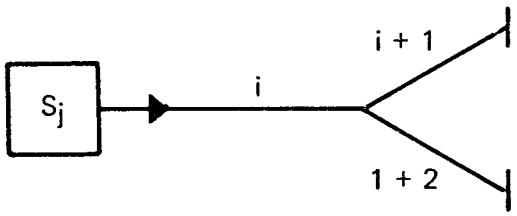

Fig. A.3 Egress queue - diverge.

$$x_i(k+1) = u_d(k)$$

$$x_{i+1}(k+1) = x_{i+1}(k) + x_i(k)/D_{i+1} - Ta_{i+1,i+2}(k)x_{i+1}(k)/D_{i+1}$$

4. link

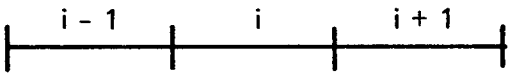

Fig. A.4 Link.

$$x_i(k+1) = x_i(k) + T[a_{i-1,i}(k)x_{i-1}(k) - a_{i,i+1}x_i(k)]/D_i$$

where $a_{i-1,i}(k)$ = velocity from i-1 to i

$a_{i,i+1}(k)$ = velocity from i to i+1

(these are defined in Section A.1)

5. diverge

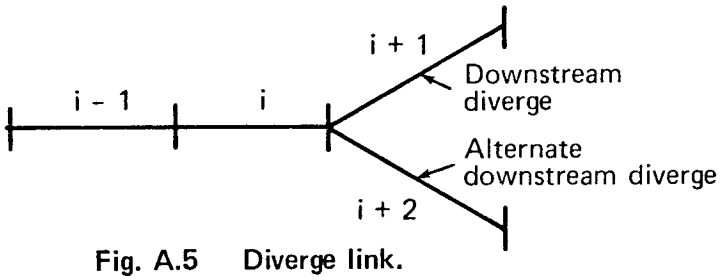

Fig. A.5 Diverge link.

$$x_i(k+1) = x_i(k) + T[a_{i-1,i}(k)x_{i-1}(k)$$

$$- [a_{i,i+1}(k)u_i(k) + a_{i,i+2}(k)(1-u_i(k))]x_i(k)]/D_i$$

6. downstream diverge

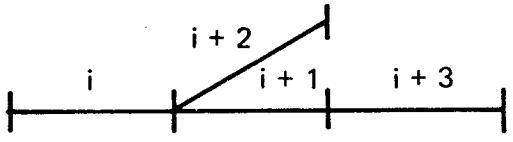

Fig. A.6 Downstream diverge link.

$$x_{i+1}(k+1) = x_{i+1}(k) + T[a_{i,i+1}(k)x_i(k)u_i(k)$$

$$- a_{i+1,i+3}(k)x_{i+1}(k)]/D_{i+1}$$

By convention, the "downstream" diverge link is assigned the lower link number.

7. alternate downstream diverge link

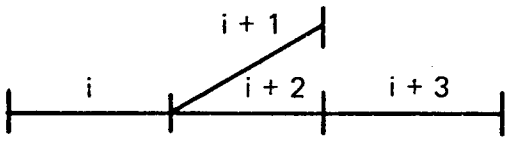

Fig. A.7 Alternate downstream diverge link.

$$x_{i+2}(k+1) = x_{i+2}(k) + T[a_{i,i+2}(k)x_i(k)(1-u_i(k))$$

$$- a_{i+2,i+3}(k)x_{i+2}(k)]/D_{i+2}$$

8. merge

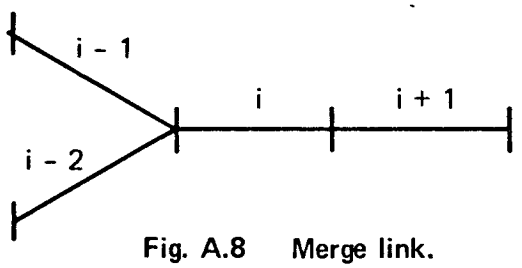

Fig. A.8 Merge link.

$$x_i(k+1) = x_i(k) + T[a_{i-1,i}(k)x_{i-1}(k) + a_{i-2,i}(k)x_{i-2}(k)$$

$$- a_{i,i+1}(k)x_i(k)]/D_i$$

9. station arrival queue - merge

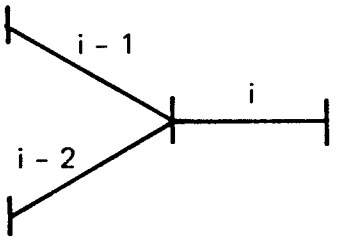

Fig. A.9 Station arrival queue - merge.

$$x_i(k+1) = x_i(k) + T[a_{i-1,i}(k)x_{i-1}(k) + a_{i-2,i}(k)x_{i-2}(k)] - \bar{u}_d(k)$$

where $\bar{u}_d(k)$ = vehicle type dispatch control that returns to j source station

10. station arrival queue - link

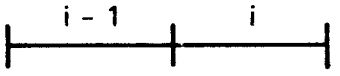

Fig. A.10 Station arrival queue - link.

$$x_i(k+1) = x_i(k) + Ta_{i-1,i}(k)x_{i-1}(k) - \bar{u}_d(k) \quad .$$

A.1 Velocity Functions

The velocity functions denoted by $a(\cdot)$ in Section 4 are defined separately for a link, diverge, and arrival queue. We give the revised function derived in Section 5 for the optimal control formulation.

A.1.1 Link

$$a_{i,i+1}(k) = \begin{cases} q_m(i+1)/y_m(i) & \text{for } y_i \le y_m(i),\ y_{i+1} \le y_m(i+1) \\ q_m(i+1)/y_i & \text{for } y_i > y_m(i),\ y_{i+1} \le y_m(i+1) \\ (1-y_{i+1}L)/(hy_m(i)) & \text{for } y_i \le y_m(i),\ y_{i+1} \ge y_m(i+1) \\ q_m(i+1)/y_i + L(y_m(i) - y_{i+1})/(hy_m(i)) & \text{for } y_i > y_m(i),\ y_{i+1} > y_m(i+1) \end{cases}$$

where y_i, y_{i+1} are at time k,

$q_{max}(i)$ = maximum flow on link i

$y_m(i)$ = density at maximum flow

L = vehicle length

h = headway.

A.1.2 Diverge

Define $a_{i,i+1}(k) = a(y_i, y_{i+1})$ given above.

diverge to downstream:

$$a_{i,i+1}(k) = a(v_i, y_{i+1})$$

diverge to alternate downstream:

$$a_{i,i+2}(k) = a(v_i', y_{i+2})$$

A.1.3 Station Arrival

$$a_{i,i+1}(k) = \begin{cases} q_m(i+1)/y_m(i) & \text{for } y_i \le y_m(i),\ y_{i+1} \le y_m(i+1) \\ q_m(i+1)/y_i & \text{for } y_i > y_m(i),\ y_{i+1} \le y_m(i+1) \\ v_a/(y_m(i)(hv_a+L)) & \text{for } y_i \le y_m(i),\ y_{i+1} > y_m(i+1) \\ q_m(i+1)[(1/y_i-(1/y_m(i))] + v_a/(y_m(i)(hv_a+L)) & \text{for } y_i > y_m(i),\ y_{i+1} > y_m(i+1) \end{cases}$$

$$v_a = \begin{cases} (2a_s(y_{max}(i)-y_{i+1})(L+\delta))^{\frac{1}{2}}, & y_{i+1} > y_m(i+1) \\ v_{max}(i), & y_{i+1} \le y_m(i+1) \end{cases}$$

δ = nose to tail distance between vehicles parked in arrival queue

$y_m(i) = y_{max}(i) - v_{max}^2(i)/(2a_s(1+\delta))$

$q_m(i+1) = v_{max}(i+1)/(hv_{max}(i+1)+L)$

where y_i, y_{i+1} are at time k.

$y_{max}(i)$ = maximum number of vehicles allowed into arrival queue

$v_{max}(i+1)$ = maximum velocity

a_s = service acceleration limit.

The velocity function for a merge is given by the link velocity function.

APPENDIX B

SUBNETWORK ADJOINT EQUATIONS AND COST RELATIONSHIPS

The equations used to compute cost, $c(\cdot)$, due to vehicle type j at time k according to link type are as follows: (link designations correspond to Fig. A.1 - A.10).

1. waiting trip queue

$$p_i(k) = p_i(k+1) - T$$

$$c(u_d(k)) = \begin{cases} (p_i(k+1) - p_{i+1}(k+1) + \bar{p}_n(k+1))u_d(k) & \text{(occupied vehicle)} \\ (p_{i+1}(k+1) + \bar{p}_n(k+1))u_d(k) & \text{(empty vehicle)} \end{cases}$$

where $\bar{p}_n$ = adjoint variable associated with arrival queue of vehicle type that has origin station the destination station of j.

2. egress queue - diverge

$$p_i(k) = p_i(k+1) + u_i(k)p_{i+1}(k+1)/D_{i+1} + (1-u_i(k))p_{i+2}(k+1)/D_{i+2}$$

$$c(u_i) = u_i(k)(p_{i+2}(k+1)/D_{i+2} - p_{i+1}(k+1)/D_{i+1})$$

3. egress queue - link

$$p_i(k) = p_i(k+1) + p_{i+1}(k+1)/D_{i+1}$$

4. link

$$p_i(k) = p_i(k+1) - T\{a_{i,i+1}(k)\,[p_i(k+1)/D_i - p_{i+1}(k+1)/D_{i+1}]$$

$$+ d_i(y_i(k)) - \lambda_i(k)\}$$

$$c(y_i(k)) = T\{a_{i-1,i}(k)x_i(k)\,[p_{i-1}(k+1)/D_{i-1} - p_i(k+1)/D_i]$$

$$+ a_{i,i+1}(k)x_i(k)\,[p_i(k+1)/D_i - p_{i+1}(k+1)/D_{i+1}]$$

$$+ \lambda_i(k)y_i(k) + d_i(y_i(k))\}$$

5. diverge

$$p_i(k) = p_i(k+1) - T\{(a_{i,i+1}(k)u_i(k)$$

$$+ a_{i,i+2}(k)(1-u_i(k)))p_i(k+1)/D_i$$

$$- a_{i,i+1}(k)u_i(k)p_{i+1}(k+1)/D_{i+1}$$

$$- a_{i,i+2}(k)(1-u_i(k))p_{i+2}(k+1)/D_{i+2} - \lambda_i(k)\}$$

$$c(u_i) = u_i(k)T\{x_i(k)(a_{i,i+1}(k) - a_{i,i+2}(k))p_i(k+1)/D_i$$

$$- x_i(k)a_{i,i+1}(k)p_{i+1}(k+1)/D_{i+1}$$

$$+ x_i(k)a_{i,i+2}(k)p_{i+2}(k+1)/D_{i+2}$$

$$+ (\mu_i'(k) - \mu_i(k))x_i(k)\}$$

$$c(v_i(k)) = T\{a_{i,i+1}(k)u_i(k)x_i(k)p_i(k+1)/D_i$$

$$-a_{i,i+1}(k)u_i(k)x_i(k)p_{i+1}(k+1)/D_{i+1} + \mu_i(k)v_i(k)\}$$

The relationship for $v_i'(k)$ is identical in form to $v_i(k)$

$$c(y_i(k)) = T\{a_{i,i+2}(k)(1-u_i(k))x_i(k)\ [p_i(k+1)/D_i$$

$$- p_{i+2}(k+1)/D_{i+2}]$$

$$+ a_{i,i+1}(k)u_i(k)x_i(k)\ [p_i(k+1)/D_i - p_{i+2}(k+1)/D_{i+2}$$

$$+ \lambda_i(k)y_i(k) + d_i(y_i(k))\}$$

The downstream diverge and alternate downstream diverge fall into either the link or diverge category.

8. merge

$$p_i(k) = \text{(same as 4)}$$

$$c(y_i(k)) = T\{a_{i-1,i}(k)x_{i-1}(k)\ [p_{i-1}(k+1)/D_{i-1} - p_i(k+1)/D_i]$$

$$+ a_{i-2,i}(k)[p_{i-2}(k+1)/D_{i-2} - p_i(k+1)/D_i]$$

$$+ a_{i,i+1}(k)x_i(k)\ [p_i(k+1)/D_i - p_{i+1}(k+1)/D_{i+1}]$$

$$+ d_i(y_i(k)) + \lambda_i(k)y_i(k)\}$$

9. station arrival queue - merge

$$p_i(k) = p_i(k+1) + T\lambda_i(k)$$

$$c(y_i(k)) = T\{a_{i-1,i}(k)x_{i-1}(k)\ [p_{i-1}(k+1)/D_{i-1} - p_i(k+1)]$$

$$+ a_{i-2,i}(k)x_{i-2}\ [p_{i-2}(k+1)/D_{i-2} - p_i(k+1)]\}$$

10. station arrival queue - link

$$p_i(k) = \text{(same as 9)}$$

$$c(y_i(k)) = Tx_{i+1}(k)a_{i-1,i}(k)\ [p_{i-1}(k+1)/D_{i-1} - p_i(k+1)/D_i]$$

APPENDIX C

DIVERGE LINK ALGORITHM

The minimization problem for diverge link variables at time k has the general form (see (5.20))

$$\text{Min } [f(z) + c^T(z)u] \tag{C.1}$$

$$\text{s.t. } 0 \leq u \leq 1$$

$$a \leq z \leq b \quad .$$

where u is a vector of routing control variables and z is a vector of interconnection variables.

Because of the relative simplicity of the optimization problem when either u or z are held constant, a projection type optimization technique is suggested. That is,

$$\min_{z \in Z} \{f(z) + \min_{u} [c^T(z)u \text{ s.t. } 0 \leq u \leq 1]\} \tag{C.2}$$

We know at an optimal solution, each $u_i = 0$ or $u_i = 1$. Thus, a straightforward approach would be to minimize over Z for each possible u. However, the number of possible control vectors is 2^J and such an approach would be computationally prohibitive for a large network. To obtain a solution we use the following relaxation procedure.

Given an initial u_o, determine y_o such that

$$f(z_o) + c^T(z_o)u_o \leq f(z) + c^T(z)u_o, \; z \in Z \quad . \tag{C.3}$$

This problem is relatively easy because of the separability of f(z) and c(z). The solution pair (z_o, u_o) is optimal if

$$c^T(z)u_o \leq c^T(z)u, \quad u\varepsilon U, \quad z\varepsilon Z \tag{C.4}$$

because substituting (C.4) into (C.3) gives

$$f(z_o) + c^T(z_o)u_o \leq f(z) + c^T(z)\, u \tag{C.5}$$

$u\varepsilon U$, $z\varepsilon Z$.

If (C.4) does not hold, then a u_1 is generated and (C.3) is repeated for u_o, u_1. Because of the nonconvexity of the problem, we cannot justify dropping u_o from consideration. The process repeats until (C.4) is satisfied. The algorithm coverges in a finite number of iterations because of the finite number of control vector z, u.

The specific algorithm at the n^{th} iteration for time step k is as follows.

1. Given $u^n_{i,j}$ n=1, ..., N determine v^N_i, v^N_i, y^N_{i+1}, y^N_{1+2} by solving the following subproblems for n=1, ..., N:

$$f^n_1 = \min_{v_i, v_i'}[c^n_1 q_m(i+1)/v_i + \mu_i v_i + d_i(v_i + v_i')$$

$$+ c^n_2 q_m(i+2)/v_i' + \mu_i' v_i']$$

$$f^n_2 = \min_{y_{i+1}} [c^n_1 L(y_m(i+1) - y_{i+1})/(h y_m(i))$$

$$+ c_4 q_m(i+3)/y_{i+1} + c_6 d_{i+1}(y_{i+1}) + \lambda_{i+1} y_{i+1}]$$

$$f_3^n = \min_{y_{i+2}} [c_2^n L(y_m(i+2) - y_{i+2}) + c_5 q_m(i+4)/y_{i+2}$$

$$+ c_7 d_{i+2}(y_{i+2}) + \lambda_{i+2} y_{i+2}]$$

where the coefficients, c_ℓ, are not functions of $u_{i,j}(k)$, $z_i(k)$.

We then solve

$$\min_n (f_1^n + f_2^n + f_3^n) \quad n=1, \ldots, N$$

to give v_i^N, $v_i'^N$, y_{i+1}^N, y_{i+2}^N .

2. Determine $u_{i,j}^{N+1}$ by

$$u_{i,j}^{N+1} = \begin{cases} 0 , & R \geq 0 \\ 1 , & R < 0 \end{cases}$$

where

$$R = (\mu_i' - \mu_i) x_{i,j} + a(v_i^N, y_{i+1}^N) c_{1,j} - a(v_i'^N, y_{i+2}^N) c_{2,j} .$$

If $u_{i,j}^{N+1} \notin \{u_{i,j}^n, n=1, \ldots, N\}$ go to step 1 otherwise, check optimality in step 3.

3. Define:

$$a_1(1) = a_1(3) = \min a(v_i, y_{i+1})$$

$$a_2(1) = a_2(2) = \min a(v_i', y_{i+2})$$

$$a_1(2) = a_1(4) = \max a(v_i, y_{i+1})$$

$$a_2(3) = a_2(4) = \max a(v_i', y_{i+2}) \quad .$$

Solve

$$\min_{u_{i,j}} \{\Sigma_j u_{i,j}(c_{3,j} + a_1(\ell)c_{1,j} - a_2(\ell)c_{2,j})$$

$$- (c_3^{N+1} + a_1(\ell)c_1^{N+1} - a_2(\ell)c_2^{N+1})\}$$

such that ℓ=1, 2, 3, 4 and

$$c_1^{N+1} = \Sigma_j u_{i,j}^{N+1} c_{1,j}$$

$$c_2^{N+1} = \Sigma_j u_{i,j}^{N+1} c_{2,j}$$

$$c_3^{N+1} = \Sigma_j u_{i,j}^{N+1} x_{i,j} \quad .$$

Let the solution be u^*. If $u^* \varepsilon \{u^n, n=1, \ldots, N+1\}$, stop. Otherwise, $u^{N+2} = u^*$ and return to step 1.

For diverge links where no subnetwork diverges, the above optimization becomes identical to the link optimization since the $u_{i,j}$ are held constant.

The subproblem that computes f_3^n is identical to the link subproblem for total density.

APPENDIX D

SOLUTION FOR AN ASSOCIATED PROBLEM

Consider the problem given by

$$\min_{u} \sum_{k=1}^{K} d^T(k)x(k)$$

such that $x(k+1) = [A(k) + \sum_i u_i(k)B_i(k)]x(k) + r(k)$

where $d(k)$, $x(k)$, $r(k)$ are n-dimensional vectors and $u_i(k)$ $i=1, \ldots, M$ are scalar control variables. We denote $u(k)$ as the M vector at time, k.

To solve the above problem we apply the Principle of Optimality [43]. Thus, at the K^{th} stage we have

$$V_1 = \min_{u(K-1)} d^T(K)x(K)$$

$$= \min [\sum_i (u_i(K-1)d^T(K-1)B_i(K-1)x(K-1))$$

$$+ d^T(K)(A(K-1)x(K-1) + r(K-1))] \quad .$$

Because u_i are routing control variables, they are multiplied only by x_i. Hence, the matrix B_i is zero except for the i^{th} column so that

$$\sum u_i(k-1)d^T(K)B_i(K-1)x(K-1) = \sum u_i(K-1)d^T(K)b_i(K-1)x_i(K-1)$$

where $b_i(K-1)$ is the i^{th} column of $B_i(K-1)$. It is now assumed that by

the flow constraints of a transportation network we have $x_i(K-1) \geq 0$ and the K^{th} stage solution is

$$u_i^*(K-1) = \begin{cases} 1 & d^T(K)b_i(K-1) < 0 \\ 0 & d^T(K)b_i(K-1) \geq 0 \end{cases} .$$

At the K-1 stage we apply the Principle of Optimality to give

$$V_2 = \min_{u(K-2)} [\Sigma u_i(K-2)(d^T(K-1)B_i(K-2) + \Sigma u_i^*(K-1)B_i(K-1)d^T(K)B_i(K-2) + d^T(K)A(K-1)B_i(K-2))x(K-2) + C]$$

where C is not a function of u(K-2). The coefficient of $u_i(K-2)$ can be written as

$$[d^T(K-1) + d^T(K)(A(K-1) + \Sigma u_i^*(K-1)B_i(K-1)]b_i(K-2)x_i(K-2)$$

and defining,

$$p^T(k) = - p^T(k+1)A^*(k) - d^T(k)$$

$$p^T(K+1) = 0$$

$$A^*(k) = A(k) + \Sigma u_i^*(k)B_i(k)$$

then the optimal control at K- 2 is

$$u_i^*(K-2) = \begin{cases} 1 & -p^T(K-1)b_i(K-2) < 0 \\ 0 & -p^T(K-1)b_i(K-2) \geq 0 \end{cases} .$$

Proceeding recursively to the k^{th} stage

$$V_{K-k+1} = \min_{u(k)} [-p^T(k+1)(A(k) + \Sigma u_i(k)B_i(k))x(k) + C] .$$

Thus, the optimal control at each time k is

$$u_i^*(k) = \begin{cases} 1 & -p^T(k+1)b_i(k) < 0 \\ 0 & -p^T(k+1)b_i(k) \geq 0 \end{cases} .$$

REFERENCES

[1] Lighthill, M.J. and Whitham, G.B., "On Kinematic Waves-II, A Theory of Traffic Flow on Long Crowded Roads", Proc. Roy. Soc. (London) 229A, pp 317-345, May 1955.

[2] Cunningham, E.P., "A Dynamic Model for Vehicle Routing in a Two-Way PRT Network", Trans. Res., Vol. 9 No. 6, 1975.

[3] Gazis, D.C., et.al., "Nonlinear Follow-the-Leader Model of Traffic Flow", Ops. Res. 9, pp. 545-567, 1961.

[4] Alberti, E. and Belli, G., "Contributions to the Boltzman-Like Approach for Traffic Flow - A Model for Concentration Dependent Driving Programs", Transpn. Res., Vol. 12 pp. 33-42, 1978.

[5] Macleod, C.J. and Al-Khalili, A.J., "Modelling of Urban Traffic Networks", Transpn. Res. Vol. 12, pp. 121-130, 1978.

[6] Garrard, W.L., et.al., "State of the Art Longitudinal Control of Automated Guideway Transit Vehicles", High Speed Ground Transp. J., Vol. 12, No. 2, 1978.

[7] Pue, A.J., "A State Constrained Approach to Vehicle-Follower Control for Short-Headway AGT Systems", ASME Journal of Dynamic Systems, Measurement, and Control, Dec. 1978.

[8] Gershwin, S., "A Design Tool and a Routing and Scheduling Technique for Personal Rapid Transit Systems", 1975 International Conference on Personal Rapid Transit, Denver, Col. June 1976.

[9] Tong, Y.M. and Morse, A.S., "Decentralized Control of Automated Transportation Systems via Cellular-Flow Coordination", 1976 Conference on Decision and Control.

[10] Kornhauser, A.L., and McEvaddy, P., "A Quantitative Analysis of Synchronous vs. Quasi-Synchronous Network Operations of Automated Transit Systems", Trans. Res., Vol 9, pp 241-248, 1975.

[11] Tomlin, J.A., "A Mathematical Programming Model for the Combined Distribution-Assignment of Traffic", Trans. Sci., Vol. 5 pp 122-130, 1971.

[12] Payne, H.J., Thompson, W.A., Isaksen, L., "Design of a Traffic-Responsive Control System for a Los Angeles Freeway", IEEE Transactions of Systems, Man, and Cybernetics, Vol. SMC-3, No. 3, May 1973.

[13] Kaya, A., "On the Optimization of Traffic Flow for Urban Transportation Systems", 1971 Joint Automatic Control Conference, St. Louis, Missouri.

[14] Merchant, D.K., Nemhauser, G.L., "A Model and an Algorithm for the Dynamic Traffic Assignment Problem", Trans. Sci., Vol. 12, No. 3, pp 183-199, Aug. 1978.

[15] Merchant, D.K., Nemhauser, G.L., "Optimality Conditions for a Dynamic Traffic Assignment Model", Trans. Sci., Vol. 12, No. 3 pp 200-207, Aug. 1978.

[16] Cantor, D.G., Gerla, M., "Optimal Routing in a Packet-Switched Computer Network", IEEE Trans. on Computers, Vol. C-23, No. 10, Oct. 1974.

[17] Gallager, R.G., "A Minimum Delay Routing Algorithm Using Distributed Computation", IEEE Trans. on Comm., Vol. COM-25, No. 1, Jan. 1977.

[18] Bersekas, D., Gafni, E., Vastola, K., "Validation of Algorithms for Optimal Routing of Flow in Networks", 1978 IEEE Conference on Decision and Control.

[19] Bertsekas, D.P., "Algorithms for Optimal Routing of Flow in Networks", Coordinated Science Laboratory Working Paper, Univ. of Illinois, June 1978.

[20] Meditch, J.S., Mandojana, J.C., "A Decentralized Algorithm for Optimal Routing in Data-Communication Networks", 1979 Conference on Decision and Control, Fort Lauderdale, Fla., Dec. 1979.

[21] Sandell, N.R., et.al., "Survey of Decentralized Control Methods for Large Scale Systems", IEEE Trans. Auto. Contr., AC-23, No. 2, April 1978.

[22] Geoffrion, A.M., "Elements of Large-Scale Mathematical Programming - Parts I and II", Management Sci., Vol. 16, No. 11, July 1970.

[23] Lasdon, L.S., "Duality and Decomposition in Mathematical Programming", IEEE Trans. on Sys. Sci. and Cyb., Vol. SSC-4, No. 2, July 1968.

[24] Pearson, J.D., "Dynamic Decomposition Techniques", from Optimization Methods for Large-Scale Systems, Wismer, Ed., McGraw-Hill, 1971.

[25] Roesler, W.J., Waddell, M.C., Ford, B.M., Davis, E.A., "Operating Strategies for Demand-Actuated ACGV Systems", Vol. I and II, APL/JHU TPR-019, Aug. 1971.

[26] Athans, M., "A Unified Approach to the Vehicle Merging Problem", Trans. Res., Vol. 3, No. 1, pp 123-164, Apr. 1969.

[27] Brown, S.J., "Traffic Merging Under Vehicle-Follower Control", 1975 International Conference on Personal Rapid Transit, Denver, Col., 1975.

[28] Whitney, D.E., "A Finite State Approach to Vehicle Merging", ASME Journal of Dynamic Systems, Measurement and Control", pp. 147-151, June 1972.

[29] Sakasita, M., "An Analysis of Merge Control for the Automated Transportation System", 1975 International Conference on Personal Rapid Transit, Denver, Col., 1975.

[30] Godfrey, M.B., "Merging in Automated Transportation Systems", Sc.D. Thesis, Department of Mechanical Engineering, M.I.T., June 1968.

[31] Caudill, R.J., Youngblood, J.M., "Intersection Merge Control in Automated Transportation Systems", Trans. Res., Vol. 10, pp 17-24, 1976.

[32] Brown, Jr., S.J., "Point-Follower Automatic Vehicle Control: A Generic Analysis", APL/JHU CP 057/TPR 025, May 1977.

[33] Liopiros, K.J., "PRT Station Operational Strategies and Capacities", 1973 International Conference on Personal Rapid Transit", Minneapolis, Minnesota, Feb. 1974.

[34] Sirbu, Jr., M.A., "Station Configuration, Network Operating Strategy and Station Performance", 1973 International Conference on Personal Rapid Transit, Minneapolis, Minnesota, Feb. 1974.

[35] Johnson, R.E., Walker, H.T., Wild, W.A., "Analysis and Simulation of Automated Vehicle Stations", 1975 International Conference on Personal Rapid Transit, Denver, Col., June 1976.

[36] Waddell, M.C., Williams, M.B., Ford, B.M., "Deposition of Empty Vehicles in a Personal Rapid Transportation System", APL/JHU TPR-28, May 1974.

[37] Tabak, D., "Application of Modern Control and Optimization Techniques to Transportation Systems", Control and Dynamic Systems, pp 346-423, Vol. 10, Academic Press.

[38] Kershner, D.L., "Network Analyses of Advanced Group Rapid Transit Systems", APL/JHU CP 070/TPR 042, April 1979.

[39] Yu, M. Ermol'ev, "Methods of Solution of Nonlinear Extremal Problems", Kibernetika, Vol. 2, No. 4, pp 1-17, 1966.

[40] Poljak, B.T., "A General Method of Solving Extremum Problems", Soviet Mathematics Doklady 8, pp 593-597, 1967.

[41] Held, M., Wolfe, P., Crowder, H., "Validation of Subgradient Optimization", Mathematical Programming 6, pp 62-68, 1974.

[42] Bazaara, M.S., Shetty, C.M., Nonlinear Programming, John Wiley & Sons, N.Y., 1979.

[43] Meditch, J.S., Stochastic Optimal Linear Estimation and Control, McGraw-Hill, N.Y., 1969.

[44] Ekeland, I., Teman, R., Convex Analysis and Variational Problems, North-Holland, American Elsevier.