Title
High Coverage Point to Point Transit (HCPPT): A New Design Concept and Simulation-Evaluation of Operational Schemes

Author
Cortés, Cristián Eduardo

Publication Date
2003

UNIVERSITY OF CALIFORNIA,
IRVINE

High Coverage Point to Point Transit (HCPPT): A New Design Concept and Simulation-Evaluation of Operational Schemes

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

In Civil Engineering

by

Cristián Eduardo Cortés

Dissertation Committee:
Professor R. Jayakrishnan, Chair
Professor Wilfred W. Recker
Professor Amelia C. Regan

2003

The Dissertation of Cristián Eduardo Cortés is approved and is acceptable in quality and form for publication on microfilm:

_____

_____

_____

Committee Chair

UNIVERSITY OF CALIFORNIA, IRVINE
2003

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank all the people who made this work possible. First, I would like to thank my advisor, Dr. R. Jayakrishnan, for his support and guidance. I also thank the other committee members, Dr. Will Recker and Dr. Amelia Regan for their comments and several fruitful discussions. The comments, guidance and advise of Dr. Kenneth Small are also greatly appreciated.

At this point, I would like to mention all my friends at ITS-UCI, particularly Riju Lavanya, Jun Oh-Seok and Laia Pagès, who helped me a lot with fruitful ideas and collaboration. In a very special way, I would like to thank my wife Verónica for her love, patient and encouragement.

This dissertation is part of a research project that was partially funded by PATH (Partners for Advanced Transit and Highways), Task Order 4111. I thank Mr. Lindsee Tanimoto from California Department of Transportation for encouragement in the project. Financial support was provided by a Scholarship from the National Scholarship Program of Chile, the Regents' Dissertation Fellowship from the University of California at Irvine and the Doctoral Dissertation Grant Award from the University of California Transportation Center.

CURRICULUM VITAE

Cristián Eduardo Cortés

EDUCATION

2003	Ph.D. in Civil Engineering, University of California, Irvine, U.S.A., Spring 2003.
1995	M.S. in Civil Engineering (Transportation, graduated *summa cum laude*), University of Chile, Santiago, Chile.
1995	B.S. in Civil Engineering (graduated *summa cum laude*), University of Chile, Santiago, Chile.

WORK EXPERIENCE

1998 to 2003	Graduate Student Researcher, Institute of Transportation Studies, University of California at Irvine.
1997 to present	Instructor Professor, Department of Civil Engineering, University of Chile.
1996-1997	Chief of Strategic Studies Area, Commission for Transport Infrastructure Investment Planning, Chile (SECTRA).
1994-1996	Transport Analyst, Commission for Transport Infrastructure Investment Planning, Chile (SECTRA).
1993-1997	Lecturer and Researcher, Department of Civil Engineering, University of Chile.
1990-1992	Transportation Consultant.

TEACHING EXPERIENCE

1996-2003	Instructor: Transportation Networks (CI-53J/73B). Design Seminar (SD 20A), (along with Professors Sergio Jara-Díaz and Antonio Gschwender). Transportation Economics (CI-43B), (along with Professor Sergio Jara-Díaz).

1989-1998	Teacher Assistant (TA) Transportation networks (CI-73B), Advanced Topics in Transportation Economics (CI-73G), Transportation Economics (CI-43B), Transportation System Analysis (CI-43A), Physics (FI-10A), Dynamic Systems (FI-21B), Mechanics I (FI-22A), Calculus (MA-22A), 1990. Linear Algebra (MA-122).

All the above courses have been offered by the School of Engineering at University of Chile.

HONORS AND AWARDS

Doctoral Dissertation Grant Award from University of California Transportation Center, 2002.

Regents' Dissertation Fellowship from University of California at Irvine, Fall 2001.

Scholarship from the National Scholarship Program of Chile (1998-2001).

PRESENTATIONS

Cortés, C.E. and R. Jayakrishnan (2003), High Coverage Point-to-Point Transit: A New Design Concept, *UCTC Annual Conference, UCLA, February 2003.*

Cortés, C.E. (2003), High Coverage Point-to-Point Transit (HCPPT): A New Design Concept and Simulation-Evaluation of Operation Schemes, *Doctoral Dissertation Session, 82th Transportation Research Board Annual Meeting, Washington D.C, January 2003.*

Cortés, C.E. and R. Jayakrishnan (2002), Real-time Routed Transit Systems-Routing Control Schemes, *INFORMS Annual Meeting, San Jose CA, November 2002.*

Cortés, C.E., R. Jayakrishnan, and J. Oh (2001), Analysis of a High Coverage Point-to-Point Transit (*HCPPT*), *INFORMS Annual Meeting, Miami Beach FL, November 2001.*

Jayakrishnan, R. and C.E. Cortés (2001), High Coverage Point-to-Point Transit (*HCPPT*): Simulation-Evaluation of Operational Schemes for Future Technological deployment, *PATH Program-wide meeting 2001, University of California Davis, October 2001.*

Jara-Díaz, S.R. and C.E. Cortés (1994), Calculation of Scale Economies from Transport Cost Functions. *Seventh World Conference on Transport Research, Sydney, 1994.*

Jara-Díaz, S.R. and C.E. Cortés (1993), Economías de Escala en Funciones Pseudo-vectoriales en Transporte. *Congreso Panamericano de Ingeniería de Tránsito y Transporte, Mexico, 1993.*

Jara-Díaz, S.R. and C.E. Cortés (1993), Análisis Crítico del Uso de Funciones Pseudo-Vectoriales en Transporte. *Cuarto Seminario de Investigación en Transporte, Universidad de Chile (SITU), September 1993.*

PUBLICATIONS

Oh, Jun-Seok, C.E. Cortés and Will Recker (2003), Effects of Less-Equilibrated Data on Travel Choice Model Estimation, forthcoming publication in *Transportation Research Record*.

Jayakrishnan, R., C.E. Cortés, R. Lavanya and Laia Pagès (2003), Simulation of Urban Transportation Networks with Multiple Vehicle Classes and Services: Classifications, Functional Requirements and General-Purpose Modeling Schemes. *Proceedings of the 82th Transportation Research Board Annual Meeting, Washington D.C, January 2003.*

Cortés, C.E., R. Lavanya, Jun-Seok Oh and R. Jayakrishnan (2002), A General Purpose Methodology for Link Travel Time Estimation Using Multiple Point Detection of Traffic, *Transportation Research Record*, vol. 1802, pp. 181-189.

Jara-Díaz, S.R., E. Martínez-Budría, C.E. Cortés and L. Basso (2002), A Multioutput Cost Function for the Services of Spanish Ports' Infrastructure, *Transportation*, vol. 29, pp. 419-437.

Cortés, C.E. and R. Jayakrishnan (2002), Design and Operational Concepts of a High Coverage Point-to-Point Transit System, *Transportation Research Record*, vol. 1783, pp. 178-187.

Jara-Díaz, S.R., C.E. Cortés and F. Ponce (2001), Number of Points Served and Economies of Spatial Scope in Transportation Cost Functions, *Journal of Transport Economics and Policy*, vol. 35, N^{o}2, pp. 327-341.

Oh, Jun-Seok, C.E. Cortés, R. Jayakrishnan and Der-Horn Lee (2000), Microscopic Simulation with Large-Network Path Dynamics for Advanced Traffic Management and Information Systems, *Proceedings of the 6th International Conference on Applications of Advanced Technologies in Transportation Engineering, Singapore, June 2000.*

Oh, Jun-Seok and C.E. Cortés (1999), A Microscopic Simulation Approach to the Evaluation of Transportation Improvement Projects: Evaluation of High-Priority Choke-Point Improvement Projects, *Institute of Transportation Studies Working Paper 99-17, University of California at Irvine.*

Jara-Díaz, S.R., E. Martínez, C.E. Cortés and A. Vargas (1997), Marginal Costs and Scale Economies in Spanish Ports: a multiproduct approach, *Proceedings of the 25th European Transport Forum, Brunel University, England, September 1997*, pp. 137-147.

Jara-Díaz, S.R., C.E. Cortés and F. Ponce (1997), Número de Puntos Servidos y Economías de Diversidad Espacial en Funciones de Costo en Transporte Aéreo, *Actas del VIII Congreso Chileno de Ingeniería de Transporte, Santiago, Chile. November 1997*, pp. 133-145.

Jara-Díaz, S.R. and C.E. Cortés (1996), On the Calculation of Scale Economies from Transport Cost Functions, *Journal of Transport Economics and Policy*, vol. 30, pp 157-170.

Jara-Díaz, S.R. and C.E. Cortés (1995), Descripción del Producto y Cálculo del Grado de Economías de Escala en Transporte, *Actas del VII Congreso Chileno de Ingeniería de Transporte, Santiago, Chile. October 1995*, pp. 15-29.

Cortés, C.E. (1995), Análisis Crítico del Uso de Funciones de Costo Pseudo-Vectoriales en Transporte, *M.S Thesis, Civil Engineering Department, University of Chile.*

Irvine, May 2003

ABSTRACT OF THE DISSERTATION

High Coverage Point to Point Transit (HCPPT): A New Design Concept and Simulation-Evaluation of Operational Schemes

by

Cristián Eduardo Cortés

Doctor of Philosophy in Civil Engineering

University of California, Irvine, 2003

Dr. R. Jayakrishnan, Chair

This dissertation research proposes the development and evaluation of a new concept for high-coverage point-to-point transit systems (*HCPPT)*. Overall, three major contributions can be identified as the core of this research: the proposed scheme design, the development of sophisticated routing rules that can be updated in real-time to implement and optimize the operation of such a design, and the implementation of a multi-purpose simulation platform in order to simulate and evaluate such a design under real network conditions.

The design is based on Shuttle-style operations with a large number of deployed vehicles under a coordinated transit system that uses advanced information supply schemes with fast routing and optimization schemes. The system design is rather innovative and ensures that no more than one transfer is needed for the travelers, by using transfer hubs as well as reroutable and non-reroutable portions in the vehicles' travel plans. It yields flexibility for demand-side benefits from options such as price

incentives for time-bound "passenger-pooling" at the stops without destination constraints, by the users.

A strict optimization formulation and solution for such a problem is computationally prohibitive in real-time. The design proposed in this dissertation is effectively geared towards a decomposed solution using detailed rules for achieving vehicle selection and route planning. If real-time update of probabilities based upon modeling the future dispatch decisions is included, then this scheme can be considered as a form of quasi-optimal predictive-adaptive control problem.

Finally, a multi-purpose simulation platform is developed as part of this research in order to evaluate the performance of the system. The final simulations of *HCPPT* required point-to-point vehicle simulation, which is not possible with off-the-shelf simulators. The simulation framework uses a well-known microscopic traffic simulator that was significantly modified for demand-responsive vehicle movements and passenger tracking. A simulated case study in Orange County showed that with enough deployed vehicles, the system can be substantially better, even competitive with personal auto travel, compared to the often-unsuccessful traditional *DRT* systems and the existing fixed route public transit. Furthermore, *HCPPT* can be incrementally implemented by contracting out services to existing private operators.

1 INTRODUCTION

1.1 Motivation and problem definition

Transit systems planning, design and operation in the United States have often been based on well-set notions of what portion of the traveling population require or would ever use transit, and on traditional paradigms of mode split behavior. Rather than provide service that can attract demand away from personal automobiles to transit, the thinking has been rather defeatist, considering transit as a necessary alternative to serve a small but more captive part of the traveling public, but not necessarily one that could yield good mobility and ridership benefits. With urban sprawl and reduced population densities being a reality in many large urban areas, Southern California being a prime example, the conventional notions have resulted in transit systems that have been roundly criticized for the high costs and poor benefits. The research in this dissertation takes a fresh look at this and develops some newer ideas for transit systems that could serve point to point travel desires in general urban networks. The intent is to develop public transit designs for networks without sufficient high-density of population and activities and without the kind of corridor structure that make traditional types of transit effective. In the process the research develops schemes with useful features that could also help in augmenting the service in urban contexts where traditional transit has been relatively successful.

To motivate the research here, consider the ongoing planning for the light rail transit project in Southern California.. Consider the costs involved in what may be termed “conventional” transit. Among the alternatives under consideration in the

particular project is a light-rail system that will involve a 30 mile line at about $2 billion capital cost (about $60 million per mile) for a projected 2020 ridership of about 75,000 trips. Now look at a different scenario. If we were to consider mini-vans in that corridor which is roughly along a freeway, the interest on that capital (say $200 Million) is enough to operate 2000 minivans (at an annualized cost of $15,000 for purchase/ replacement) and pay 3000 workers (at say $55,000/yr) every year. Two thousand vans making 10 round trips (90 minutes) and handling even 5 unlinked trips each way yields 200,000 trips a day – All of that just from the interest on the capital spent on the "traditional" fixed line rail transit! Of course, clearly much more than five passenger trips per 30 miles of vehicle trip could materialize if vehicles are available essentially all the time for travel, and if they actually reach a wider area around the corridor, in the larger network itself.

Indeed, very conservative numbers were used for the argument above, but the benefits are indisputably in favor of multiple-passenger vehicle travel on existing roadways,the arguments about hidden subsidization of road travel notwithstanding. Many experts in transportation generally know and accept similar thoughts as were just made, but have had difficulty in countering the argument from the advocates of traditional transit systems – namely, "do you have anything better? Transit is for only certain demographic segments; we have to provide it. Nobody else may use transit; and so we have to accept that we will be spending big money for it". Also, those who feel that bus transit is a much more cost-effective alternative than rail and would like to see more bus travel, find it difficult to counter the argument that "buses aren't sexy," as notoriously stated by a prominent Los Angeles city politician. The argument is that

"none who can afford to drive a car will be on a bus – perhaps on a train, but never on a bus". These arguments go round and round in vicious cycles. It is true that buses themselves may not be the solution to the problem, though it is a much cheaper alternative in many contexts than rail transit. Perhaps the answer lies in using smaller "buses" with much more service flexibility. Perhaps the answer lies in taxi or mini-van type services. If the question is whether we can ever operate road transit vehicles in a manner where sufficient travel demands can potentially occur, the answer is yes. It can be done; however, it requires quite a different approach towards transit.

Let us consider what would be the most generic public transportation design under the most unrestricted demand generation and distribution pattern. That is, consider uncertain demand where customers would ask for service in real-time at random points on the network. The solution of such a general problem naturally should result in flexible vehicle routes and schedules adjusted in real-time as the demand calls come in. In the past, primarily due to lack of information regarding demand and supply in real time, most of the high-demand transit systems have been restrictively designed to comprise "fixed routes and schedules". Computationally poor real-time algorithms and routing capabilities are also factors that have made the "fixed route-schedule" design the only viable option for high-demand travel necessities.

An important and well-known problem with fixed route transit is the lack of accessibility for wider population and to sufficient destinations. Park-and-ride options exist in some cases, but in most examples, the biggest impediment in attracting auto drivers to transit has been the fact that "bus and rail go to every other place one doesn't want to go to, and never comes anywhere near for one to board them". If it takes 15

minutes of walking and 15 minutes of waiting for a bus line with 30 minute frequency for a 15 minute travel to a destination 5 miles away, which has to be followed by 15 minutes more of walking, why would anybody who can afford driving and parking fees not drive for 10 minutes? Any amount of transit mode choice research will not throw any new insights into solving this problem, but this is the reality in many dispersed urban areas of the U.S. Alternatives that allow travel essentially from any origin to any destination with minimal additional access time and minimal waiting time need to be provided if travelers are to be attracted to public transit.

The thinking that transit primarily needs "fixed routes and schedules" is itself perhaps no longer valid if computer and communication technology and efficient algorithms offer ways for optimized travel in public (or privately operated) transit vehicles. Fixed route services in transit without near-the-door pickup and drop is never attractive other than in some very high-density corridors of the type only rarely found in Southern California and in many cities that have undergone suburbanization. After all, from a theoretical standpoint, fixed routes and schedules are nothing but a subset of truly optimal flexible and dynamic solutions, if we were able to find such optima. One could even extend an argument that if advanced information technology and optimization capabilities existed before vehicles, we would not be seeing any fixed route, fixed schedule travel – perhaps a stretch, but an interesting thought nonetheless.

Therefore, if the objective were to offer a transit alternative competitive with the automobile, a high-coverage shuttle/van type of system would be the only alternative. That is, to offer a public transit option without the extreme waiting times and travel discomfort characteristics of traditional "fixed-route and schedules" scheme, not to

mention cost, that makes private personal automobile travel so much more attractive. This was the hope behind the demand responsive transit systems developed from over three decades back. Before proceeding with further exposition of the ideas in this research, let us consider such systems from the past and what went wrong with them, to use a rather impetuous line as a conclusion.

The traditional dial-a-ride demand responsive transit systems (henceforth *DRT*) have been classified as a many-to-many type of service and included capacity constraints as well as soft time window constraints at the pick-up and delivery locations. *DRT* systems consist of one or more multiple occupant vehicles, which take passengers from their origins to their destinations within a service area. Though demand-responsive transit has been in existence in several cities around the US (Davenport, IA, has had one from 1934!), serious research into larger-scale demand-responsive transit did not start till the 1960s. Many demonstration projects (Peoria, IL, 1964; Flint, MI, 1968; Mansfield, OH, 1970) were only marginally successful at best. The most intensive academic research into DRT was at MIT starting in 1970, in the well-known project CARS directed by Prof. Nigel Wilson (see chapter 2 for more details). The work resulted in heuristic algorithms and a demonstration project by MIT at Rochester (Wilson and Colvin, 1977) and another demonstration project by MITRE Corporation at Haddonfield, NJ. The generally accepted conclusion was that, perhaps due to the modest computational capabilities available then, manual dispatching performed better than computer dispatching (Black, 1995). Although it spurred a lot of research over several years about *DRT* systems, neither a successful methodology nor any transit simulation results have been provided for solving large-scale *DRT* problems. That is, *DRT* systems

were never considered as an alternative to serve large areas with a large set of vehicles. This dissertation starts with the premise that it could well be applicable in a larger scale and is perhaps more applicable for the larger contexts than otherwise.

The primary problem faced by existing implementations of *DRT* systems is the low occupancies found in the buses/vans and the resulting cost of operation. The operating costs per passenger is rarely under $10.00 and is often around $15.00, the cause for that being the unacceptably low passenger loads (occupancy) found in the vehicles, ranging from 0.51 to 2.74, with an average of 1.39 (Black, 1995). The studies show that the request-to-arrival-time of Dial-a-ride in comparison with automobile door-to-door-travel-time is often in the range of 2 To 3 (i.e., it takes 30 to 45 minutes for a vehicle to arrive for service, for a trip that can be made in 15 minutes by car!). This means that demand-responsive systems tried in the past are not competitive with auto travel and thus will have low ridership, with normally only those without an available auto-option using it.

Demand responsive systems are not really responsive to real-time travel requests due to the poor coverage of the network by the vehicles (too few vehicles running). Moreover, there is an embedded system design problem along with inappropriate routing schemes and demand considerations. As an example, despite transfers causing serious extra costs in travel, allowing it with proper designs could significantly improve the overall flexibility and thus the travel costs for many passengers. However, transfers are not part of the traditional *DRT* designs already burdened with the lack of large-enough fleets of vehicles. The result was the poor level of service due to low "coverage". This is in part why efforts in *DRT* designs have been devoted to solving small problems,

mostly oriented toward the service of small communities or passengers with specific requirements (elderly, disabled). If a regular customer required an automobile kind of service, he (she) would rather call for a taxicab or drive a personal vehicle rather than use the public *DRT* system. Improving the level of service, reducing waiting and access times and thus in turn improving the demand are the reasons for the "high-coverage" notion in this dissertation research, however impractical it may seem at first.

The point is clear that if demand responsive transit is to ever become cost-effective, the occupancies in the vehicles should go up, at least to 3 or 4 passengers on average for van-type service, and 15-20 passengers for bus-type service. This will not happen unless the level of supply of vehicles is significantly higher, and the time between call and service is significantly lower to make even those who think of transit as an option to select it over personal automobiles. If the service needs to be quick in its response, it is imperative that vehicles are available within say 2 or 3 miles and can be rerouted in such a way that the passengers can at least reach a transfer node for further travel. The home waiting time should be no more than 10 minutes or so, preferably less. The design should also ensure that no more than one transfer is made by passengers traveling up to 10 or 15 miles, since this is an aspect that significantly affects the demand, and sometimes overlooked in transit network design. We claim that a key aspect here is vehicle availability that reduces waiting times, and real-time information schemes that reduce uncertainty.

Now, let us consider the real-time optimization problem involved here. Let us assume that we are able to solve the large-scale pick-up-and delivery problem in real time with uncertain point-to-point travel demand generation in real-time. In many cases,

we would find solutions such as those shown in figure 1.1. The solution would show different vehicles picking up and delivering passengers traveling on their own vehicle tours that minimize some type of a generalized cost including the total travel times of the passengers. While in a static problem similar to a multiple-vehicle traveling salesman problem we would not find the tours crossing each other, in the dynamic problem (if were to solve it) would yield tours that would cross each other in space though not in space-time. Note that if we were also able to solve for a true optimum that included the best multiple-vehicle options for travelers as well, we would have transfers between vehicles (again with costs for transfer properly accounted for in the formulation). An efficient transfer in some cases could also be by dropping a passenger at one point for a later pickup by another vehicle (if unnecessary travel by the passenger is of worse cost than waiting at a point for the transfer). Such a solution would require the finding of transfer points in continuous space-time that of course brings an added dimension making the already unsolvable large problem effectively unthinkable as well. Figure 1.1 is just to aid the thinking and is not representative of the actual solutions of such large dynamic pick-up and delivery problems. It does show one important aspect, which is that vehicle tours would tend to gravitate to certain high-capacity road corridors such as a freeway or a high capacity arterial street.

The optimal solution will send these vehicles on to optimal pick up and delivery routes in local streets with travel on higher-speed arterial/freeway (trunk) routes whenever possible. This is the reason for the designs of feeder *DRT* and trunk express transit (*BRT* or high-speed rail, for instance) in the past.

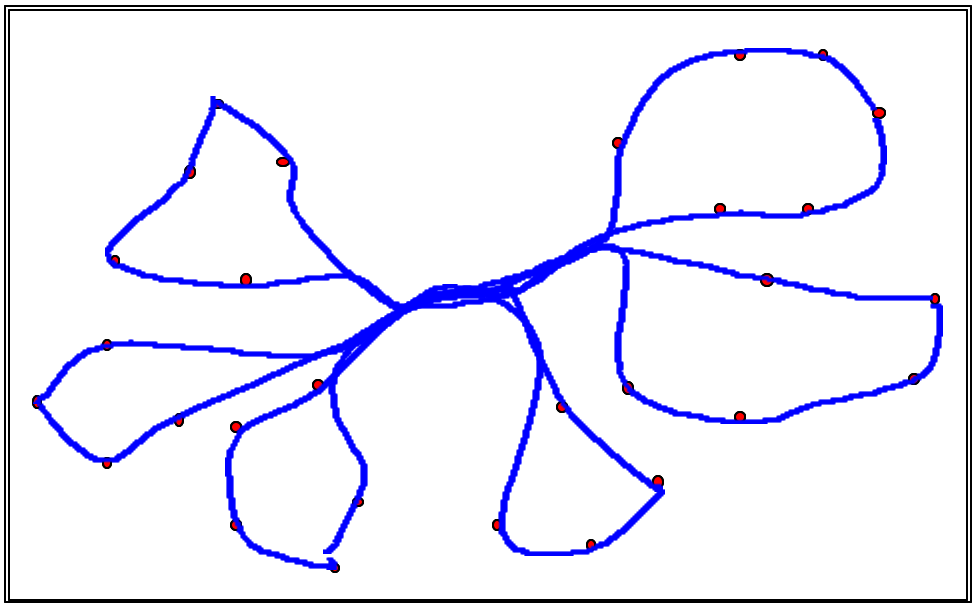

Figure 1.1 The type of solution in a stochastic pick-up and delivery problem

If we accept that we cannot optimally solve the above problem for real-time operations for any better than a handful of points and vehicles, then an option is to intuitively "expect" the characteristics of the optimum. If we fix such characteristics in the system design, it would yield the problem solvable, though only to a near-optimal solution. In the above case, fixing the transfer points is perhaps the first such option to be considered. The conceptual reason for hubs and timed-transfer terminals proposed in the past literature is no different, even though real-time optimizations were not envisaged. It is logical to expect fixed transfer points in the type of real-time pick-up and delivery design this dissertation focuses on.

However, for point-to-point travel, such a design would appear to need at least TWO transfers as shown in figure 1.2. Transfer points add flexibility to the system operation as well as the use of vehicles, especially along express corridors where multiple vehicles could share a roadway with better level of service (such as exclusive

lanes or entire roads for transit use). In addition, vehicle productivity should be always enhanced under such a scheme due to the possibility of vehicles to feed (and get feeds from) other vehicles at transfer points, thus increasing the average passenger occupancy.

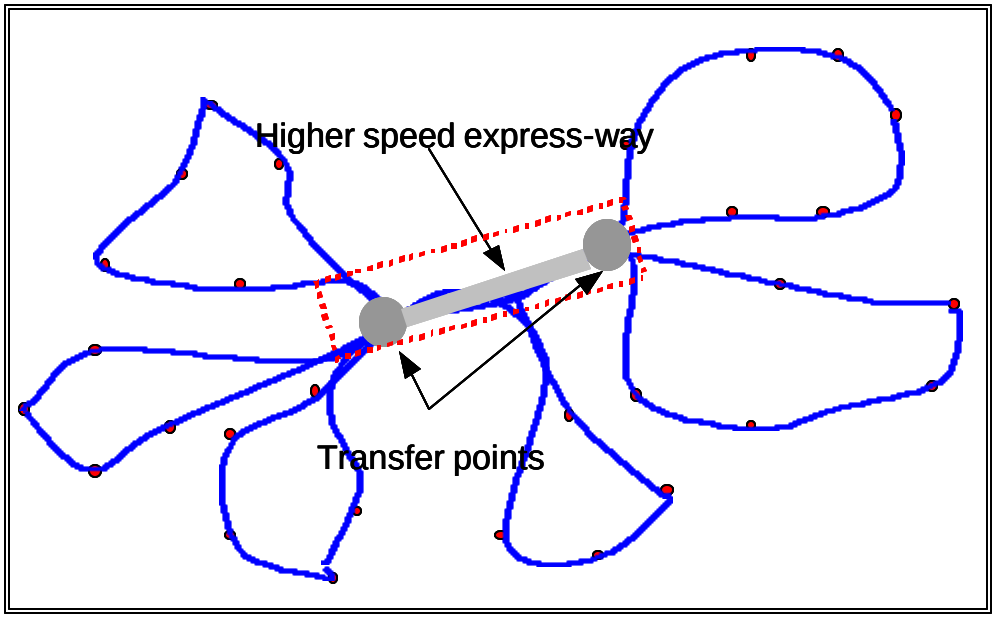

Figure 1.2 Constrained design for solving of a real-time pick-up and delivery problem

The designs proposed in this dissertation result from an attempt to develop a scheme where such travel can be accomplished with at most ONE transfer, at least between points not more than 10 or 15 miles apart. The routing schemes also ensure higher vehicle occupancy (number of riders) and can be considered similar to solutions obtained if a multiple objective optimization of the large optimization problem, maximizing ridership and minimizing travel/wait times could ever be attempted in real-time. The proposed designs allow real-time quasi-optimization using route improvement techniques where cost functions updated in real-time are utilized to solve local insertion heuristics along with traveling salesman problems for selection and routing of vehicles

to passenger pickup locations. The small capacity of the vehicles allow easy solutions of such routes with only 5 or 6 points at most in the candidate vehicle tours at any time.

The conceptual design is for door-to-door travel, allowing at most one transfer and based on real-time personalized travel desires. The implementation of such a system is now possible due to advances in communication and computing technologies. While it is demand-responsive, the concept is significantly different from older demand-responsive transit systems, which were often failures. The proposed system requires high coverage, referring to the availability of a large number of transit vehicles (often minibuses or vans), which could also operate in conjunction with private and paratransit systems. As mentioned above, the design strictly eliminates more than one transfer for any passenger.

An important claim is that the system could potentially provide a transit alternative that is much more competitive with personal auto travel than conventional transit systems, due to significantly lower travel and waiting times. The passenger demand for such a system is uncertain, but the simulations experiments performed in the context of this dissertation show that under a variety of acceptable demand levels, the system can operate with high cost-effectiveness.

As described in the later chapters, the technology for the kind of system proposed here has already been demonstrated in recent field trials of paratransit operations with online real-time customer calls vehicle to vehicle communication and broadcasting. The cost aspects of the system are also briefly considered in this research. The primary argument is relatively simple. If taxi systems can operate with an average of say $15 per trip with an average passenger occupancy of slightly over one per vehicle, it should be

possible to operate a system with 2.5 or 3 times as much occupancy for much cheaper fares. With much more of productive times (taxis are idle for a large fraction of time), a proper design can significantly reduce the operator cost per trip. If even half of the current transit subsidies (2 to 3 times the fare paid by the customer) is available for a high-coverage point to point system, it is not at all unreasonable that average trips can cost as little as a few dollars. An average customer cost of even $5 for a trip of 10 miles is not impossible and compares not too unfavorably with personal autos costing about $3 in fuel, vehicle maintenance and parking. The system could have the level of service comparable to taxis, perhaps much better due to reduced waiting times. These numbers are highly context specific and need to be validated first. This is the intent behind this research that develops the efficient designs and algorithms for operation and then simulates the operation in a large system to show that the above claims are reasonable at least in terms of the level of service and occupancy.

Basic Theoretical Outline:

The system essentially attempts to solve a stochastic real-time passenger pick-up and delivery problem with large number of vehicles. A strict optimization formulation and solution for such a problem is computationally prohibitive in real-time. The design proposed here effectively allows the problem to be decomposed and solved using detailed rules that achieve vehicle selection and route planning in a manner that is similar to adaptive predictive control using probabilities and expected values updated in real-time.

1.2 Research direction and methodology

This dissertation is the first step in a long-term effort that would culminate in field operational tests in the future with possible cooperation from city and transit agencies and/or private fleet operators. This will be after the operational paradigms are well designed and ready to be put into practice using advanced technological capabilities. Detailed simulations have shown the kind of social and economic factors to be considered in policy decisions. It also looks at this research as part of a bigger effort to develop various aspects such as cost functions and demand models, most of which are at present inadequate for demand responsive transit analysis.

The focus of this dissertation is in describing the details of the concept, providing arguments in favor of the system, developing routing schemes and simulating different demand-supply scenarios to show the benefits of the proposed idea. This dissertation does not purport to consider every issue in detail, but does look at the key design aspects initially and identifies myriad other topics that require future research.

The methodology is based on simulations using the UCI Testbed capabilities for various designs, supply levels and demand levels on a candidate network. There are three primary areas where the dissertation work is focused on:

1) Proposition of a conceptual scheme.
2) Development of real-time routing-scheduling algorithms.
3) Development of a general-purpose transit simulation platform.

The general tasks developed in this dissertation are the following:

- Development of a spatial network simulation framework, using discrete-event simulation techniques. The characteristics of this scheme as well as the simulation results are presented in Chapter 3. This first experiment is performed in order to show feasibility on the supply and performance side. The parametric simulation study results differ based on the number of vehicles deployed, demand level, system design, control decision design, and so on. More vehicles reduce travel and weighting times, but may also reduce vehicle occupancy and productivity unless demand is sufficient.

- Development and refinement of candidate operation designs and efficient real-time routing-scheduling algorithms. The routing decisions must be made in real-time; therefore some variables (associated with travel time measures) are assumed to be random with unknown distributions. The design proposed here effectively allows the problem to be decomposed and solved using detailed rules that achieve vehicle selection and route planning in a manner that is similar to adaptive-predictive stochastic control using probabilities and expected values updated in real-time.

- Development of simulation software using the online ATMIS Research Testbed capabilities for various designs, supply levels and demand levels on a candidate network, specifically a part of the Orange County Network in California. Essentially, an API integration for the PARAMICS simulator is developed in order to evaluate

the system under real-time conditions under real-network conditions coded in PARAMICS.

With regard to data sources, on the demand side, the demand models under the new OCTAM regional model for Orange County have been used in the final simulation experiments. In addition, a representation of a real network of Orange County was used to simulate the routing schemes and algorithms finally selected, through the aforementioned API interface integration for the PARAMICS simulator, focusing on transit modeling schemes. The interface along with the API simulator allows the modeler to evaluate the proposed schemes under real network scenarios and realistic traffic conditions.

This dissertation has developed significant guidelines on the applicability of the scheme along with the development of software for applications. A set of tools for evaluating in detail the future implementation of new designs for large-scale point-to-point transit systems is one of the core products of this research effort.

As further research, a parametric analysis of the impact on demand models and the impact of the system implementation on the modal split is addressed. The economic focus should contemplate the analysis of the cost structure and pricing associated to the proposed system under various schemes.

1.3 Outline of the Dissertation

In Chapter 2 a detailed literature review regarding *DRT* and combined transit systems is presented, emphasizing the state of the art in solving the dynamic case. The description and solution algorithms used in the past for such systems are addressed. In addition, a description of the pick-up and delivery problem with time windows, its formulation and solution algorithms are shown as a basic transportation problem reference.

Chapter 3 summarizes the basic proposed scheme as mentioned in the previous two sections, focusing on the potential of such a scheme, its characteristics, and the different options and travel strategies compatible with this conceptual idea. In addition, a numerical feasibility study is developed, based on a spatial discrete-event simulation methodology.

Chapter 4 presents the routing and scheduling rules developed in order to implement the system proposed in Chapter 3. Additional heuristics for terminal passenger management, vehicle distribution and real-time operational schemes are developed and addressed in this chapter.

In Chapter 5 a real-time adaptive predictive control scheme is developed in order to model stochastic delays for dynamic real-time vehicle routing problems in pick-up and delivery. These delays come from future unexpected vehicle rerouting and have to be incorporated into the cost function in the routing and scheduling rules of Chapter 4. A manner of updating these rules using system information is also addressed and analytically formulated.

Chapter 6 summarized the final simulation approach, the details in the development of the PARAMICS API and the requirements for modeling the system proposed in this dissertation. In Chapter 7, the simulation results are shown and analyzed, for various strategies, demand and supply levels.

In Chapter 8, the general and specific conclusions are delineated. The chapter includes also a section with recommendations and further investigation topics.

2 GENERAL BACKGROUND: DESCRIPTION OF TRANSIT SYSTEMS AND SOLUTION ALGORITHMS

2.1 Description and features of flexible transit systems

The objective of this chapter is to provide a description of the state of the art literature in the development, design, technological requirements and routing schemes of flexible transit systems studied and implemented nowadays. The background includes various topics, from algorithm development issues to network design considerations.

In order to introduce the problem, and also as a way to understand the inherent features of flexible transit systems, in this first section a classification of the existing classes of vehicles and services is presented. Perhaps the most logical way to distinguish between several kinds of transit systems is by their functionality. It is possible to enumerate three different systems: transit, paratransit and fleet systems. The first two are definitely more important in view of the objectives of this dissertation research.

1) Transit: Such services are normally composed of fixed route systems with routes and schedules planned in advance. Trains, light rail vehicles, metros, buses normally continue traveling in a circuit or back-and-forth route to pick up or drop off passengers at the stop locations. The service characteristics of interest in simulation include items such as whether these services share the right of way among each other as well as personal autos.

2) Paratransit: Taxis, limousines, shuttles, and demand-responsive systems are examples of this. This class of services can be further subdivided according to their routing or/and scheduling characteristics. Using this feature, we have the following services:

- Dial-a-ride: According to Black (1995), it is a service, which is more flexible than conventional transit services. It can be demand-responsive in the routing schemes (vehicles go exactly where the passenger wants - door to door service) or/and the scheduling (vehicles arrive when the passenger wants). Taxis would be effectively a special case of such services, where the passengers do not share rides.
- Jitney: According to Black (1995), jitney is a cross between taxi service and regular bus route. Jitneys operate along a fixed route, and they may have fixed stops. But there is no fixed schedule and passengers share the vehicle.
- Ride Sharing: In this case travelers form groups to share vehicles that operate when and where they want. From a simulation standpoint these would be similar to personal autos, but with potentially more stops for passengers to join the ride.
- Vehicle Sharing: These are systems where the rides are not shared, but different passengers or passenger groups share the same vehicle for multiple rides. The primary difference from a dial-a-ride or taxi service would be that there are no drivers for these vehicles and as such the vehicles are not "dispatched" (though there may be methods to control the passenger selection of vehicle in locations such as transit centers, as in the case of the proposed station-car schemes).
- Mixed systems: These are systems which may be considered similar to regular transit systems rather than paratransit, in that the vehicles may have fixed routes and

transfer points in certain parts of the network but with non-fixed, demand-responsive or real-time routed portions in other parts of the network.

3) Fleet services: Commercial services and emergency vehicles form the largest subdivision in this class. Regarding commercial vehicles, it is possible to see this system as a pick up and delivery problem with prior planning. At the beginning of the day or season drivers know where their demand is located for that period. Finally, emergency fleets include ambulances, fire and paramedical vehicles, and police vehicles. These may often be real-time routed services with a fixed drop-off location.

In the next section, a general review of the pick-up and delivery problem with time windows is presented, mainly based on the reviews by Desrochers *et al*. (1988) and Desrosiers *et al*. (1995). This problem along with the traditional vehicle routing problem represents an important conceptual benchmark for the proposed system in the context of this dissertation. In Subsection 2.3, a discussion about demand responsive systems and algorithms is conducted. Next, in Section 2.4, an overview about mixed services as defined above, is presented. Finally, in Section 2.5 some final remarks are outlined.

2.2 The general pick-up and delivery problem: considerations, restrictions and heuristics algorithms to solve the problem under specific conditions

In this section, the general pick-up and delivery problem with time windows is mathematically formulated. Notice that this problem represents the more general

formulation for the static case, and it represents a good starting point in order to understand the problem behind any flexible door-to-door transit design and optimization procedures.

According to Desrosiers *et al*. (1995), the pick-up and delivery problem with time windows (PDPTW) involves the satisfaction of a set of transportation requests by a heterogeneous vehicle fleet housed at several depots. A transportation request consists of picking up a certain number of customers at a predetermined pick-up location during a departure time interval and taking them to a predetermined delivery location to be reached within an arrival time interval. The departure and arrival time windows are based on desired pick-up or delivery time requests satisfied by the customers. Loading and unloading times are incurred at each vehicle stop.

For the transportation of people, Desrosiers *et al*. (1995) identifies three problem objectives to be minimized (in a hierarchical fashion) subject to a variety of constraints. The first objective involves the minimization of the number of vehicles or the total vehicle fixed costs required to satisfy the transportation requests. For this minimum value, a second objective considers the minimization of the total distance or travel time. In the case of passengers (unlike the problem of transportation of goods), an objective is introduced in order to minimize the inconvenience created by pick-ups or deliveries performed either sooner or later than desired by the customers. The latter PDPTW context is called dial-a-ride or demand responsive transit problem.

The PDPTW involves several constraints, which can be classified into visiting constraints (each pick-up and delivery has to be visited exactly once), time window constraints to be satisfied at each stop, vehicle capacity constraints, depot constraints,

coupling constraints (stating that for a given request the same vehicle must visit the pick-up and delivery stops), precedence constraints and resource constraints on the availability of drivers and vehicles.

Some of the previous constraints could be relaxed in a more complex scheme such as a mixed service where the customer could be delivered by a different vehicle from the one that picked him(her) up, changing the structure of the coupling constraints as defined above. Time window constraints could enter as soft constraint into the cost function imposing a penalty on pick-up or delivery times. As discussed in Section 2.3.3.2 for the dynamic case, in most of the dynamic cases a hard time window constraints does not make too much sense, and in some cases it could even make the problem unfeasible.

Let us formulate separately the one-vehicle case from the multiple vehicle pick-up and delivery problem. In Section 2.3, an exhaustive review of algorithms and heuristics to solve such a problem for the transportation of passengers (dial-a-ride or *DRT* problem) is presented.

The single vehicle pick-up and delivery problem

The original notation by Desrosiers *et al.* (1995) is utilized here. Consider a set of n requests. Associated to the pick-up location of request i a node i, and to his delivery location a node $n+i$. Let $N^P = \{1,\ldots,n\}$ be the set of pick-up nodes and $N^D = \{n+1,\ldots,2n\}$ be the set of delivery nodes, and define $N = N^P \cup N^D$. The set of nodes of the network is $V = N \cup \{o,d\}$, where o is the associated with the origin-depot location while d is associated with the destination-depot location.

Request i demands that p_i customers be transported from node $i \in N^P$ to the corresponding node $n+i \in N^D$. Let $l_i = p_i$, for $i \in N^P$, and $l_{n+i} = -p_i$ for $n+i \in N^D$. The capacity of a single vehicle is given by *Q*. For each pair of distinct nodes $i, j \in V$, t_{ij} represents the travel time, which includes the service time at node i, and c_{ij} denotes the travel cost. Let $[a_i, b_i]$ denotes the time window associated to node $i, i \in N$. Retain in the set A only the arcs $(i, j), i, j \in V$, which satisfy a priori capacity and time constraints, and other restrictions such as those based on precedence, and the same location.

Three types of variables arte used in the formulation: binary flow variables $X_{ij}, (i, j) \in A$, time variables $T_i, i \in V$, and load variables $L_i, i \in V$. The flow variable $X_{ij}, (i, j) \in A$ equals 1, if the vehicle travels form node i to node j, and 0, otherwise. The time variable $T_i, i \in V$, represents the time at which service begins at node i. The load variable $L_i, i \in V$, gives the total load on the vehicle just after the service is completed at node i. It is assumed that the vehicle departs empty from the depot at a given fixed time, *i.e.*, $L_0 = 0$ and $a_0 = b_0 = T_0$.

The nonlinear mathematical formulation is as follows

$$\textit{Minimize} \sum_{(i,j) \in A} c(L_i) c_{ij} X_{ij} \tag{1.1}$$

subject to

$$\sum_{j \in N \cup \{d\}} X_{ij} = 1, \qquad \forall i \in N \tag{1.2}$$

$$\sum_{j \in N^P} X_{o,j} = 1, \tag{1.3}$$

$$\sum_{i \in N \cup \{0\}} X_{ij} - \sum_{i \in N \cup \{d\}} X_{ji} = 0, \qquad \forall\ j \in N \tag{1.4}$$

$$\sum_{i \in N^D} X_{i,d} = 1, \tag{1.5}$$

$$X_{ij}(T_i + t_{ij} - T_j) \leq 0, \qquad \forall\ (i,j) \in A \tag{1.6}$$

$$a_i \leq T_i \leq b_i, \qquad \forall\ i \in V \tag{1.7}$$

$$X_{ij}(L_i + l_j - L_j) = 0, \qquad \forall\ (i,j) \in A \tag{1.8}$$

$$l_i \leq L_i \leq Q, \qquad \forall\ i \in N^P \tag{1.9}$$

$$0 \leq L_{n+i} \leq Q - l_i, \qquad \forall\ n+i \in N^D \tag{1.10}$$

$$L_0 = 0, \tag{1.11}$$

$$T_i + t_{i,n+i} \leq T_{n+i}, \qquad \forall\ i \in N^P \tag{1.12}$$

$$X_{ij}\ binary, \qquad \forall (i,j) \in A \tag{1.13}$$

Constraints (1.2) to (1.2) are the aforementioned visiting constraints, (6.7) are the time window constraints. Equations (1.9) and (1.10) represent capacity interval constraints at pick-up and delivery nodes respectively. Constraint (1.6) describes the relation between the flow variables and the time variables, preventing subtour formation. Constraint (1.8) describes the relation between the flow variable and the vehicle load at each node. Finally, constraint (1.12) imposes precedence relations on the associate pick-up and delivery nodes.

The objective function is quite general and includes a nonlinear penalty that depends on the total vehicle load after the service is completed at node *i*.

For passenger cases, several exact as well as heuristic algorithms for solving such a problem are reviewed in Section 2.3.3.2.

The extension to the multi-vehicle case (*m*-PDPTW) is straightforward. The notation used is a direct extension of the one presented in the previous section. Let K, indexed by k, be the set of vehicles to be routed and scheduled. Then (V^k, A^k) is the network associated with a specific vehicle k. The set $V^k = N \bigcup \{o(k), d(k)\}$ is the set of nodes with $o(k)$ and $d(k)$ denoting the origin-depot and the destination-depot of vehicle k, respectively; the set A^k contains all feasible arcs, a subset of $V^k \times V^k$. The cost and time elements, the vehicle capacities and the three types of variables are defined adequately. Note that the index k serves to define a specific vehicle and a depot location, and more generally, the initial vehicle conditions of a heterogeneous fleet of vehicles. It is assumed that each vehicle k departs empty from its origin-depot, at a specified time values given by $T^k_{o(k)} = a_{o(k)} = b_{o(k)}$.

The corresponding mathematical formulation in this case is as follows

$$\textit{Minimize} \sum_{k \in K} \sum_{(i,j) \in A^k} c^k(L_i^k) c_{ij}^k X_{ij}^k \tag{1.14}$$

subject to

$$\sum_{k \in K} \sum_{j \in N \cup \{d(k)\}} X_{ij}^k = 1, \qquad \forall i \in N^P \tag{1.15}$$

$$\sum_{j \in N^P \cup \{d(k)\}} X_{0(k),j}^k = 1, \qquad \forall k \in K \tag{1.16}$$

$$\sum_{i \in N \cup \{0(k)\}} X_{ij}^k - \sum_{i \in N \cup \{d(k)\}} X_{ji}^k = 0, \qquad \forall\, k \in K, \forall\ j \in N \tag{1.17}$$

$$\sum_{i \in N^D \cup \{0(k)\}} X_{i,d(k)}^k = 1, \qquad \forall\, k \in K \tag{1.18}$$

$$X_{ij}^k (T_i^k + t_{ij}^k - T_j^k) \le 0, \qquad \forall\, k \in K, \forall\, (i, j) \in A^k \tag{1.19}$$

$$a_i \le T_i^k \le b_i, \qquad \forall\, k \in K, \forall\, i \in V^k \tag{1.20}$$

$$X_{ij}^k (L_i^k + l_j - L_j^k) = 0, \qquad \forall\, k \in K, \forall\, (i, j) \in A^k \tag{1.21}$$

$$l_i \le L_i^k \le Q^k, \qquad \forall\, k \in K, \forall\, i \in N^P \tag{1.22}$$

$$0 \le L_{n+i}^k \le Q^k - l_i, \qquad \forall\, k \in K, \forall\, n+i \in N^D \tag{1.23}$$

$$L_{0(k)}^k = 0, \qquad \forall\, k \in K \tag{1.24}$$

$$T_i^k + t_{i,n+i}^k \le T_{n+i}^k, \qquad \forall\, k \in K, \forall\, i \in N^P \tag{1.25}$$

$$\sum_{j \in N} X_{ij}^k - \sum_{j \in N} X_{j,n+1}^k = 0, \qquad \forall\, k \in K, \forall\, i \in N^P \tag{1.26}$$

$$X_{ij}^k \ge 0, \qquad \forall\, k \in K, \forall\, (i, j) \in A^k \tag{1.27}$$

$$X_{ij}^k \ \textit{binary}, \qquad \forall\, k \in K, \forall\, (i, j) \in A^k. \tag{1.28}$$

The nonlinear objective function minimizes the total travel cost. Constraints (1.15) to (1.18) and (1.28) form a multi-commodity flow problem. Constraints (1.19) describe the compatibility requirements between routes and schedules, while (1.20) accounts for the time window constraints. Constraints (1.21) show the compatibility between routes and vehicle loads, while (1.22) and (1.23) are the capacity intervals for pick-ups and deliveries respectively. Constraints (1.24) impose an initial load condition on each vehicle. Constraints (1.25) are precedence constraints, while (1.26) ensure that the same

vehicle will visit the pick-up as well as the associated delivery. Finally, the formulation involves nonnegativity constraints on flow variables (1.27) and binary requirements (1.28).

The previous formulation is for the multi-vehicle static case. A strict dynamic formulation is not straightforward. The solution of such dynamic problems has been conducted by developing heuristic techniques and in some cases adaptation of static algorithms (see Section 2.3.3.2 for details on these techniques and a discussion).

In addition, this formulation, and particularly constraints (1.26) restrict the possibility of incorporating transfer points and fixed route segments as part of the design. In addition, the cost function does not explicitly incorporate user cost components (travel time, waiting time at pick-up points, etc), which makes the original formulation very simplistic when designing a complex transit logistic scheme. In addition, there is no way of including operational heuristic rules into the formulation in order to, for example, evenly distribute vehicles over space at any time, develop some terminal (if transfers could be incorporated) operations and management, etc.

The general problem has been solved under special requirements and conditions. In most cases, heuristic methods have been used in order to design and optimize such a system. A complete review of the most important contributions in the past is summarized in the next section.

2.3 Exact and heuristic algorithms for solving *DRT* routing and scheduling

2.3.1 Generalities

In this section a detailed literature review regarding the dial-a-ride demand responsive systems (henceforth *DRT*) is presented, emphasizing some general issues, routing algorithms, methodological procedures and field implementations. Although the *DRT* problems have been treated as a many-to-many type, the proposed scheme in the context of this dissertation (see Chapter 3) could be interpreted as a combination of many-to-one and one-to-many systems. In addition, the kind of problems commonly studied has been oriented to low transit demand systems and selected passengers satisfying very specific conditions. In this research, a high-coverage-point-to-point transit system is proposed, and therefore, the literature must be analyzed and discussed taking into account all these details and possible differences between the typical analysis and the mentioned approach.

Note that although *DRT* systems have been in existence in several cities around the US, serious research into larger-scale demand-responsive transit did not start till the 1960s. Many demonstration projects (Peoria, IL, 1964; Flint, MI, 1968; Mansfield, OH, 1970) were only marginally successful at best. The most intensive academic research into demand-responsive transit ("Dial-a-ride") was at MIT starting in 1970, in the well-known project CARS directed by Prof. Nigel Wilson. The work resulted in heuristic algorithms and a demonstration project by MIT at Rochester (Wilson and Colvin, 1977) and another demonstration project by MITRE Corporation at Haddonfield, NJ. The

generally accepted conclusion was that, perhaps due to the modest computational capabilities available then, manual dispatching performed better than computer dispatching (Black, 1985). Although this spurred a lot of research has been over the years in of the related problems, neither a successful methodology nor transit simulation results have been provided for solving large-scale demand-responsive transit systems.

As mentioned in Chapter 1, dial-a-ride problems are mostly found in demand-responsive transportation systems oriented to the service of small communities or passengers with specific requirements (elderly, disabled). These problems have been classified in the many-to-many type and include capacity constraints as well as soft time window constraints at the pick-up and delivery locations. Many-to-many demand responsive transportation systems consist of one or more multiple occupant vehicles, which take passengers from their origins to their destinations within a service area (Daganzo, 1978).

2.3.2 Technological issues

In the specialized literature it is possible to find some work in the automation of the dial-a-ride systems, which is a very important issue to be considered nowadays. Technological issues are fundamental when proposing a dynamic system with algorithms and decisions taken in real-time.

Dial (1995) proposes a modern approach to many-to-few dial-a-ride transit operation, which uses more recent off-the-shelf consumer and computer technology. He distinguishes the autonomous dial-a-ride transit system from the conventional ones, and ensures improved service and reduced costs under the new approach.

The proposed system employs fully automated order-entry and routing-and-scheduling systems that reside exclusively on board the vehicle. Here, fully automated means that under normal operation, the customer is the only human involved in the entire process of requesting a ride, assigning trips, scheduling arrivals and routing the vehicle. There are no telephone operators to receive calls, nor any central dispatchers to assign trips to vehicles, nor any human planning a route. The vehicles' computers assign trip to vehicles and plan routes optimally among themselves, and the drivers' only job is to obey instructions from their vehicle's computer.

Furthermore, the superiority of this system over conventional dial-a-ride prevails regardless of the size of the system and becomes more significant as the system gets larger.

The proposed system, called Autonomous Dial-A-Ride (ADART), is currently being field-tested in Corpus Christi, Texas, by the Regional Transportation Authority in partnership with the Volpe Center. As mentioned above, this system relies on a network of on-board computers that communicate with each other. In fact, all dispatching, routing and scheduling decisions are made by these computers on board each vehicle. The on-board computers assign trips and plan routes optimally among themselves using local heuristics.

ADART technology encompasses a high level of automation, consolidating scheduling, fare collection, credit verification, and vehicle routing into a single automated system. There is no dispatcher, and the driver's only job is to obey instructions from the vehicle's computer. Consequently, an ADART fleet covers a large service area without any centralized supervision.

Teal (1993) describes the evolution of the DRT system over the last years, and also identifies the major issues for modernizing the current and potential systems. In addition, Stone *et al*. (1993) describe the major considerations for selecting alternative software products in order to automate paratransit systems. Both works point out the potential improvements in terms of level of productivity and reliability with the ability to track vehicle locations, communicate with drivers and clients, and access traffic information on a continuous basis.

There have been some recent studies focused on the potential impact of Advanced Public Transportation Systems (APTS) on service productivity and cost. APTS technologies include Automatic Vehicle Location (AVL) Systems, Computer Aided Dispatching (CAD) Systems, and Mobil Data Terminals (MDT). Khattak and Hickman (1998) provide an overview of CAD systems. Stone *et al*. (1993) discuss several issues on dispatching and scheduling of paratransit systems using the benefits of modern computers, technology and advanced algorithms. Chira-Chavala and Venter (1997) show the improvements in productivity from using advanced technology. They study a case in Santa Clara County. Wallace (1997) quantifies the potential benefits of trip reservations by using APTS technology through a survey of paratransit customers in Southeastern Michigan. Higgins *et al*. (2000) conclude that the implementation of AVL along with advanced scheduling seems to be the primary factor in increasing efficiency by 10.3% for Houston's METROLift Service.

On the other hand, *FOCCS* system developed first in Europe (Gerland, 1991) is somewhat related to our research mainly because of technology aspects. Their objective

was to integrate railway, line-haul bus, school charter services and demand-actuated means of transport into one uniform public transport network.

Finally, Fu (2002) presents a simulation system designed to model a variety of technology-oriented dial-a-ride paratransit systems operated in an urban context. The main purpose is to evaluate the potential effects that these technologies may bring on a paratransit system. The author focuses the analysis on the evaluation of AVL and digital communication, as part of a conceptual system called APOS. An AVL model is proposed, based on GPS technology with precise definition and simulation of the communication system model, the dispatcher model, the real-time operational events, and the on-line dynamic scheduling components. An algorithm is developed for the static DARP with a set of modified constraints (see next section for details) as well. Among the major conclusions the research shows that systems with AVL have a clear advantage over ones without AVL in terms of vehicle productivity, strongly dependent on the proportion of real-time demand in the system. This conclusion seems to be evident, however, it is an important point to be considered in this research since the proposed scheme has been designed mainly for real-time demand, in which the application of technological tools such as AVL could improve the operation and coordination of the system components in a future field experiment.

2.3.3 Development of solution algorithms

In terms of development of algorithms, Gendreau and Potvin (1999) classify the research on the basis of the adopted problem-solving approach, either simple insertion procedures (Wilson and Weissberg, 1976; Wilson and Colvin, 1977; Jaw *et al.*, 1986, for example)

or repetitive application of static algorithms at each input update, (see for example Psaraftis, 1980).

Previously, Stein (1976) provided a first classification of the problem, recognizing two general approaches based not only on the solution procedure followed but also on the nature of the problem studied. He identifies the insertion approach, in which a new incoming passenger is quoted a time of arrival and must be allocated (inserting his/her origin and destination) into the prospective route of one of the buses (see for example Wilson and Weissberg, 1976; Wilson and Colvin, 1977; Daganzo, 1978; etc.). On the other hand, the second approach discussed by the author considers: a partition of the region in which the system operates, the existence of line-haul connecting the sub-regions and also the inclusion of several transfer points (where buses stop, picking up and delivering passengers). At that time, the number of transfers considered in this kind of systems was not an important issue. This is an aspect that significantly affects the demand, and therefore it must be considered in transit network design (see for example Potter, 1976).

Two very exhaustive reviews and discussions of the various approaches proposed for solving dial-a-ride problems, are found in Bodin *et al*. (1983), and Savelsbergh and Sol (1995). They divide the pickup and delivery problems into static and dynamic cases, with single and multi-vehicle and, with and without time windows. In addition, Salomon and Desrosiers (1988) published an excellent review of the vehicle routing problem with time window constraints.

In order to structure the remainder of this section, the *DRT* problems will be classified according to the following issues:

- Analytic models of many-to-many transportation systems.
- Exact and Heuristics algorithms for solving dial-a-ride routing and scheduling problems.
  - The static case without time windows (single vehicle case and multi-vehicle case)
  - The static case with time windows (single vehicle case and multi-vehicle case)
  - The dynamic case (single vehicle case and multi-vehicle case)

2.3.3.1 Analytic models of many-to-many transportation systems

The use of analytical models to describe *DRT* systems is not very common, however Daganzo (1978) develops an analytical framework to predict average waiting and riding times considering non-transfer door-to-door systems with a dynamically dispatched fleet of vehicles. He analyzes three different dispatching algorithms with a simple deterministic model, which is then generalized to capture the relevant stochastic phenomena. The three algorithms studied are: Algorithm I: After each stop, the bus is routed to the nearest feasible point (whether an origin or a destination of one of the passengers in the bus. Algorithm II: The bus alternates pick-ups and deliveries (always selecting the closest pick-up and the closest feasible destination). Algorithm III: The bus collects a fixed number of passengers and then delivers them.

The simple model developed by Daganzo allows him to predict quickly and accurately the average total time in the system. The contribution of his work is on the conceptual simplicity of his approach under both deterministic and stochastic conditions. The formulation obtained was successfully compared with simulated data, therefore,

under very simplistic conditions it is maybe a good way of implementing or modifying many-to-many demand responsive transportation systems without using complex simulation schemes. The author's premise could be correct under simplified scenarios, however under different conditions as those summarized in the previous chapter, the use of simulation schemes and more complex formulations are in most cases unavoidable.

In addition, Daganzo (1984) presents a preliminary study of the feasibility of checkpoint dial-a-ride systems, compared to a fixed route system with no transfers and door-to-door dial–a-ride systems. Analytical results show the situations in which a system is more attractive than the others in terms of cost-effectiveness. For example, he mentions that as the demand level decreases, demand responsive systems become relatively more attractive than fixed route systems, and checkpoint systems might possibly become cost-effective.

2.3.3.2 Exact and heuristics algorithms for solving dial-a-ride routing and scheduling problems:

The static case without time windows

In the first case, let us discuss the problem with a *single vehicle* of capacity *K* and *n* customers specified in advance, without time windows. The pick-up and delivery location of each customer is known. The only constraint is that the pick-up must precede the delivery of the customer.

Psaraftis (1980) develops a dynamic programming approach to solve the static case, using a linear objective function. The problem is to design a route for a vehicle

which will feasibly service all the customers known at time $t=0$. The constraints on the route are four: (i) every customer is picked-up before he is delivered, (ii) the vehicle capacity K can not be exceed, (iii) the maximum position shift constraints are satisfied and finally (iv) it is feasible to travel between adjacent activities for the route j. The purpose of the third constraint becomes important for the dynamic case, namely in preventing the possibility that the algorithm will indefinitely defer the service of any particular customer.

The objective is to minimize a weighted combination of the time needed to serve all clients and the total degree of dissatisfaction clients experience until their delivery. Dissatisfaction is assumed to be a linear function of the time each client waits to be picked up and of the time he spends riding in the vehicle until his delivery. This leads to the following objective function:

$$\min \; w_1 T + w_2 \sum_{i \in N} \left(\alpha W_i + (2 - \alpha) R_i \right) \tag{1.29}$$

where T denotes the route length, W_i the waiting time of client i from the departure time of the vehicle until the time of pickup, R_i the riding time of client i measured from pick up to delivery, and $0 \le \alpha \le 2$. The problem is solved using a straightforward dynamic programming algorithm. The state space consists of vectors $(L, k_1, k_2, .., k_n)$, where L denotes the location currently being visited ($L=0$ at the starting locations, $L=i$ at the origin of client i and $L=i+n$ at the destination of client i) and k_i denotes the status of client i ($k_i=3$ if client i has not been picked up, $k_i=2$ if client i has been picked up but has not been delivered and $k_i=1$ if client i has been delivered). Starting with state *(0,3,3,..,3)*

the algorithm explores new states preserving feasibility of the partial routes constructed so far. Due to the complexity of the algorithm of $O(n^2 3^n)$, only small instances with 10 requests at most can be handled.

Fischetti and Toth (1989) develop and additive bounding procedure that can be used in a branch-and-bound algorithm for the single-vehicle dial-a-ride problem without capacity constraints. Additive bounding refers to the use of several bounds for a problem type additively rather than separately.

Ruland and Rodin (1997) formulate the single vehicle pick-up and delivery problem as an integer program. Its polyhedral structure is explored and four classes of valid inequalities are developed. They finally develop a branch-and-cut algorithm based on the system constraints in order to solve the formulated integer program. They show computational performance of their algorithms for different problem sizes, showing a quick increase in computational execution time as the problem size increases, arguing that larger problems exhibit a tremendous growth in the search tree resulting from many iterations without generating additional constraints. They only explore problems of reduced size (no more than 15 nodes).

Psaraftis (1983b) develops an $O(n^2)$ heuristic to solve this problem based on the minimum spanning tree of the nodes of the problem. He shows that the algorithm's worst case performance is four times the length of the optimal dial-a-ride tour, using a simple two-phase approximation algorithm for the single vehicle dial-a-ride problem in the plane.

Previously, Stein (1976) conducted a probabilistic analysis of a single approximation algorithm, without capacity constraints. The algorithm construct a *TSP*

tour through all the origins and a *TSP* tour through all the destinations and then concatenates them. Stein shows that if n origin-destination pairs are randomly chosen in a region of area a using a uniform distribution, there exists a constant b such that the length Y_n^1 of the tour constructed by the algorithm satisfies

$$\lim_{n\to\infty} \frac{Y_n^1}{\sqrt{n}} = 2b\sqrt{a} \qquad \text{with probability 1.}$$

where b is a constant (recently, some empirical experiments have shown that $b \approx 0.713$). Stein (1976) also proves that if n origin-destination pairs are randomly chosen over a region of area a, using a uniform distribution, the length Y_n^* of the optimal tour satisfies

$$\lim_{n\to\infty} \frac{Y_n^*}{\sqrt{n}} = 1.89b\sqrt{a} \qquad \text{with probability 1.}$$

This implies that the algorithm has an asymptotic performance bound of *1.06* with probability *1*, which means that the algorithm solves the problem to within *1.06??*. Stein (1976) extends his analysis to the *multi-vehicle* case without time windows under a probabilistic scenario.

Cullen *et al.*(1981) used a set partitioning model as a basis for an interactive approach for solving the problem. In an interactive optimization approach, the user is embedded within the optimization algorithm, interrupting the algorithm to guide its steps by aggregating certain nodes together, forcing certain nodes apart, and so forth. The authors use an interactive approach that utilizes the concept of dual prices within a set

partitioning algorithm framework to indicate various promising alternatives to the user. The problem is decomposed into a clustering part and a chaining part. Both parts are solved in an interactive setting. The algorithm approach in both parts is based on set partitioning and column generation.

A cluster consists of a seed arc and a set of clients assigned to this seed arc. The total number of clients assigned to a seed arc may not exceed the vehicle capacity Q. Let (u^+, u^-) denote the seed arc of the cluster and let S denote the set of clients assigned to this seed arc. The cost c of serving this cluster is approximated by $c = 2\sum_{i \in S} d_{u^+ i^+} + d_{u^+ u^-} + 2\sum_{i \in S} d_{u^- i^-}$. The clustering problem can be formulated as a set-partitioning problem. Let J be the set of all possible clusters. For each $j \in J$, let c_j denote the approximate cost of serving the cluster, and for each $i \in N$, $j \in J$ let a_{ij} be a binary constant indicating whether client i is a member of cluster j or not. Furthermore, introduce a binary decision variable y_j to indicate whether a cluster is selected or not. The clustering problem is now:

$$\min \quad \sum_{j \in J} c_j y_j \tag{1.30}$$

s.t.

$$\sum_{j \in J} a_{ij} y_j = 1 \qquad \forall\, i \in N$$

$$y_j \in \{0,1\} \qquad \forall\, j \in J$$

Because the set of all possible clusters is extremely large, a column generation scheme is used to solve the linear programming relaxation of this set-partitioning problem. The master problem is initialized with all columns corresponding to clusters consisting of a

single client. The row prices $(\pi_1, \pi_2, ..., \pi_n)$ are computed and used to define a sub-problem to generate columns with negative reduced costs. The master problem now heuristically tries to improve the current solution by using some of the new columns. Then, new row prices are calculated and the sub-problem is solved again.

The sub-problem to be solved is a location-allocation problem. Let $c_{ij} = 2d_{u_j^+ i^+} + 2d_{u_j^- i^-}$ and $f_j = d_{u_j^+ u_j^-}$. For each $j \in J$ the binary variable x_j denotes whether cluster j is used or not. The sub-problem has the following mathematical programming formulation:

$$\min \quad \sum_{j \in J} \sum_{i \in N} (c_{ij} - \pi_i) z_{ij} + \sum_{j \in J} f_j x_j \tag{1.31}$$

s.t.

$$\sum_{i \in N} z_{ij} \leq Q x_j \qquad \forall\ j \in J$$

$$\sum_{j \in J} z_{ij} \leq 1 \qquad \forall\ i \in N$$

$$z_{ij} \in \{0,1\} \qquad \forall\ i \in N, \forall\ j \in J$$

$$x_j \in \{0,1\} \qquad \forall\ j \in J$$

The problem is solved approximately in the following manner: choose a set of clients to form the seed arcs. With these seed arcs fixed, solve the resulting assignment problem. With these assignment fixed, solve the resulting location problem. Continue alternating until convergence (either maximum number of iterations or no further improvement).

The clusters in the solution to the clustering problem are inputs to the chaining problem. In the chaining problem a subset of these clusters is selected and linked to form pickup and delivery routes. In the set partitioning formulation of this problem the rows

correspond to clients again, but columns now correspond to vehicle routes. The column generation procedure links a subset of the seed arcs of the selected clusters. Each linking of seed arcs is then translated into a column for the set partitioning problem by placing a 1 in row *i* if client *i* is part of one of the clusters in the linking.

Kikuchi (1984) develops a vehicle dispatching procedure designed to minimize empty vehicle travel and idle time for this problem. The objective in this paper is to minimize unproductive vehicle time, therefore, the fleet size is minimal.

The static case with time windows (single vehicle case and multi-vehicle case)

Psaraftis (1983a) modifies the dynamic programming algorithm developed in Psaraftis (1980) to solve the *single-vehicle* problem with time windows. The major difference between the new and the original algorithm is the use of forward instead of backward recursion. The new algorithm requires the same computational effort as the previous one and is able to recognize infeasible problem instances. Time windows are handled by the elimination of time infeasible states.

Kikuchi and Rhee (1989) extend the previous vehicle scheduling models developed by Kikuchi (1984) to a demand-responsive transportation system, assuming many-to-many travel demand, multiple vehicles, and time window constraint for pick-up and delivery. The model builds vehicle schedules one vehicle at the time, maximizing the number of trips requests to be handled by each vehicle. They develop a two-step model (initial route construction and subsequent insertion of additional trips on the

initial route). They use a tree structured search technique for checking feasibility along with accomplishing their final objective.

Desrosiers *et al.*(1986) also present a dynamic programming approach, which is very effective when the time windows are tight and the vehicle capacity is small.

Sexton and Bodin (1986 a, b) consider the dial-a-ride problem with desired delivery times specified by the clients. The objective used by the authors minimizes the total inconvenience that a customer may experience. Total inconvenience is expressed as a linear combination of excess ride time and deviation from the desired delivery time. This approach is also applicable if all customers specify pick-up time rather than a desired delivery time.

The solution approach applies Benders decomposition to a mixed 0-1 non-linear programming formulation, which separates the routing and scheduling component.

They use space-time separation indicators between tasks, measuring the travel time between locations at which the tasks are performed.

Let $\overline{A}_{k^-}$ be the desired delivery time of client k. The latest possible pick-up time $\overline{D}_{k^+}$ is then defined as $\overline{A}_{k^-} - t_{k^+k^-}$. The space-time separation indicators between tasks i and j are defined as follows

$$\begin{aligned} s_{i^+j^+} &= t_{i^+j^+} + \overline{D}_{j^+} - \overline{D}_{i^+} \\ s_{i^+j^-} &= t_{i^+j^-} + \overline{A}_{j^-} - \overline{D}_{i^+} \\ s_{i^-j^+} &= t_{i^-j^+} + \overline{D}_{j^+} - \overline{A}_{i^-} \\ s_{i^-j^-} &= t_{i^-j^-} + \overline{A}_{j^-} - \overline{A}_{i^-} \end{aligned} \qquad (1.32)$$

The algorithm was tested on real-life problems for up to 20 customers. The solutions obtained were considered very good compared with solutions used in practice.

The *multi-vehicle* static case with time windows is treated by Dumas *et al.*(1991). They present a set partitioning formulation for the static case and a column generation to solve it to optimality. The approach is very robust in the sense that it can be adapted easily to handle different objective functions and variants with multiple depots and a non-homogeneous fleet of vehicles. The authors develop an approximation algorithm to solve the problem, based on the creation of miniclusters, which represents segments of vehicle routs starting and ending with an empty vehicle. Every minicluster is treated as a transportation request that entirely fills a vehicle.

Ioachim *et al.* (1995) utilizes the same two-phase solution approach (cluster first, route second philosophy) in order to approximate the global routing solution allowing the insertion of additional requests during the operation day. Such insertions are facilitated when periods of free time can be joined into longer non-interrupted periods. They propose a non-linear objective function, which accomplishes this while also minimizing operational costs. The approach can solve problems of about 200 customers and 85 miniclusters in a very reasonable computational time.

Psaraftis (1986) describes and compares two algorithms for scheduling large-scale advance-request dial-a-ride systems. One is the GCR (Grouping/Clustering/Routing) described in Jaw *et al.* (1982) and the other is the algorithm developed by Jaw *et al.* (1986) called ADARTW (Advanced Dial-a-Ride with Time Windows). He gives an overview of both procedures, emphasizing the differences

in their operational scenarios, describes computational experience with both procedures and includes worst-case considerations for ADARTW.

ADARTW is one of the most traditional heuristic algorithms for solving the multi-vehicle problem. The time window constraints consist of upper bound on the amount of time by which the pick-up or delivery of a customer can deviate from the desired pick-up or delivery time. They add an additional constraint associated to the maximum time that a customer can spend riding in the vehicle. They use a traditional sequential insertion algorithm to assign customers to vehicles and to determine a time schedule of pick-ups and deliveries for each vehicle. The objective function is to minimize a combination of customer dissatisfaction and resource usage.

Time windows on both the pickup and delivery time of a client are defined based on a prescribed tolerance W and the specified desired pickup or delivery time. If client j has specified a desired pickup time (P-customer), the actual pickup time APT_j should fall within the time window $EPT_j < APT_j < EPT_j + W$; if he has specified a desired delivery time (D-customer), the actual delivery time ADT_j should fall within the window $LDT_j - W < ADT_j < LDT_j$. Moreover, his actual travel time should not exceed his maximum ride time: $LDT_j - LPT_j \leq MRT_j$. A graphic representation of these bounds is shown in Figure 2.1.

DRT_j is defined as the desired ride time for customer j. For each type-P(D) customer ADARTW derives three types of time, which, together with that customer EPT (LDT), constitute a pair of time windows (one for pickup and one for delivery) for that customer. These times are defined as follows:

For type-P customer (*EPT* specified, see Figure 2.1a):

$$\text{Latest pickup time } LPT = EPT + W \tag{1.33}$$

$$\text{Earliest delivery time } EDT = EPT + DRT$$

$$\text{Latest delivery time } LDT = LPT + MRT$$

For type-D customers (*LDT* specified, see Figure 2.1b):

$$\text{Earliest delivery time } EDT = LDT - W \tag{1.34}$$

$$\text{Latest pickup time } LDT = LDT - DRT$$

$$\text{Earliest pickup time } EPT = EDT - MRT$$

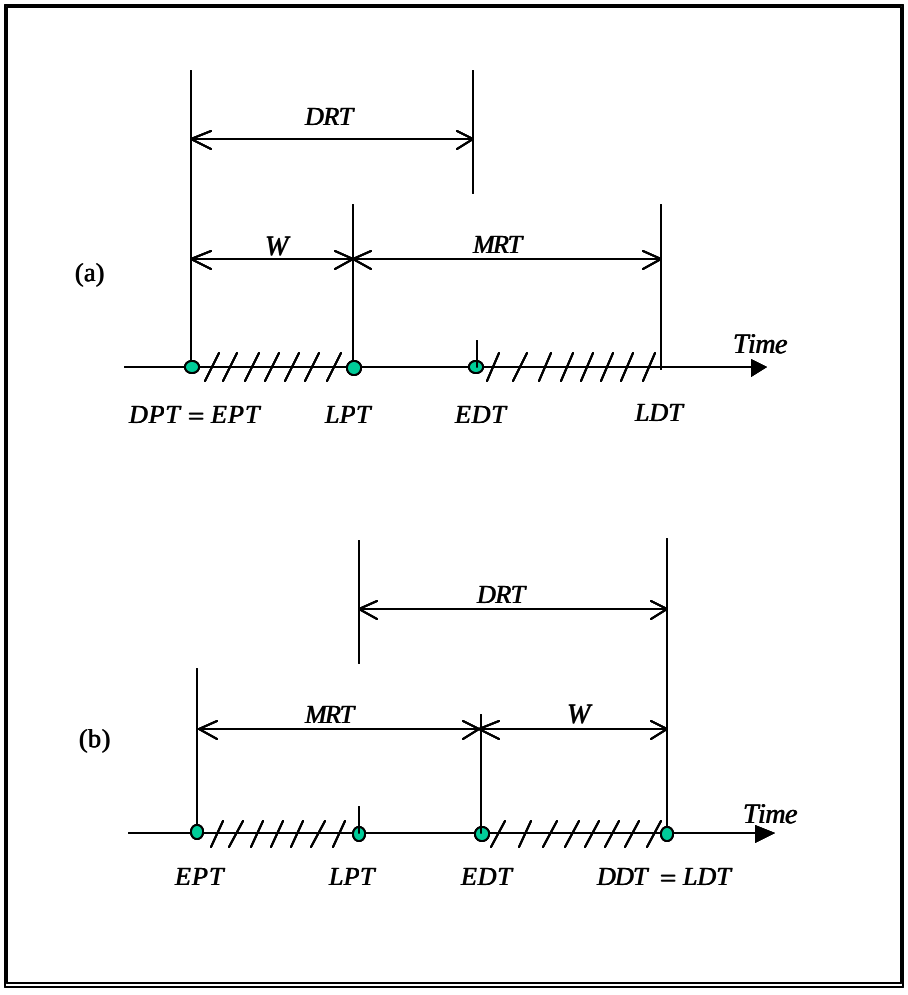

Figure 2.1 Time windows in ADARTW algorithm, (a) for type-P customers, (b) for type-D customers.

The algorithm utilizes an explicit defined objective function. The basic idea is to build routes and schedules by sequentially inserting each customer into the most promising "provisional" route and schedule constructed thus far.

Sexton and Bodin (1986 a, b) extend their algorithm for the single-vehicle dial-a-ride problem with desired delivery times to a two-phase algorithm for solving the multi-vehicle case. In the first phase, customers are assigned to vehicles and a single-vehicle dial-a-ride problem is solved to construct an initial feasible solution. In the second phase, the algorithm attempts to reassign customers with bad service to other vehicles in order to find a better solution. They basically develop a traditional cluster first, route second approach. The set of requests is partitioned into vehicle dusters and a single vehicle dial-a-ride solved for each cluster. To further reduce total user inconvenience, requests are then relocated one at a time in vehicles different than where they were originally assigned.

The most recent approximation solution procedures include the column management scheme by Davelsbergh and Sol (1998), another insertion heuristic by Madsen *et al*. (1995) and the parallel insertion heuristic developed by Toth and Vigo (1997).

The dynamic case (single vehicle case and multi-vehicle case)

The dynamic case considers the case in which calls are generated dynamically. Therefore, the system has to be examined every time a new call is generated. At that time, some of the earlier customers have been delivered to their destinations, and

therefore they are no longer included in the systems. The remaining earlier customers who have requested service have been assigned to a vehicle and are either on board the vehicle or are waiting to be picked up. Moreover, at this time, the route and schedule of each vehicle is known. The problem is to determine the assignment of the new customer to a vehicle and the new route and schedule for the vehicle that the customer is assigned to.

Note that now the assignments are carried out in real time, and therefore the algorithms must be efficient and fast.

A very interesting review about dynamic vehicle routing problems is developed by Psaraftis (1988), in which he identifies the important issues that differentiates the dynamic case with respect to the static one, highlighting some methodological issues and pointing out the generic design features that a dynamic vehicle routing procedure should posses. He also discusses the adaptation of static approaches to a dynamic setting, and describes the algorithms for the dynamic routing in a particular case.

The main differences between the dynamic and the static case according to Psaraftis are:

- Time dimension is essential;
- Problem may be open-ended;
- Future information may be imprecise or unknown;
- Near-term events are more important;
- Information update mechanisms are essential;
- Re-sequencing and reassignment decisions may be warranted;
- Faster computation times are necessary;

- Indefinite deferment mechanisms are essential;
- Objective function may be different;
- Flexibility to vary vehicle fleet size is lower;
- Queuing considerations may become important;

In addition, he provides some important design features in case of dynamic systems. He affirms that the procedure should be interactive; it should also have a "restart" capability. The system should also be hierarchically designed and user friendly.

In terms of algorithms, there are two ways to deal with such a problem.

One alternative is to adapt a static approach. The success of this procedure depends on the specific approach and setting though. A successful real-time implementation of a static approach has been reported in Bell *et al*. (1983), for the routing and scheduling of a fleet of vehicles delivering a bulk product stored at a central depot. The routing "core" of the procedure is the static algorithm of Fisher *et al.* (1982), which is based on a mixed integer programming formulation of the problem and a solution using Lagragian relaxation and multiplier adjustment method. One feasible way to adapt a static algorithm to a dynamic case is to rerun the procedure virtually from scratch each time a (significant) revision of the input occurs. An application of such an approach is found in Psaraftis (1980), who extends the dynamic programming algorithm described for the static immediate request dial-a-ride to the dynamic case. In this case, new customer requests are automatically eligible for considerations at the time they occur. The procedure is an open-ended sequence of updates, each following every new customer request. The algorithm optimizes only over known inputs and does not

anticipate future requests. The dynamic case justifies the constraint bounding the pick-up (or delivery) position shift, in order to prevent indefinite deferment.

An alternative and more commonly used adaptation would be to handle dynamic input updates via a series of "local" operations, applied via the execution of an insertion heuristic (possibly followed by an interchange heuristic), after the static core algorithm is executed. This would involve running the static algorithm just to initialize the process (say, once every day), and rely on "local" operations for all subsequent input updates.

Local operations provide a reasonable way to handle dynamic input updates; their major advantage being execution speed. The fastest local operation method is an "insertion" approach, in which a new request is inserted within the current schedule, without perturbing the sequence of visits already planned. The main drawback of these methods according to the review by Psaraftis is that it cannot take care of the need of possible resequencing or reassignment operations.

The first application of the insertion approach for solving the dynamic version of the dial-a-ride problem was carried out by Wilson and Weissberg (1976), and Wilson and Colvin (1977). The insertion involved both the pick-up and delivery locations and takes care of the implicit precedence constraint between the pick-up and the delivery.

The objective function to be minimized over all feasible insertion places includes the delay between the time of occurrence of the service request and its planned pick-up time, the ride time from the pick-up time to the delivery locations and the deviation between the planned and promised pick-up times. They also incorporate an additional term in order to spread the tour length equally among all the vehicles.

The functional form used is quadratic in the variables defined previously. The basic algorithm was later extended through deferred assignments to alleviate the myopic behavior of this simple insertion approach (Wilson and Miller, 1977). The algorithm is solved for the multi-vehicle case using the branch-and-bound technique.

Customers are broken down into classes depending upon the service demanded. Let D_{ij} be the disutility for the *ith* customer in class *j*. Then

$$D_{ij} = a_j w_i^2 + b_j R_i^2 + c_j P_i^2 \tag{1.39}$$

where a_j, b_j and c_j are parameters; w_i is the pick-up time for customer *i* minus the time for request for service for customer *i*; R_i is the ride time from pick-up to delivery for customer *i*; P_i is the actual pick-up time minus the promised pick-up time for customer *i*. Thus, if it is proposed that customer *p* in class *q* be assigned to a vehicle then, the total disutility for all customers is

$$D_{pq} + \sum_i \sum_j \left(D_{ij}^1 - D_{ij} \right) \tag{1.40}$$

where D_{ij}^1 is the disutility for customer *i* in class *j* after customer *p* in class *q* is assigned to a vehicle and D_{ij} is the same before customer *p* in class *q* is assigned to the vehicle. The additional term in the objective function that attempts to spread the tour length equally among all the vehicles, is defined for vehicle *v* as follows

$$\left(TL_v^1 - TL_v\right)\left(d\,\overline{TL} + e\,TL_v\right) \tag{1.41}$$

where d and e are parameters, TL_v^1 is the tour length for the vehicle under consideration after the new passenger is inserted into the vehicle's route, TL_v is the tour length for the vehicle under analysis before the new passenger is inserted into such a route, and $\overline{TL}$ is the mean tour length after assignment. Thus, the total objective function seeks the best vehicle v to assign the new demand. The vehicle v is the one which minimizes z where

$$z = D_{pq} + \sum_i \sum_j \left(D_{ij}^1 - D_{ij}\right) + \left(TL_v^1 - TL_v\right)\left(d\,\overline{TL} + e\,TL_v\right) \tag{1.42}$$

Wilson further extends the algorithm in order to incorporate an automatic reassignment capability that systematically reconsiders previous assignments, as part of the overall control procedure. He implements this procedure in Rochester as part of a field study (Wilson and Colvin, 1977).

A similar insertion heuristic is proposed in Roy *et al*. (1984) for the transportation of the disabled. In this application, a fair amount of requests are known in advance. Initial routes are thus constructed for those requests. The algorithm fists forms mini-clusters of customers that fit naturally together due to time-spatial proximity. A route is then constructed within each mini-cluster with an insertion procedure. Finally, the solution is improved by moving customers from one route to another. Real-time requests are incorporated in the initial solution framework using such an insertion procedure. An important feature of this work is the use of rolling horizon: the earliest

pick-up time of any given request must fall within the horizon to be considered for inclusion in the current solution.

Madsen *et al.* (1995) developed another insertion approach for a demand responsive transportation system established by the Copenhagen Fire-Fighting service for the elderly and disabled. In this case, a fair amount of requests are known in advance and are scheduled statically through an adaptation of the insertion algorithm of Jaw *et al.* (1986). Real-time requests are handled in a sequential fashion using an insertion rule that minimizes a weighted sum of components that measure the inconvenience to the new customer, the additional inconvenience to other customers already assigned to that route and various operations costs.

Interchange methods can be used after the insertion to improve further, the set of routes and schedules. Such methods are based on the concept of "k-interchange" made widely known by Lin (1965) and by Lin and Kernighan (1973) for the TSP. Again, there is a drawback of these methods, and is that they tend to become computationally prohibitive as k increases.

2.4 Mixed services: various approaches and solution algorithms

As discussed in the previous section, there are multiple technical references regarding dial-a-ride service routing-scheduling design in the literature over the last years. In contrast, a limited work is found in integrating door-to-door services with fixed route lines, usually called hybrid or mixed service in the specialized literature. The idea of integration of trunk lines and feeder services adds flexibility to the system, and also it should improve vehicle productivity mainly on trunk lines. In addition, the idea of

shifting some dial-a-ride demand to fixed routes may alleviate some of the demand pressure for the door-to-door service vehicles caused by special requirements unable to use traditional fixed route schemes (elderly, disabled, etc.).

A recent research in Malucelli *et al.* (1999) and Crainic *et al.* (2001) describes a new flexible collective transportation system. Their system considers conventional fixed route lines combined with lines based on flexible itinerary and timetable. The system exploits the idea that the integration of flexible "many to few" and "few to many" flexible systems, yields a "many to many" transportation system which is able to meet the personal transportation needs of a large customer set. The limited research in this area includes the work by Liaw *et al.* (1996), and Hickman and Blume (2000).

Both approaches are based upon developing insertion heuristics. Hickman and Blume (2000) formulate the problem of scheduling both passenger trips and vehicle trips for a proposed integrated service. The service works in three stages: the demand responsive service connects passengers from their origin to the fixed route service and (or) from the fixed route service to their final destination. The model assumes a fixed route schedule, desired pick-up and delivery points, typical time windows constraints for pick-ups, deliveries and transfers and some additional passenger level of service constraints. Using this information, they determine which trips are eligible for integrated service using the passenger level of service constraints. A schedule for both passenger and vehicle trips is created in order to minimize a measure of the cost of service. They use a modified insertion version of the heuristics by Jaw *et al.* (1986) in order to schedule integrated transit trips that accommodates both passenger and vehicle scheduling objectives.

Liaw *et al.* (1996) define the integrated mode as a bimodal dial-a-ride problem (BDARP), including paratransit vehicles as well as fixed route buses. They consider unlimited transfer as a way to combine modes. They design a decision support system (DSS), which automatically constructs efficient paratransit vehicle routes and schedules for the BDARP. Because of the combinatorial complexity of the BDARP, it may take a prohibitive amount of computational time to find a global optimal solution for large-scale problems; therefore, the authors have designed an approximated algorithm for the off-line decision support system, described as follows:

- *Step 1*: Determine an initial feasible service configuration. This includes determining the possible fixed bus route (and hence entry and exit points) to be used in serving each request, generating the associated paratransit vehicle trips, and then allocating these paratransit vehicle trips to available pieces of work. (This step is supported by the on-line DSS).
- *Step 2*: Compute the cost measure for the initial feasible service configuration. That is, find the optimal route and schedule for each piece or work using the single vehicle dial-a-ride algorithm, and then determine the total cost measure of these pieces of work.
- *Step 3*: Attempt to find a new feasible service configuration with reduced cost measure. That is, attempt to reconstruct the service configuration so that total cost measure is decreased. When a new service configuration is found with reduced total cost measure, perform the reconstruction forming new service configuration, and continue Step 3. Otherwise, go to Step 4.

- *Step 4*: Apply the on-line operation to those requests on stand-by, determine the required paratransit vehicle trips and attempt to insert them into the current schedules of pieces of work.

Liaw *et al*. (1996) insertion heuristics is tested on a data set from Ann Arbor, Michigan showing an average increase in 10% in the number of requests hat can be accommodated and an average decrease of 10% in the number of paratransit vehicles required, as compared with the manual results with no fixed route buses. Hichman and Blume (2000) on the other hand, illustrate their method using a case study of transit service in Houston, Texas, showing the advantages in cost as well as the impact on passenger level of service from implementing integrated transit service.

One strong assumption made by Hichman and Blume (2000) when specifying the generalized time or disutility function used in their algorithm, is to add a fixed transfer penalty (in their case, they add a penalty equal to 5 minutes), independent of the number of transfers realized. The number of transfer is also part of the cost function, but it only has a linear influence on the general expression. Analytically, the disutility function Z is written as follows

$$Minimize \quad Z = \beta_1 WT + \beta_2 IVT + \beta_3 XT + \beta_4 NX \tag{1.43}$$

where $Z =$ generalized time or disutility, $WT =$ total waiting time, $IVT =$ in-vehicle travel time, $XT =$ transfer time, $NX =$ number of transfers, and $\beta_1, \beta_2, \beta_3, \beta_4 =$ weights (coefficients) on each variable.

Finally, the research by Horn (2001, 2002) proposes an interesting insertion heuristic algorithm and is mixed in the sense of combining different types of transportation systems. Thus, taxis, demand-responsive services and conventional timetable services, such as buses or trains are modeled. The author develops a system with demand coming up in real-time. At the same time, mode choices as well as scheduling decisions are taken based upon a time-windowed incremental insertion procedure.

With regard to transfer disutility measures, some work has been done in measuring the impact of intermodal transfer disutility, using stated preference data (Liu *et al.*, 1997) and using a macroscopic simulation experiment (Liu *et al.*, 1998). However, none of these studies have considered the nonlinear effect in transfer disutility that, as mentioned in Chapter 1, could result in excessive perceived cost not captured by the existent methodologies and treatments. Mixed services designs have overlooked this effect, which could seriously impact the potential demand of such systems.

2.5 Final remarks

In this chapter, a complete literature review regarding flexible transit systems operations and routing/scheduling algorithms is presented. The chapter starts with the description and mathematical formulation of the general pick-up and delivery problem; then most of the specialized literature in dial-a-ride and demand responsive algorithms is presented at length. A section discussing the study of mixed services is added to the discussion.

Today, there are hundreds of dial-a-ride systems operating in the country. Most of them are small, however, using less than 20 vehicles, and dispatching them manually.

They are operated by transit agencies, private non-profit organizations, and sometimes for profit firms under contracts.

As mentioned by Black (1995), subsidization is the rule. Rarely do passenger fares cover costs, and some services are even free. The majority of the systems are limited to special clients like the elderly or disabled. Although, a lot of efforts have been devoted in developing tools and algorithms for optimizing such systems, ridership seems to be very low, implying very high operation costs involved. Fares have generally been kept low to attract passengers, increasing the needed subsidies to run these systems.

In this dissertation as mentioned in the introduction chapter, the goal is to propose a radically different transportation scheme, somewhat competitive with the automobile. The system requires high coverage, referring to the availability of a large number of vehicles, unlike traditional dial-a-ride and paratransit services, which could operate in conjunction with private and eventually paratransit systems. In the next chapter, detailed description of the concept along with some preliminary feasibility simulations are presented.

3 RESEARCH APPROACH: THE PROPOSED CONCEPT

3.1 General considerations

This chapter presents a new conceptual design and preliminary feasibility simulation results for a flexible transit system for travel from any point to any point based on real-time personalized travel desires, which is now possible due to advances in communication and computing technologies. From the discussion of the previous chapters, it is clear that the research on this area must first develop conceptual designs different from older schemes for demand-responsive transit, which have often proved to be failures.

The research direction relies on certain premises, the primary one being that the failure of earlier demand-responsive transit systems stemmed from low passenger demands caused by excessive waiting time for patrons, poor coverage of the networks by demand-responsive vehicles, and poor computational algorithms and routing capabilities. Lack of flexibility for demand-side management was an added difficulty.

The proposed system is of "*High-coverage*," referring to the availability of a large number of transit vehicles (often minibuses or vans – actually they could also operate in conjunction with private transit and paratransit systems) in a way where the travel option is at least somewhat competitive with personal auto travel. The design strictly eliminates more than one transfer for any passenger and introduces "passenger pooling" at pickup points with pooled passengers being able to travel to any destination thus avoiding a drawback of the "car-pooling" paradigm in auto travel. The cost-

effectiveness of such a system directly depends on the passenger occupancy of the vehicles, and simulations will determine what level of supply in terms of transit vehicles, and what design of routing/rerouting schemes and transfer hub locations would render the system effective in a candidate urban context.

In this chapter, the proposed concept is outlined, and results from initial simulations (which do not use some of the advanced routing schemes possible as described in Chapter 4 and 5) are provided to show that the concept can yield a transit system that is efficient.

The new system relies heavily on new technologies for implementation, primarily information systems and vehicle location identification systems such as *GPS* and map displays. Real-time operations using automatic updating of vehicle routing in real-time, and information dissemination to the passengers in the vehicles as well as at home, is a key component.

On a cursory look, the proposed system may look similar to paratransit systems (or taxis), but it has substantially different characteristics in terms of vehicle deployment, coverage design and application of technology. It is also a system ideally suited for much more efficient public investment than in conventional transit. It is perhaps deployable only by initial public investment, though it has tremendous flexibility in terms of potential private-public cooperation. A newly initiated research project (funded by the PATH program of the California Department of Transportation) is expected to yield implementable system design, routing guidelines and software, as well as lead to a field test with deployed advanced technology in the future. It is a public transit approach that could potentially remove the common criticisms about conventional

transit systems and provide alternatives with much more competitiveness to personal auto, however an initial assessment of feasibility is needed, and that is attempted in this chapter.

The primary problem faced by existing implementations of dial-a-ride systems is the low occupancies found in the buses/vans and the resulting cost of operation. Black (1995) reports some operating statistics for 20 of the larger demand-responsive systems in US for fiscal years ending in 1992. *Vehicle productivity* (average number of passengers picked up every vehicle hour) ranged from 1.44 to 5.31, with an average of 2.92. The resulting operating cost per passenger ranged from $6.23 to $18.34, with an average of $9.88. Black (1995) also mentions another performance indicator, the *level-of-service index*, which is the ratio of request-to-arrival time by dial-a-ride to door-to-door travel time by automobile. The lower the ratio, the better the service. Experience shows that the index is usually in the range of 2.0 to 3.0 (i.e., 20 to 30 minutes waiting for a 10 minute trip!). This means that demand-responsive systems tried in the past are absolutely not competitive with auto travel and thus will have low ridership, with normally only those without an available auto-option using it. The average passenger loads reported by Black (1995) are very low, showing the inefficiency. A few numbers are even less than one, meaning that a large portion of the time the vehicles are empty. That is in part why efforts have been to use the systems in smaller contexts, mostly oriented to the service of small communities or passengers with specific requirements (elderly, disabled), often driven by the requirements in the Federal acts for the disabled.

The point is clear that if demand responsive transit is to ever become cost-effective, the occupancies in the vehicles should go up, at least to 3 or 4 passengers on

average for van-type service, and 15-20 passengers for bus-type service. This will not happen unless the level of supply of vehicles is significantly higher, and the time between call and service is significantly lower to make even those who think of transit as an option to select it over personal automobiles. The waiting time at pick-up should be no more than 10 to 20 minutes. The design should also ensure that no more than one transfer is made by those traveling up to 10 or 15 miles, since this is an aspect that significantly affects the demand, and sometimes overlooked in transit network design.

The proposed scheme is described next. The scheme proposed next combines many-to-one and one-to-many operation systems, considering both a trunk and a surface network, involving high level of supply (high-coverage) and transfer hubs in demand-responsive transit.

3.2 High Coverage Point-to-Point Transit (*HCPPT*): the proposed concept

The HCPPT system may look similar to paratransit systems (or taxis), but it has substantially different characteristics in terms of vehicle deployment, coverage design, coordinated routing, and application of technology for information supply. The scheme is based on one among a number of available transit vehicles (mini-buses or vans) being rerouted in real time, as a call comes in, with the transit vehicles being assigned to certain coverage areas ("cells"), say a hexagon with one-mile edges. The transit vehicles travel to transit "hubs", one of which could, for example, be designed for say seven of such "cells" (grouped as a "cell cluster" or "zone cluster").

Each transit vehicle is assigned to a home area where it has a "reroutable" portion of its trip in a given cell, and it has a "non-reroutable" portion of travel on a trunk line to a given neighboring hub it is assigned to go to. A passenger going from a given area to another, can be routed in a vehicle that picks up and drops him/her at the adjacent area hub (the vehicle returns) from where he/she travels to the destination point on another vehicle that is reroutable there. The general scheme as well as the way of operation of *HCPPT* vehicles are shown in Figure 3.1, parts a) and b) respectively.

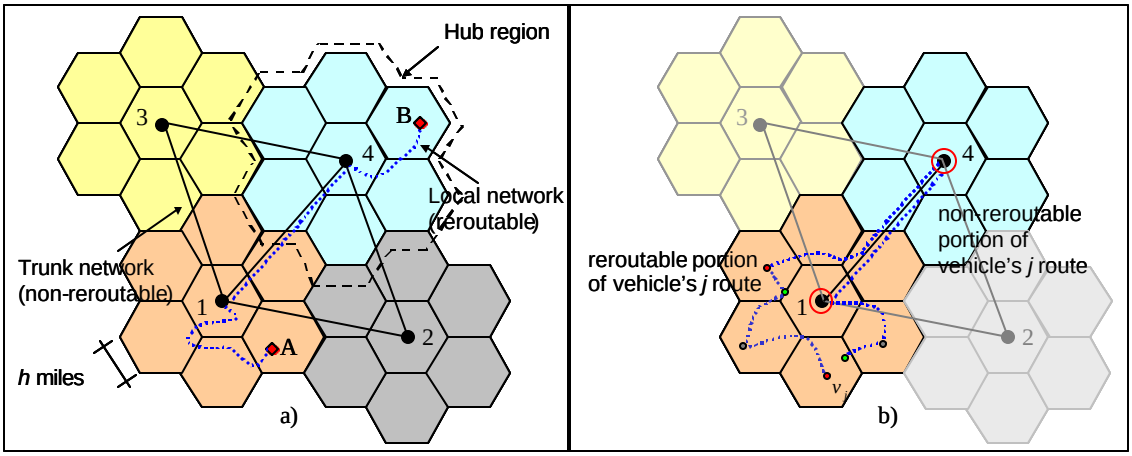

Figure 3.1 A sample scheme of the proposed concept

As an example, in Figure 3.2a, a travel from A to B can be done first on a vehicle that is reroutable in the origin cell in area 1, and assigned to go to hub 4. The remaining travel after the transfer can be in a vehicle assigned to the destination cell but returning from hubs 1, 2 or 3. An alternative option (Figure 3.2b) is to select a vehicle form the origin zone that is going to hub 2 and transferring at hub 1 to a vehicle that is returning from hub 1 to the destination cell where it is assigned to be reroutable.

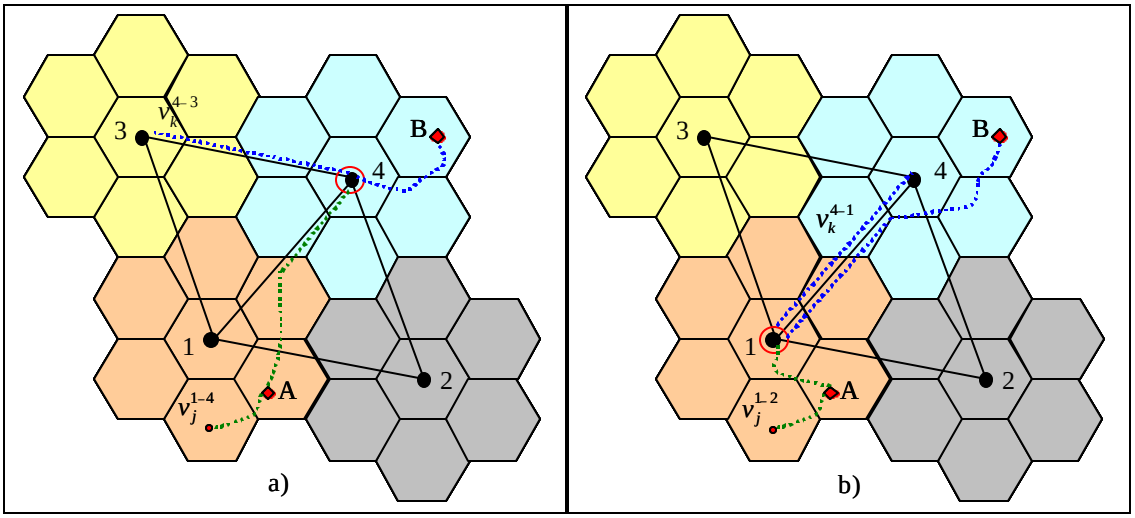

Figure 3.2 Travel options from origin A to destination B

An algorithm using real-time rerouting cost-comparisons (which are used to modify locally optimal traveling salesman tours) does the vehicle selection for passenger and vehicle rerouting. The flexibility in travel options and the optimal routing notions used cause the system to be very efficient. Appropriate cost function comparisons also ensure solutions that would improve the vehicle occupancy on the trunk lines with minimal extra waiting at the home or transfer nodes.

Most travel is restricted to within cell-clusters or between adjacent cell-clusters. If vans are to be scheduled for longer distances, there will be inefficiencies due to the number of OD pairs. Long trips are possible though, by using the same scheme presented above, but conditioned to a reduced number of vehicles when transferring at the hub location. Notice from the system that vehicles can transfer at both hubs, depending on the travel requirements of passengers on board at the time the vehicle is sent to the hub for entering the trunk portion of its route. In case of non-adjacent trips, say a trip from hub 2 to hub 3, should be served by a vehicle reroutable within hub area

2 and going to either hub 1 or 4. The passenger can be picked up there only by vehicles that are reroutable in hub 3., restricting the travel options in about half. This constraint allows the system to operate in the same flexible manned described above, but adding potential demand between non-adjacent hubs. Note that in this case the extra penalty will be only due to extra waiting time at hub, the rest of the operation and level of service should remains unaffected.

However, for an example cell size of 1-mile edges for the hexagons, travel between points up to over 15 miles is possible. The cells can have any shape according to demand patterns in real application. Note that the system makes sense only with sufficient number of vehicles in the system (2 or 3 per square mile appear to be enough).

The various issues involved in the design and implementation of such a system are addressed in this dissertation. The *HCPPT* system is particularly suited for fast real-time routing schemes based on optimized vehicle selection algorithms and decomposed local solutions for the "pick-up-and-delivery" logistics. A real-time control scheme based on constantly updated demand and vehicle-position probability distributions has been also developed (see Chapter 5 for details) and this aspect is a basic difference from many of the *DRT* systems of the past. Next, the main aspects of the scheme are listed:

- Smaller vehicles (typically 7–seat vans or even cars). Lower fares (less than 1/2 of taxis, and considerably less if subsidies are considered – perhaps as low as current transit fares) due to higher average vehicle occupancies, higher than taxis.

- Vehicle occupancy can be bettered with "passenger-pooling" or "stop-pooling" - Passengers join at the same stop, for a fare incentive. Better than car-pool due to lack of restrictions in destinations.
- Use of information technology. Internet for user calls, wireless and GPS location systems for vehicles.
- Recurrent demand from signed-up customers, as well as demand from others.
- reroutable and non-reroutable portions in the vehicle trips, with caps on number of reroutings.
- Implementation can be incrementally phased or contracted out for private fleet operators.

As introduced in Chapter 2, a system with all the technical capabilities to implement *HCPPT* is currently under field-test in Corpus Christi, TX, with federal funding, under the leadership of Robert Dial of Volpe Center (the ADART system). While that system handles higher coverage and real-time routing based on AVL technology, its routing schemes are different and it is without the one-transfer, reroutable/ non-reroutable design and logistical optimization developed here.

In this chapter, such a proposed system is studied using a discrete event simulation approach. In what follows, the simulation procedure is described, detailing the objects, system states and events occurring throughout a simulation run. In what follows, a particular implementation of the simulation concept along with some preliminary results are shown in detail.

3.3 Feasibility study

3.3.1 Simulation scheme and computational implementation

In the present section the simulation framework developed for the preliminary examination of feasibility of the proposed system is described.

The simulation was coded in C++, based on a discrete-event system simulation approach (Banks *et al.*, 1995). The most relevant event here is associated to the state of transit vehicles at any time. The random generation of service requests is the other event to be simulated under this approach. The cyclic sequence of vehicle states (tasks) is defined as follows:

- *State 1*: Waiting for being assigned to pick-up a customer.
- *State 2*: Already assigned to pick up a specific passenger and going to his(her) origin location.
- *State 3*: Going from last origin of a call location to trunk network entrance point (can be origin hub or another point).
- *State 4*: Dropping passengers at origin hub.
- *State 5*: Going from trunk network entrance point to final hub.
- *State 6*: Dropping passengers at final hub.
- *State 7*: Picking up passengers at final hub.
- *State 8*: Going from final hub to either first customer to be delivered if vehicle is full or origin hub otherwise.
- *State 9*: Picking up passengers at initial hub.
- *State 10*: *TSP* delivering of passengers.

Note that this sequence of vehicle tasks simulates in a very simple way the proposed system presented in the next section.

In *States 1* and *2*, transit vehicle j (of current position v_j) is available to pick up passengers over its assigned origin cluster zone, till it is full or till there are no more calls to pick up within the zone. The vehicle (among the available ones) fulfilling the minimum cost resulting from the call insertion in its current schedule should be the one that gets rerouted during the pick-up portion. In the experiments presented in the case study, we consider a simple cost function as a vehicle assignment indicator.

In *State 3*, if either the transit vehicle is full or there are no current calls generated, it proceeds to the trunk network and towards its assigned destination hub. We assumed a uniform distributed demand over the space, and therefore we assign our fleet homogeneously over the possible destination zones from each origin zone. Let us define HO_j and HD_j as the origin and destination hub zone of transit vehicle j respectively. The vehicle could enter the trunk network either at the origin hub HO_j or at any point along the route (the point minimizing the total travel time between the decision point and the hub destination as is shown in Figure 3.2 is chosen). If the vehicle stop at the origin hub, it normally means that some passenger's destination is different from that of the vehicle's, and that they must be dropped there (*State 4*).

Our definition of pick-up vehicle availability, as mentioned above, considers vehicles assigned to the customer area in states *1* and *3*, i.e., vehicles in their reroutable portion not currently assigned to pick-up a new customer.

Next, the vehicle travels along the trunk route till the destination hub HD_j (non-reroutable portion of the route). At the destination, all the passengers must get off from the vehicle. Every passenger waiting at the hub with destination at the assigned vehicle zone (hub HO_j) should board the vehicle till the travel vehicle is full. Next, the vehicle goes back to its origin for picking up passengers waiting to be dropped to their destinations within the corresponding cluster zone HO_j (reroutable portion of the vehicle route). This involves *States 5,6, 7,8* and *9*.

Finally an approximated algorithm is used for a *TSP* traveling salesman problem-solution in order to deliver passengers to their destination within zone HO_j. This algorithm operates in a very efficient way, and because the number of destinations to be visited by every vehicle is no more than about four to seven points, the solution should match with the real solution of the *TSP* problem. After finishing the *TSP* tour, the vehicle is available to pick up new customers (*State 1*).

Note the simplicity of the demand-responsive routing algorithms for solving *States 1 and 4*. Although it is not clear intuitively how efficient such an operational design is, the simplified simulation could give us an idea about the feasibility of the system and the advantages of developing and implementing much more sophisticated algorithms to support dynamic fleet management.

For example, note that *State 4* and *State 1* are solved sequentially, though solving the picking up and dropping off processes simultaneously would of course be more efficient, taking advantage of the low vehicle load along the *TSP* route, specially when the vehicle is delivering the last passengers of the sequence. Such operational schemes are currently being developed and implemented into the system. In the next section, we

present our preliminary simulation experiments, emphasizing all the assumptions and some interesting results obtained from them.

In Figure 3.3 the vehicle schedule described in this section is shown.

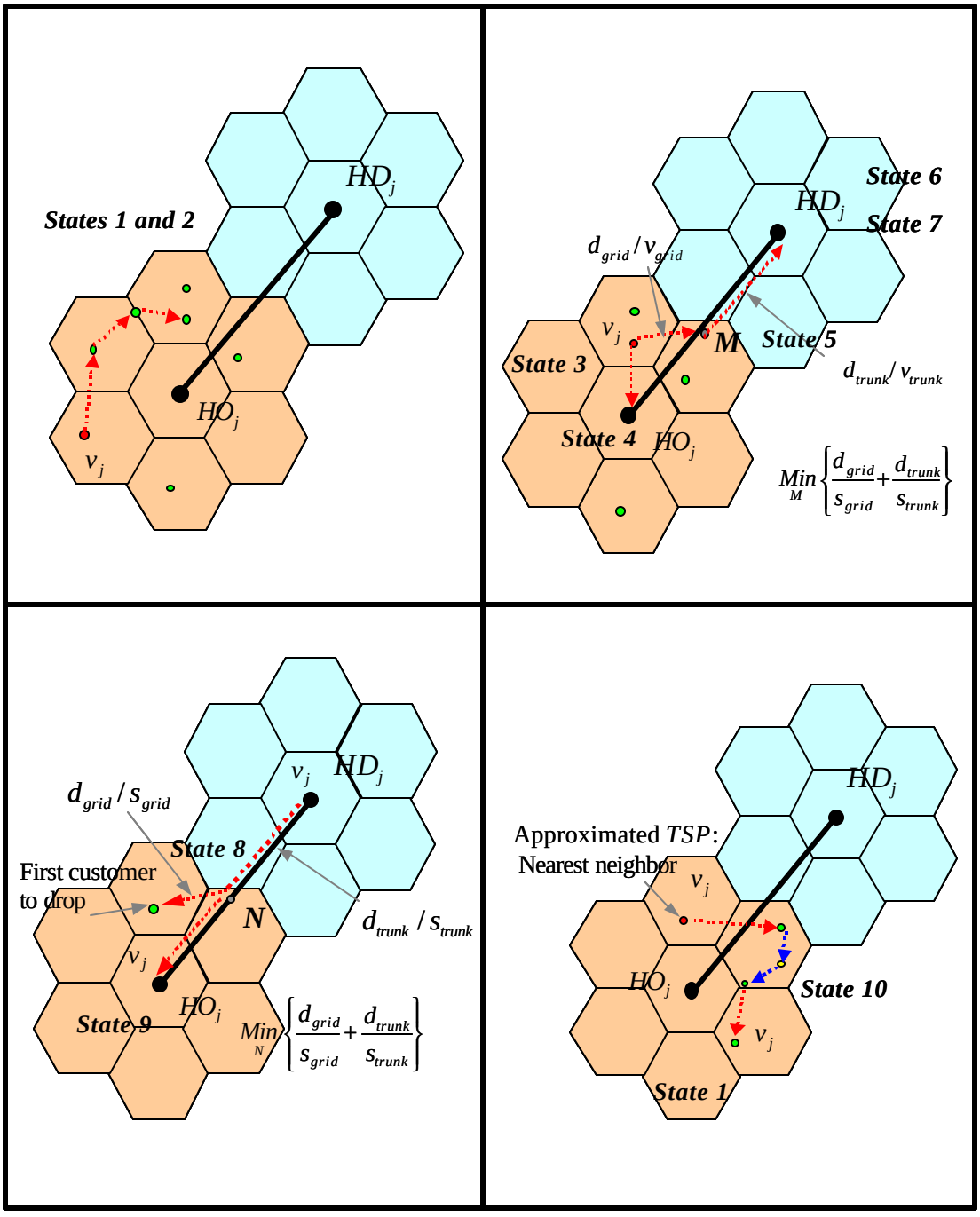

Figure 3.3 Vehicle schedule representation.

3.3.2 Simulation assumptions and preliminary results

3.3.2.1 Generalities

In this chapter, the simulations conducted are strictly to show initial feasibility. Thus certain simplifying assumptions have been made. Schemes for routing vehicles using real-time updates of probabilities and elaborate expected cost values, described later, are not used in the simulation since they require application to a real network with real-time dynamic traffic and demand conditions. In Chapters 4, 5, 6 and 7 these final routing schemes and experiments are developed.

The experiment was carried out over an abstract simplified network considering different levels of service. The simulations themselves are based on approximate travel-time calculations, to show initial feasibility. Also, the smaller cell sizes would reduce the travel times while keeping the waiting time at pick-up somewhat same, as we use essentially the same fleet size density (1 or 2 per square mile, as proposed in the scheme). Thus, the initial study is performed to look at the worst case performance of the system. As the results indicate below, we were able to find competitive fleet usage results and wait-vs-travel time results.

The proposed *HCPPT* system was simulated over an area of 165 square miles. This area is divided in four cluster zones, each containing a total of seven hexagonal cells 1.5 miles on each side. This results in an approximated distance between adjacent hubs (centroid of the cluster zone) of about 7 miles. Transit vehicles could initially be

located at any cell zone centroid, including of course the hub location itself, in order to take advantage of the terminal capacity already implemented there.

For the non-reroutable portion of the vehicle trips, the travel is assumed to be along a trunk network between hubs. Euclidean distances and a 50 mph speed are assumed for travel. On the other hand, for the reroutable portion of the vehicle routes (within cell zones) the level of service is assumed to be worse than that on the truck network. A combination of Manhattan and Euclidean travel distances and 25 mph speed are used.

The demand is generated randomly in the simulation, because demand models for this kind of service are not available currently. The call generation was assumed to follow a spatial uniform distribution over the study area. The call frequency is a random Poisson process with a mean λ (pax/ hr-square mile). In this study $\lambda = 2, 3, 4, 5$ are considered. These are numbers comparable to existing transit demand in many US cities, though the new system could potentially generate much higher demands. Thus, these are conservative numbers. For now, let us assume small vehicles like vans (with capacity say 7 pax/vehicle).

The boarding times are assumed to follow a normal distribution $N(2.0,1.0)$ per customer at pick-up location and a uniform distribution (say between 5 and 10 seconds per customer) at a hub. The alighting time is considered uniformly distributed between 0.5 and 1.0 minute per customer at the destination, and between 5 and 10 seconds per customer at a hub terminal.

The most conservative assumptions are related to the pick-up decision (involving vehicle in *States 1, 2* and *3*) and delivery (*State 10*) procedures. First, let us simulate the

case in which the transit vehicle assigned to pick-up a call already in the queue of customers waiting the one closest to the customer location at the moment of the decision, restricting this decision to vehicles in *States 1* and *3*. Second, vehicles in *State 2* are not allowed to change their just assigned passengers dynamically by a new appearing customer, even though this operation could eventually make the whole system be better off. Third, vehicles in *State 10* (dropping passengers) are not allowed to pick up a customer until delivering all the customers on the vehicles.

The approximated *TSP* algorithm tour coded for solving the passenger-drop problem is the well-known nearest-neighbor algorithm (Rosenkrantz *et al.*, 1977), which turns out to be very efficient for solving problems of this size. Its efficiency is measured in terms of computational time involved and the optimality of the resulting *TSP* tour.

In addition, certain demand-control is attempted as well. This refers to the assumption that we will have a certain average number of passengers at each pick-up. A probability of having more than one person waiting to be picked-up at any origin is assumed (say *P=1/3*). Additionally, among all the pooling origins, an additional discrete probability of having one or two additional people waiting there is considered (say *0.5* for each case). This is a relatively controllable demand pattern, as fare incentives can be used to encourage passengers to "pool" together at stops.

Finally, under the assumptions above, three hours of simulation were run. The key point to be analyzed from the results of the simulation is the trade off between cost effectiveness of the demand-responsive system (directly depending on the average load of the vehicles) and the level of service offered (measured through passenger waiting and travel times).

3.3.2.2 Pick up decision: simplified cost functions and routing algorithms

The decision taken by either a manager or the vehicle driver in order to assign a specific customer to a specific transit vehicle is, doubtlessly, one of the most difficult decisions to be modeled and simulated efficiently. A large number of heuristic algorithms based on different approaches have been developed in the literature to deal with this problem. However, none of them have been designed to take efficient pickup decisions for a dynamic system like *HCPPT*.

In Chapters 4 and 5, more complete and complex algorithms and strategies are formulated to solve this assignment decision efficiently. These strategies include issues like dynamic update of stochastic control variables, new routing schemes and so on. For this feasibility study only certain simplified system cost function expressions along with some preliminary pick-up strategies are used. Note that these pick-up algorithms have been developed independently of the *TSP* algorithms implemented for delivering passengers. A much more efficient combined strategy (simultaneous pick-up and delivery) is also implemented in Chapter 4.

First, notations are defined for use throughout the section. All the variables are defined for a specific time instant t. Henceforth, index t will be omitted in order to simplify the notation. This is just a preliminary notation useful for this example. In Chapter 4 a more rigorous notational scheme will be outlined.

General notation:

i) An arbitrary cluster zone is associated to hub $i \in H$

ii) An arbitrary customer (or call) Z_l, has the following characteristics:

- $origin\_call_l coordinates = (xo_l, yo_l) = ao_l$
- $destination\_call_l coordinates = (xd_l, yd_l) = ad_l$
- Hubs associated to Z_l:
  - HZO_l
  - HZD_l where $HZO_l \neq HZD_l$ (no internal deliveries are considered in this experiment)
- PS_l number of customers waiting at ao_l ("pool size")

iii) An arbitrary vehicle *j* has the following features:

- Position coordinates: $v_j = (xv_j, yv_j)$
- L_j: load in vehicle *j* in (pax/vehicle)
- $Vehicle\ capacity = L_{MAX}$, $\left(L_j \leq L_{MAX}\right)$
- $HO_j =$ origin (home) hub vehicle *j*
- $HD_j =$ destination (visit) hub vehicle *j*

iv) Additional performance measure:

- $E[tPHome(i)]$: Expected pick up time per pax at home (Hub *i*)

v) Travel time expressions at any time *t:*

- $tS(a,b)$: local network travel time between points a and b
- $tT(a,b)$: trunk network travel time between points a and b

where: $tS(a,b) = \frac{D_S(a,b)}{s_G}$ and $tT(a,b) = \frac{D_T(a,b)}{s_T}$

$D_S(a,b), D_T(a,b)$ are defined as the local network distance and the trunk network distance between points *a* and *b* respectively, and they are calculated as follows:

$$D_S(a,b) = d\left(\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}\right) + (1-d)\left(|x_a - x_b| + |y_a - y_b|\right) \quad (3.1)$$

$$D_T(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (3.2)$$

where $\boldsymbol{d}$ is an arbitrary parameter (we use 0.5 in our experiments) and s_G *and* s_T represent the average speed on local and trunk networks respectively.

The pick-up rules developed here will be applied to vehicles in *States 1, 2* and *3*. Basically, when a call is generated, it enters a queue of waiting customers within a specific hub zone. Among all the available vehicles j in the zone, the one with minimum cost C_j is assigned to pick up customer z_l. C_j is calculated as follows:

$$C_j = \left(L_j + PS_l\right)\frac{D_S(v_j, ao_l)}{s_G} \quad (3.3)$$

The previous expression is a simple representation of the user cost function, in which the travel time spent in picking up Z_l is weighted by the number of people already on the vehicle plus one. This additional unit represents the extra waiting time of customer Z_l if assigned to vehicle j. Different weights for travel and waiting times could be incorporated in future experiments along with an operator cost component.

From the preliminary experiments, it was found that most of the transfers occurred at the origin hub (vehicle *State 3*), because the vehicles picked up several passengers having their destination different from that of the vehicle. This operation generates inefficiency since the vehicle can not access the trunk network at the optimal point. Also, the waiting time at each hub increases because customers have a smaller chance of taking the right vehicle. In order to solve this difficulty, we incorporate a penalty value q defined as follows:

$$q \quad \begin{cases} q & \text{if } HD_j \neq HZD_l \\ 0 & \text{otherwise} \end{cases}$$

The penalty q is measured in time units. Finally, a new expression for calculating C_j is generated. Analytically:

$$C_j = \left(L_j + PS_l\right)\frac{D_S(v_j, ao_l)}{s_G} + q \tag{3.4}$$

Furthermore, if we wanted to expand the available vehicle set, we should analyze vehicles in *State 2* besides those in *States 1 and 3*. In that case, we are changing dynamically a previous pick-up decision, and therefore we should consider not only the cost associated to the $L_j + PS_l$ customers involved, but also the extra cost associated to the customer previously assigned to the route of vehicle j, n_j. Thus, if vehicle j is in *State 2*, expression (3.4) turns to

$$C_j = \left(L_j + PS_l\right)\frac{D_S(v_j, ao_l)}{s_G} + \frac{PS_{n_j}}{s_G}\left(D_S(v_j, ao_l) + \right.$$
$$\left. + D_S(ao_l, n_j) - D_S(v_j, n_j)\right) + PS_{n_j} E\left[tPHome(HO_j)\right] + q \qquad (3.5)$$

The second term in equation (3.5) represents the additional cost incurred in not picking up the customer n_j by vehicle j. In the simulations, we used $q = 15$ minutes, which appeared to provide the best routings among several values tried in preliminary experiments.

All pick-up rules described in expressions (3.4) and (3.5) are very straightforward and work reasonably well in terms of system performance and productivity levels, although there are too many factors not being considered here. It is easy to see that the system performance would improve considerably if we incorporated some more sophisticated rules to account for fundamental issues such as the dynamic nature of the system. Since calls are generated dynamically and the pick-up and delivery decisions are taken in real time based on system information at the decision time, decision rules are developed that depend on the expected number of future insertions into pre-established vehicle routes. Note that passenger assignments could change over time because of changes in system conditions. Thus, a new set of algorithms needs to be developed, considering stochastic travel times to be expected for future assignments. These travel times can be calibrated online based on historical information regarding the performance of the system, just as in adaptive predictive control systems. This capability of fine-tuning the system is what essentially brings out the optimal nature of the solutions. In this process, we essentially accomplish a notion of optimality in a problem that in a pure optimization formulation as a stochastic pick-up-and-delivery

problem has prohibitive computational requirements (as the real examples will include hundreds of vehicles and passenger calls and pick-up points). The routings are still based on optimal insertions and *TSP* calculations, but within a decomposed solution space thanks to the cell-structure of the design. However, real-time values updated for travel times and expected number of pick-ups, would yield results conceivably better than from solving approximated versions of classical pick-up-and-delivery problems. We must also mention that we have indeed generated classical optimization formulations which do not rely on such designed decompositions, but we cannot even consider providing comparative solutions, as we are unable to solve them due to the number of vehicles and demand points. A key issue in the dynamic problem is the finding of transfer points. When multiple vehicle tours are considered in the time-space, infinite possibilities exist for transfers, and solving for optimal (one-transfer) solutions adds a further dimension to an already nearly-intractable problem. This underscores the need for system designs that also help in finding dynamic solutions efficiently.

3.3.2.3 Case study results

The system performance measures discussed by Black (1995) are adopted in order to analyze the system design efficiency and performance under different simulation conditions and different demand levels as well. Let us first define the *level-of-service index* at hub level f_i as follows

$$f_i = \frac{\text{Average passenger waiting time at pick-up location}}{\text{Average door-to-door ride time}} \tag{3.6}$$

where the denominator ride time is calculated as the average *door-to-door* ride time from hub *i* to all the adjacent hub destinations. We also introduce another performance index, the *ride time index* r defined at system level. That is

$$r = \frac{\text{Average vehicle ride time}}{\text{Average door - to - door ride time}} \tag{3.7}$$

The *door-to-door* ride time is the time of travel when no other passenger is picked up. We obtained an average *door-to-door ride time* (average weighted by the number of instances per origin-destination pair) equals to 31.76 minutes, in a separate calculation

Elaborate research is needed to find "optimal" operational schemes. The intent in this feasibility study is only to show a reasonable operational scheme and to show that even such a system can have good performance. However, it was attempted to find routing procedures that yield reduced ride and waiting times.

It is beneficial to balance the vehicles' spatial distribution among the different portions of the trip at any time. Specifically, it seems that excessive ride times are due to the inefficient assignment of vehicle in *States 1, 2* and *3* when the number of available vehicles to pick-up a passenger is too low. So, if the number of available vehicles is less than one-third of the total vehicles assigned to a cluster zone (say 14 vehicles), it was decided not to assign a vehicle unless the cost involved in picking up the customer waiting for service is less than an arbitrary value C_{MAX}. Four values for C_{MAX} were tested, in time units: 10, 15, 20 and 25 minutes. Test with $C_{MAX} = \infty$, which means no cost

restriction, yielded the seam results in all cases as C_{MAX} = 25. This is because the C_{MAX} rule never applied in the network contexts due to the pickup cost being under 25 minuts. The final results are reported in Table 3.1 and Figure 3.4.

Table 3.1 shows the performance statistics for different routing patterns (resulting from various C_{MAX} values). Ride times are found to be as low as 34.43 minutes, resulting in a *ride time* index of 1.08. This index shows the rate between *door-to-door* service ride time and the resulting ride time provided by our system. So, r = *1.08* means that the ride time provided is 8% more than that provided by a competitive *door-to-door* service. For example, under the lowest demand level reported in Table 3.1, for a 10 minutes *door-to-door* trip, our service could offer the same service in about 11 minutes. On the other hand, for higher demand levels, r does not exceed 1.57, which is in any case, better than the values reported in previous *DRT* experiments.

Waiting times up to 10 or 20 minutes at pick-up (for lower demand rates), and around 30 minutes for higher demand levels, are very reasonable for this system. It is comparable to taxi services, and is better than in the *dial-a-ride* services of the past and possibly no worse than the walk and wait time in fixed route transit services as well. In addition, waiting time can be rescheduled to other activities if the pick-up is at home. f (*level of service index*) values represent the importance of the waiting time at home compared with the ride time itself. Previous *DRT* systems have shown values up to even 2 or 3 (i.e., waiting times were two to three times travel times) were unacceptable, however we found values as low as 0.26 and no greater than 1.05 in the worst case. $f = 0.26$ means that for a 20 minute trip, the user must wait around 5 minutes until service. Also waiting times at hubs are low enough (no more than 11 minutes in the

worse case). However, there is a tradeoff between these two indices, reflected through the adopted routing strategy. Figure 3.3 shows in detail the tradeoffs between the two indices for different routing patterns (resulting from various C_{MAX} values). Note the opposite tendency between indices. Also, we can appreciate that vehicle productivity increases with r and decreases with f, as expected. In other words, if we decide to impose a strategy in order to improve ride times (lowering the *ride time index*), it could result in worse productivity levels.

With regard to the productivity measure (average load in passengers/vehicle), the results could be better if it were not for the load difference between the reroutable and the non-reroutable portions (3.81 and 0.69 in the most extreme case). For details, see Table 3.2 with disaggregated information for a demand level of 5 pax/sq. mile, and C_{MAX} *= 20 minutes*. Note that there is a considerable imbalance between the loads in the reroutable and non-reroutable portions of the system.

For the particular case reported in Table 3.2, we can see that on a *Pentium III* the routing takes virtually no time at all, mostly due to the few delivery points and with picking up being handled using heuristics, which is easy under high coverage. Also the small vehicle sizes cause not more than 5 or 6 delivery points, thus an approximate *TSP* can be used. Note that computation time was just 0.0108 seconds per *TSP* function call. Computation time for routing was considered a problem in the past *DRT* system but this is not a concern in the newly proposed system.

3.3 Final remarks

The main goal of this chapter was to present a new design concept for implementable demand-responsive transit systems, which rely on real-time communication and computing technologies, and advanced routing algorithms for efficient operation. A spatial system was simulated under certain conditions and simplified routing algorithms so far. In spite of that, results show considerable improvements respect to previous *DRT* system implementations, specially considering the *level-of-service* and *ride-time* indices, between *0.*26 to 1.*05* and *1.08* to 1.*57* respectively, compared for example with *levels of services* between *2.00* to *3.00* reported in the past. However, it may be noted that the new system is not directly comparable to older *DRT* systems and that the results could be different depending on the network and demand contexts. Furthermore, the need for more depots, etc., should be studied in elaborate detail before cost comparisons can be made.

On the other hand, the vehicle productivity reflected by average passenger load is still low, but reasonable even for the simulated cases, considered a case worse than the better designs mentioned in the proposed scheme. Average passenger loads between 2 and 3 makes it much better than the 1 to 1.5 reported for earlier demand-responsive systems. This number can be improved substantially by using information technology, and also by implementing algorithms based on probabilistic system conditions in order to anticipate demand patterns, thus avoiding worthless portions of the practically empty trips.

Coordination between hub locations and use of information on waiting passengers etc., can be used for much more efficient routing schemes than attempted here. The fact that the results are still very good shows the promise of the new concept.

Average waiting times are very reasonable especially at hub terminals. Additionally, the order of magnitude of 10 to 20 minutes of waiting time at home is very good considering the location of this activity (the customer can do a different activity at home while waiting for the vehicle). Additionally, the average waiting times at hub are also reasonable. The ride times are good enough if we take into account the high demand level simulated considering a personalized transit system like this. At this point, we should emphasize that this is our first feasibility experiment, with several strong assumptions and simple algorithms to be improved in future simulations.

Finally, let us highlight an important point, taking into account that in both cases the computational efforts (in calling the routing algorithms used) are not an issue anymore, even considering a high demand levels and big fleet sizes. Twenty years ago, because of the modest computer capabilities available then, researchers realized that manual dispatching performed better than computer dispatching. Now, we can estimate insignificant computer times running the algorithms, allowing managers to take routing decisions almost in real-time.

The distributed nature of our approach appears to be what gives us benefits on the routing side, although we know that there are better routing schemes that could be developed for further efficiency and optimality, especially when real-time operations with overlapping operational zones are considered.

The pilot study certainly indicates that this system have a reasonable chance to be successful if proper fleet operations schemes are developed. This gives validity to the concepts, and detailed simulations will give much more conclusive evidence if such a system is worth testing on the field. In the next chapter a detailed set of routing and scheduling rules are formulated in the context of the above-described *HCPPT* scheme.

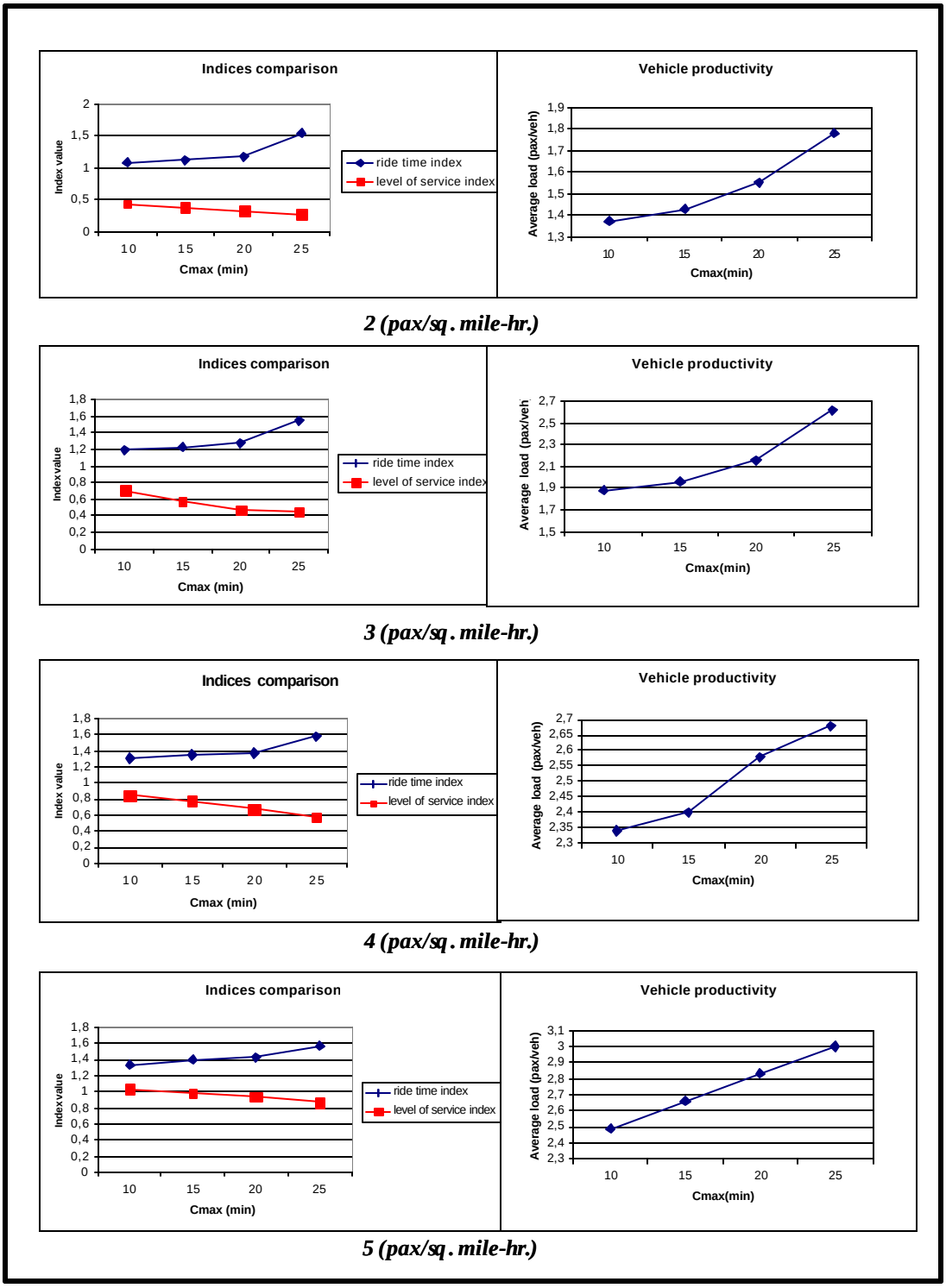

Figure 3.4 r **and** f **indices and vehicle productivity.**
(Each C_{MAX} shows a different routing pattern)

Table 3.1 Performance measures for various demand levels (C_{MAX}= *10, 25, 20, 25* result in various routing patterns)

Demand (pax/sq. mile-hr)	2				3			
C_{MAX} parameter (min)	10	15	20	25	10	15	20	25
Average veh. cycle time (min)	42.42	43.11	43.32	42.93	51.66	53.63	56.74	60.72
Average load (pax/veh.)	1.37	1.43	1.55	1.78	1.88	1.96	2.16	2.62
Percentage of free vehicles (%)	0.60	0.00	0.00	4.76	0.00	0.00	0.00	1.79
Average perc. without pax (%)	40.78	40.54	40.73	48.21	32.13	31.13	30.05	31.49
Average pax ride time (min)	34.43	35.61	37.47	48.91	38.22	39.10	40.50	49.23
Ride time index	1.08	1.12	1.18	1.54	1.20	1.23	1.28	1.55
Average wait time home (min)	13.76	11.70	9.85	8.31	22.29	18.08	14.68	14.09
Average wait time hub (min)	8.04	2.27	3.01	2.46	3.91	5.80	7.00	8.78
Level of service index	0.43	0.37	0.31	0.26	0.70	0.57	0.46	0.44
Demand (pax/sq. mile-hr)	4				5			
C_{MAX} parameter (min)	10	15	20	25	10	15	20	25
Average veh. cycle time (min)	62.26	63.50	65.60	56.87	65.78	69.81	72.97	75.82
Average load (pax/veh.)	2.34	2.40	2.58	2.68	2.49	2.66	2.83	3.00
Percentage of free vehicles (%)	0.00	0.00	0.00	0.60	0.00	0.00	0.00	0.00
Average perc. without pax (%)	26.40	25.27	24.62	34.48	23.54	23.09	21.50	21.46
Average pax ride time (min)	41.71	42.81	43.94	50.29	42.40	44.42	45.32	49.95
Ride time index	1.31	1.35	1.38	1.58	1.34	1.40	1.43	1.57
Average wait time home (min)	27.20	24.49	24.47	18.08	32.86	31.07	29.61	27.73
Average wait time hub (min)	7.12	8.39	11.05	7.29	8.62	10.72	10.84	10.37
Level of service index	0.85	0.77	0.68	0.57	1.03	0.98	0.94	0.87

Table 3.2 Details of simulation (5 pax/sq. mile-hr.) C_{MAX} = 20 minutes

			Average	Average			HUB			
HUB			travel time	load			1	2	3	4
			(min)	(pax/veh.)	% time without pax	(%)	20.41	22.04	25.38	18.17
	ongoing	r	28.82	2.52	% time no scheduled	(%)	0.38	0.44	0.53	0.40
		nr	8.75	2.93	Aver. Wait time home	(min)	35.99	28.04	23.22	31.79
1	return	nr	8.41	1.90	Aver. Wait time hub	(min)	16.03	5.90	4.38	17.05
		r	32.20	3.57	Level of service index		1.11	0.89	0.74	0.99
		total	78.18	2.93			1	2	3	4
	ongoing	r	33.10	3.02	Level of service matrix	1	-----	45.06	43.69	47.48
		nr	8.84	3.81	ride time (min/pax)	2	48.30	-----	-----	46.91
2	return	nr	8.37	1.11		3	44.10	-----	-----	45.85
		r	20.84	2.91		4	46.33	42.76	41.87	-----
		total	71.16	2.86						
	ongoing	r	31.49	2.62	Average ride time			45.32	(min)	
		nr	8.75	3.81	Ride time index			1.43		
3	return	nr	8.34	0.69	Average load			2.83	(pax/veh)	
		r	15.84	2.93	Percentage of free vehicles			0.000	(%)	
		total	64.41	2.61						
	ongoing	r	28.82	2.54			HUB			
		nr	8.79	3.11			1	2	3	4
4	return	nr	8.53	1.84	Demand (pax/UT)	1	-----	123	107	84
		r	31.99	3.37		2	167	-----	-----	153
		total	78.12	2.87		3	143	-----	-----	149
Traveling salesman problem approximated				Algorithm		4	91	135	108	-----
(two or more pax delivery)										
Function called			252	(times)			1	2	3	4
Total resource time			2.73	(secs)	Perc. Last hub (%)	1	-----	61.79	62.62	48.81
Average resource time function			0.0108	(secs)		2	54.49	-----	-----	59.48
Average number of pax dropped			5.643	(pax/TSP)		3	65.73	-----	-----	61.07
						4	61.54	63.70	65.74	-----
One pax delivery			21	(times)	Average perc.last hub	(%)				60.50

r: reroutable portion of the trip
nr: non-reroutable portion of the trip

4 HEURISTIC RULES FOR ROUTING-SCHEDULING *HCPPT*: THE DYNAMIC CASE

4.1 Generalities

All pick-up rules described in Chapter 3 for the feasibility study are very straightforward and work reasonably well in terms of system performance and productivity levels, although there are many factors not considered in that example. It is easy to see that the system performance should improve considerably if some more sophisticated rules are incorporated to account for fundamental issues such as considering the interaction among system entities at different states, combining pick-up and delivery operations in the same local optimization decision, improving the insertion algorithm and cost formulation, including terminal management for reducing the *TSP* tour lengths and most importantly being aware of the dynamic nature of the system when taking scheduling-routing decisions.

The inherent design of the *HCPPT* scheme allows the modeler to decompose the large-scale system into smaller pieces, and to formulate local optimization sub-problems that are apparently mutually independent of each other. The routings are still based on optimal insertion algorithms and *TSP* calculations, but within a decomposed solution space thanks to the cell-structure of the design.

In a traditional optimization scheme, the formulation itself would become simpler than the sophisticated routing-scheduling rules developed in this chapter (for example, see the PDPTW formulation in Chapter 2), however, the final global

optimization procedure would become unfeasible and ultimately unable to handle such a big system (in terms of vehicles as well as passengers) when decisions have to be taken in real time. Moreover, traditional routing vehicle problems are very restrictive in terms of system design. In fact, it is hard to imagine a strict non-linear optimization problem formulation and solution of a *HCPPT* kind of scheme. In other words, decomposing the *HCPPT* problem gives a notion of optimality that is computationally prohibitive in a pure optimization formulation as a stochastic pick-up-and-delivery problem. Classical optimization formulations, as those showed in Chapter 2 which do not rely on such designed decompositions were explored, but it was impossible even to consider providing comparative solutions, as they were unable to be solved due to the number of vehicles and demand points.

Moreover, in that solution process, it seems illogical to perform local optimization without considering the interactive effects at least in the vicinity of the candidate solution. Along with the notion of local optimality, there are some global considerations to be taken into account. This fact justifies the complexity of the routing and scheduling rules developed here.

Besides, since calls are generated dynamically and decisions are taken in real time based on system performance, the aforementioned routing rules will also depend on the expected number of future insertions into pre-established vehicle routes. Note that passenger assignments could change over time because of changes in system conditions. Thus, the proposed algorithms consider stochastic travel times to be expected for future assignments. These travel times can be calibrated online based on historical information about the performance of the system, just as in adaptive-predictive control systems (see

Chapter 5 for details). This capability of fine-tuning the system is what essentially brings out the optimal nature of the solutions. Real-time values updated for travel times and expected number of pick-ups, would yield results conceivably better than from solving approximated versions of classical pick-up-and-delivery problems.

The new heuristics are based on the following criterion: when the dispatching program receives a new pick-up request, it evaluates the additional cost over the known system (including operators and passengers) resulting from the insertion of such a customer into the expected route of all candidate vehicles. Then, the requested service is assigned to that vehicle with the minimum incremental cost. The important issue here is to have correct expressions for such generalized costs. Such a pick-up decision could eventually impact the status of the system at a hub area level, since the chosen vehicle could be a good candidate to serve other pending pick-up requests assigned to different vehicles at a lower cost. If so, those vehicles could potentially change their schedules and routes too, resulting in a series of vehicle route adjustments until no further improvements occur. Such heuristics are proposed in this chapter to reasonably adjust vehicle schedules and routes in order to deal with this dynamic local imbalance, without adding excessive complexity to the formulations.

Let us illustrate these new stochastic concepts with a simple example. Suppose that two vehicles (say, vehicles j and k) are available to pick-up new requests within certain hub region, and they currently have a pre-established sequence of stops, including scheduled pick-ups and deliveries (so far, z_m for pick-up m and d_n^j regarding the *nth* delivery associated to vehicle j) as shown in Figure 4.1.

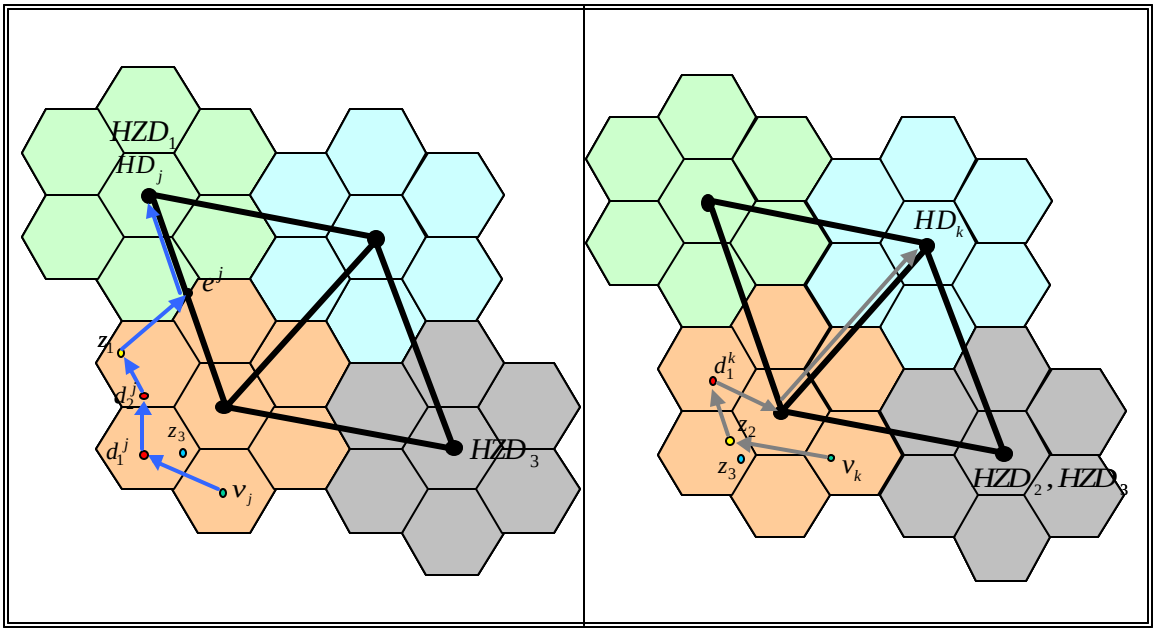

Figure 4.1 Initial schedules of vehicles *j* and *k*

A more precise notation for pick-ups, deliveries and vehicle schedules will be needed in the development of more complex formulations. All these details will be introduced in Section 4.2.

Note that vehicle k has to access the trunk network at the hub position since the destination hub of its scheduled pick-up z_2 is different from its own hub. At time t (when vehicles are in positions v_j and v_k) a new pick-up request z_3 is introduced in the system. There are two options for the manager to schedule that request: whether into the route of vehicle j or into the route of vehicle k. Suppose that in both cases, the best option is to insert the new request immediately. That means, before delivering passenger d_1^j in case of vehicle j, and before picking-up group of passengers z_2 waiting at their origin in case of vehicle k. A strict cost function formulation should incorporate all these

effects when computing the additional cost incurred in making such an insertion. The two candidate insertions are shown in Figure 4.2.

Note that after inserting z_3 into vehicle j's route, it has to stop at the origin hub before continuing until its associated destination hub, since customer z_3 is traveling towards a different destination. Therefore, the manager must decide which insertion results in a lower cost to the system over the base situation in Figure 4.1. For this particular case, the decision will be based on

$$\begin{aligned} IC_j(z_3) &= CI_j(z_3) - C0_j \\ IC_k(z_3) &= CI_k(z_3) - C0_k \end{aligned} \tag{4.1}$$

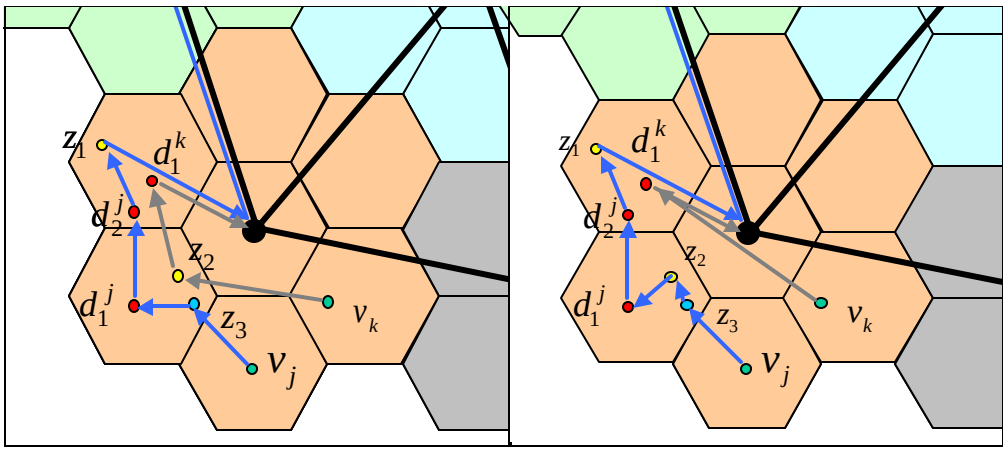

Figure 4.2 Candidate insertions into the schedule of vehicles *j* and *k*

where $CI_j(z_3)$ represent the cost incurred by vehicle j on all its scheduled passengers (on board and waiting to be served) in case of inserting the new customer into its original route. $C0_j$ on the other hand, represents the cost on the system if vehicle j continues to follow its original schedule. For this particular case, expression (4.1) is calculated by

summing the incremental cost between stops over the entire sequence. The manager will assign customer z_3 to that vehicle with a lower IC. Suppose that $IC_j(z_3) < IC_k(z_3)$. Thus, customer z_3 will be inserted into vehicle j's route. After taking that decision, the system needs to be adjusted since vehicle j could now eventually serve a customer already scheduled to another vehicle's route (say, route k), if that decision represented a potential improvement. Besides, the cost criterion remains the same, with the difference that in this case it will be necessary to compute the incremental cost of adding one customer to one vehicle's route, and compare that increment in cost with the decrease in cost due to excluding that customer from his/her original scheduled route. In the example, if customer z_3 were assigned to vehicle j, this vehicle could also pick-up customer z_2 originally assigned to vehicle k. In fact, if the incremental cost of inserting z_2 to vehicle j route $IC_j(z_2)$ is smaller than the benefit of excluding that customer from k's route $IB_k(z_2)$, the manager could take the decision of adjusting both vehicle schedules. The possible impact of such a modification is illustrated in Figure 4.3. The diagram on the left shows the original assignment while the one on the right shows the impact on the system after this adjustment.

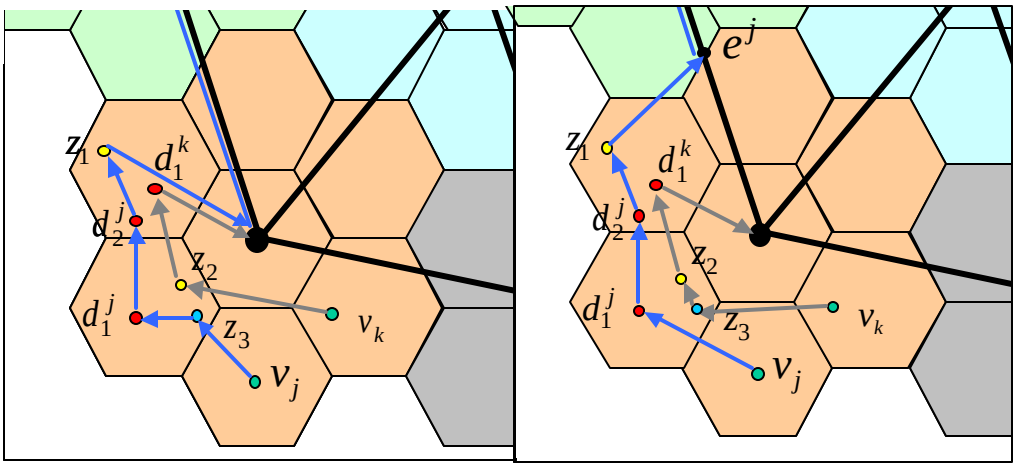

Figure 4.3 Route adjustment impact

Given the dynamic nature of these decisions over time, it becomes evident that in order to take realistic pick-up decisions, we should accept that travel time variables in the cost formulations are stochastic, and they will depend on the expected number of insertions not scheduled when taking any pick-up decision. Let us illustrate this new concept with an example:

Before insertion, vehicle j had scheduled a sequence of stops $\{d_1^j, d_2^j, z_1, HD_j\}$. If conditions had remained invariant over time, the travel time spent from the current location and the first delivery d_1^j would simply be the travel time between the two physical points, given the network traffic conditions at the time of that time. If vehicle j were assigned to pick customer z_3, the travel time between the same two points would increase considerably (1 intermediate stop), and if the adjustment situation in Figure 4.3 occurred, it would increase even more (2 intermediate stops). This uncertainty in travel

time due to extra stochastic insertions over time is directly related to the cost expressions, which depend explicitly on travel time, and therefore, is strongly related to the decision rules.

Figure 4.4 shows the possible values that the travel time between vehicle *j*'s current position and its first delivery, based only on the possible situations described above. The possible combinations can become increasingly complicated, depending on the demand, the number of vehicles available, and so on.

For the simple case in Figure 4.4, the future vehicle assignments are not known with certainty, it could be assumed that there is a probability P_1 that vehicle j fulfills its first delivery assignment without any reschedule (diagram A). If, on the other hand, vehicle j is deviated to pick-up z_3 (diagram B) with probability $(1-P_1)$, there is also a probability that from there, vehicle j could pick-up an extra passenger z_2, with probability $(1-P_2)$, as in diagram C.

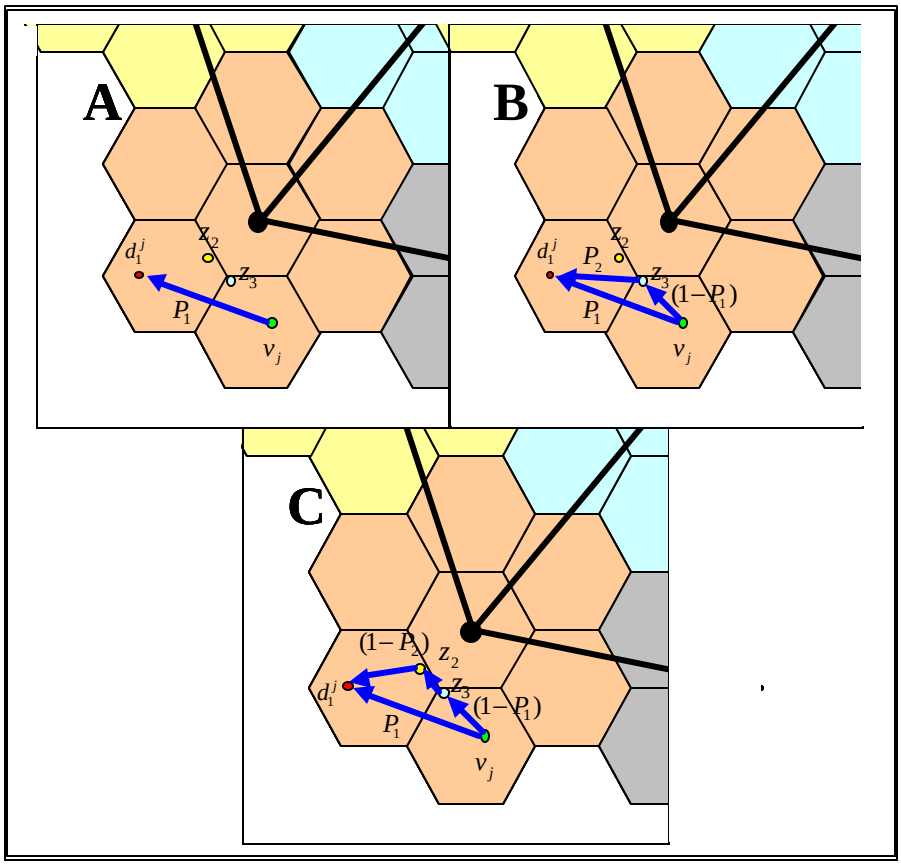

Figure 4.4 Stochasticity in travel time

Thus, under these simple conditions, if $tSN(v_j, d_1^j)$ is the base travel time between the vehicle position and its first delivery, the expected travel time for arriving at its first delivery will be

$$E[tSN(v_j, d_1^j)] = P_1\, tSN(v_j, d_1^j) + (1-P_1)\big[tSN(v_j, z_3) + \\ + (1-P_2)\big\{tSN(z_3, z_2) + tSN(z_2, d_1^j)\big\} + P_2\, tSN(z_3, d_1^j)\big] \tag{4.2}$$

The variables on the right hand side are not rigorously defined yet and it will be done later in this chapter. Expression (4.2) is just to show that the uncertainty of travel time coming from unexpected vehicle reassignments could be significant and relevant when computing pick-up insertion cost comparisons among several alternatives. A practical way of estimating these expected values and probabilities become much more complicated under more general conditions. However, ignoring this stochastic effect could yield sub optimal routing-scheduling decisions taken by the dispatching module module when running the system. In Chapter 5, a detailed theory is developed in order to consistently calibrate expressions such as (4.2) in real time using information obtained from the system itself. The assumption behind the proposed approach is that the expected travel time as in (4.2) depends upon both the base travel time and the expected number of non-scheduled insertions, as a function of vehicle and system conditions (in this case, potentially z_2 and z_3 with certain probability).

The main goal of this chapter is to consistently formulate incremental cost expressions for vehicle routing and scheduling in real-time under the *HCPPT* scheme operation. At this point, expected travel times from expected number of non-scheduled insertions are assumed known and computable. In Chapter 5, a way of estimating these expected travel times from system and vehicle conditions using an adaptive-predictive control approach is developed at length.

In addition, routing and scheduling rules are developed for both reroutable and non-reroutable portion of vehicle routes according to the *HCPPT* scheme. Moreover, some heuristics are developed in order to optimize deployment and distribution of vehicles at any time. Additional queuing rules and terminal operations are proposed for

minimizing waiting times at hubs along with clustering *TSP* delivery tours. The computer implementation and performance of insertion algorithms are also described in Sections 4.3.3, 4.4 and 4.5. A final discussion regarding the assumptions and conditions for applying the *HCPPT* routing schemes on real networks is conducted in Section 4.7.

In the next section, all general terms and notation to be used in the context of the dissertation (and mainly in this chapter) are defined.

4.2 Basic definitions

First of all, the mathematical notation that we will use throughout the entire chapter is analytically defined. All the variables will be formulated for a specific decision time *t*. Henceforth; any index *t* will be omitted in order to simplify the notation:

- Set of hubs or major interchange terminals $H = \{1,2,\ldots,N_H\}$ where N_H is the number of cluster zones operated by the transit system.
- An arbitrary cluster or hub area *i* is defined by:
  - Corresponding major interchange terminal, hub $i \in H$
  - $ah(i) = (x_h(i), y_h(i))$: physical location of centroid or terminal $i \in H$ at any time.
- Set of hexagonal cells $C = \{1,2,\ldots,N_C\}$ where N_C is the number of cells operated by the transit system.
- An arbitrary cell area *n* is defined by:
  - Associated major interchange terminal, hub $ch(n) \in H$
  - $ac(n) = (x_c(n), y_c(n))$: physical location of cell-centroid $n \in C$ at any time.
- At decision time *t*, the following state definitions are associated with vehicle *j*:

- $v_j = (xv_j, yv_j)$: current vehicle position
- $HO_j \in H$: Hub associated with the reroutable portion of the vehicle j route. Hereafter, the"home hub for vehicle j".
- $HD_j \in H$: Hub associated with the non-reroutable (trunk) portion of the vehicle j route, defined according the *HCPPT* system description, hereafter "visit hub of vehicle j". Note that always $HO_j \neq HD_j \quad \forall j$
- CS_j : sequence of stops scheduled for vehicle j from the current position v_j (position 0 in CS_j) till visit hub HD_j. In set notation $CS_j = \{0,1,2,\ldots,N_j+3\}$

  CS_j defines the scheduled vehicle path into the hub zone, which can change over time, according to the embedded dynamic rerouting and rescheduling schemes. The cardinality of this set of stops is $N_j + 3$. The number of stops regarding any pick-up or delivery operation is N_j. The remaining three elements in the sequence account for the transition between reroutable and non-reroutable portion of the route as will be discussed later in this chapter.
- Correspondence between stop sequence numeration and physical position (Cartesian coordinates) of any stop that belongs to CS_j :

$$a_j(s) = (x_j(s), y_j(s)) \qquad \forall s \in CS_j$$

  thus, $a_j(0) = (x_j(0), y_j(0)) \equiv v_j = (xv_j, yv_j)$
- $LP_j(0)$: Number of passengers just picked up from area hub HO_j by vehicle j when leaving v_j.

- $LD_j(0)$: Number of passengers to be dropped off at their destination within area hub HO_j by vehicle j when leaving v_j, where $LD_j(0) = LDI_j(0) + LDE_j(0)$, if the delivery load is disaggregated into internal and external deliveries.
- $L_j(0)$: vehicle *j* load, when leaving v_j in (pax/vehicle).

Therefore for a generic stop *i*,

$$L_j(i) = LP_j(i) + LD_j(i) = LP_j(i) + LDI_j(i) + LDE_j(i)$$

- $ZB_j = \{1, 2, \ldots, L_j(0)\}$: set of passengers on board when vehicle leaves position v_j. This set can be split into:

  $ZBP_j = \{1, 2, \ldots, LP_j(0)\}$: subset of passengers just picked up

  $ZBD_j = \{LP_j(0)+1, \ldots, L_j(0)\}$: subset of passengers to be dropped off (including both internal and external trips)
- Vehicle capacity $= L_{MAX}$, therefore always $(L_j \le L_{MAX})$

- Z_l: random customer request. The group of customers Z_l, is composed of PS_l individual customers, having their origin at $ao_l = (xo_l, yo_l)$. In mathematical notation, every service request (or group of customers) takes the form $Z_l = \{1, 2, \ldots, PS_l\}$. Note that here the concept of transit pooling is being introduced into the analysis. In addition, the individual customers in Z_l, can be split into internal and external trips. The former customers are those who are traveling within the same hub area, while the latter are those who travel towards an adjacent hub. Analytically, two additional subsets for internal and external trips are defined accounting for the aforementioned sub-sets:

$$ZI_l = \{1, 2, \ldots, IS_l\}$$

$$ZE_l = \{IS_l + 1, \ldots, PS_l\}$$

where $ZI_l \cup ZE_l = Z_l$. IS_l and ES_l represent the cardinality of the internal and external trip subsets respectively; therefore, $IS_l + ES_l = PS_l$. The destination will be different depending on the individual customer. That is $ad_l(n) = (xd_l(n), yd_l(n)) \quad \forall\, n \in Z_l$.

- $Z_l(n)$: n^{th} passenger than belong to group *l*.
- Hubs associated with group of customers Z_l:
  - Origin hub HZO_l
  - Destination hub, which can be different for any individual customer $HZD_l(n) \quad \forall\, n \in ZE_l$. In case of internal trips, origin and destination hubs are both the same.
  - Destination cell $CZD_l(n) \quad \forall\, n \in Z_l$.
- $ZA_j(k)$: vehicle *j*'s scheduled pick-up requests at time after vehicle *j* leaves stop $k \in CS_j$. $ZA_j(k) = \left\{ \bigcup_r Z_r \right\}$: where each group of customers *r* has been scheduled to be served by vehicle *j* and when vehicle *j* is at stop *k*, they are waiting for the it to pick them up at their pick-up spot. In case of assignments occurring at a hub stop: $ZA_j(k) = \left\{ \bigcup_{r,n} Z_r(n) \right\}$. The cardinality of $ZA_j(k)$ is $NZA_j(k)$.

$ZA_j(0)$ denotes the scheduled pick-up requests at time t, that is, at the current decision time.

- ZC: set of customers in the system at time t.

- Correspondence passenger-customer. Let us define the following one-to-one mapping between passengers on board and set of customers in the system:

$$\forall\, m \in ZB_j, \exists\, Z_l(n) \in Z_l, Z_l \subset ZC \begin{cases} zb_j(m) = l \\ zbp_j(m) = n \end{cases}$$

- Correspondence sequence-customers. Let us define the following one-to-one mapping between sequence elements and customers in the system:

$$\forall\, k \in CS_j \begin{cases} z_j(k) = l \\ zp_j(k) = n \end{cases}$$

For the k^{th} element of this sequence CS_j, there exist the following options:

If $z_j(k) > 0 \Rightarrow \exists\, Z_l \subset ZA_j(0) : z_j(k) = l$

If $(zp_j(k) > 0 \wedge zp_j(k) \leq ZI_l)$, then the k^{th} element in the sequence is an internal delivery and corresponds to the n^{th} element of Z_l.

Else if $(zp_j(k) > 0 \wedge zp_j(k) > ZI_l)$, then the k^{th} element in the sequence is an external delivery and corresponds to the n^{th} element of Z_l.

Else, i.e. $zp_j(k) = 0$, then the k^{th} element in the sequence is the stop for picking up group Z_l.

Else, i.e. $z_j(k) = 0$, then the k^{th} element in the sequence is not a stop related to any customer in $ZA_j(0)$. It corresponds to a delivery of an on-board passenger instead. Analytically, $zp_j(k) \in ZB_j$ identifies such a correspondence.

- Additional performance measures:
  - $E[D(i)]$: Expected number of passengers/vehicle dropped at hub $i \in H$
  - $E[P(i)]$: Expected number of passengers/vehicle picked up at hub $i \in H$
  - $E[tDHub(i)]$: Expected drop time per passenger at hub $i \in H$
  - $E[tPHub(i)]$: Expected pick up time per passenger at hub $i \in H$
  - $E[tDHome(i)]$: Expected drop time per passenger at home (Hub area $i \in H$)
  - $E[tPHome(i)]$: Expected pick up time per passenger at home (Hub area $i \in H$)
  - $E[WTHub(i)]$: Expected waiting time per passenger at hub $i \in H$
  - $E[WTHome(i)]$: Expected waiting time per passenger at home (Hub area $i \in H$)
- Assumed operator cost OC in $[US\$/\min]$.
- In case of passenger cost, different weights were assumed for travel and waiting times. Analytically ATT and AWT will represent the value of travel and waiting time respectively, perceived by any traveler, both in $[US\$/\min]$. Theoretically, it is expected the value of time to change over time, due to the effect of customer "impatience". Some authors have treated these variables as continuous functions of time, but in most cases they are assumed to remain constant, mainly because of

simplicity. In the context of this analysis, both ATT and AWT were discretized, allowing them to take two different values in order to capture the effect of customer impatience.

In the next sections, it is assumed that those passengers who have traveled a longer portion of their trip (those who come from a neighboring hub) perceive the value of travel time at a different value (ATT_2 instead of ATT_1 for all other travelers). In other words, it has been recognized that the value of time is really a function of the time itself; however, it was decided to use a discrete formulation for the sake of simplicity.

In addition a constant multiplicative weight ACT is defined for the cumulative travel time of passengers on board in the cost function formulation.

In the next section, an incremental cost function formulation is presented in order to assign a vehicle to pick up a new pick-up service request, when the vehicle is moving within the reroutable portion of its route. The methodology is based on system-wide cost minimization criteria. The objective is to compare all feasible insertions of a specific customer into the route schedules of all available vehicles. In order to do that, an objective function is designed to evaluate the incremental cost of each insertion into a vehicle route based on current information along with certain system performance measures. After evaluating this incremental cost function for every available vehicle, the new group of customers is assigned to that vehicle with the minimum incremental cost.

This cost incorporates both user and operator components combined into one general expression.

4.3 Scheduling-routing rules: vehicle assignment to serve a new pick-up request

The objective of this section is to specify the routing rules for assigning a group of customers' service request to be picked-up by a specific transit vehicle. The decision is taken in real time and the demand is assumed to be unknown in advance. The assignment problem is decomposed into pieces, and the final goal of the assignment will be to optimally bring the customers to the proper adjacent hub from where another vehicle will take them to their final destination. In this section the formulation for optimizing the assignment of the first leg of their trip is developed, independently of the second leg of their trip from the transfer hub until their final destination according to the operational scheme presented in Chapter 3. The second stage of the passenger trip treatment will be discussed in Section 4.5.

Although the optimization is centered on pick-up decisions, passengers to be delivered are considered in the cost expressions when taking routing decisions. In addition, internal deliveries (with origin and destination in the same hub area) will be included in the routing decision as well.

The section has been split into two pieces. In Section 4.3.1, the analysis of the deterministic case is shown, detaining all basic cost components and showing fairness of the assignment decision, but assuming that the cost expressions will depend only on the system conditions (vehicle travel time, vehicle load and vehicle assignments) that are

known at decision time t (which is the time of the new call). In Section 4.3.2, the inclusion of stochasticity in travel time and vehicle load calculations is added to the cost functions according to the scheme discussed above in Section 4.1.

4.3.1 Deterministic case

Let us assume we know the traffic conditions in the physical network at any time. Then, it is possible to distinguish the local network travel time between two points in the surface area, $tSN(a,b)$ from the trunk network travel time between two points along express corridors $tTN(c,d)$. Points a,b,c,d are completely identified by their Cartesian coordinates on the 2-dimensional plane.

At time t, an arbitrary vehicle j in the reroutable portion of its route is assigned to follow a sequence of stops CS_j from its current position v_j. At v_j it will be assumed that all vehicle conditions and features are known.

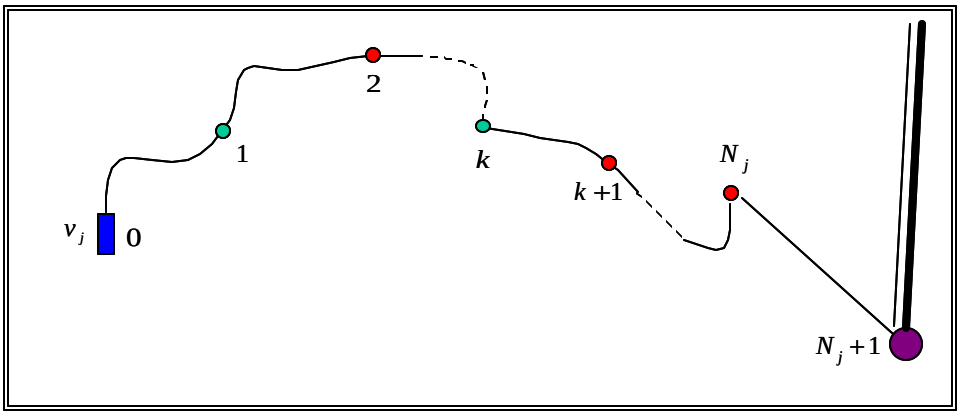

Figure 4.5 Graphical representation of a vehicle sequence of stops

At this point, it is important to mention that *all the variables to be defined and analytically formulated hereafter will depend explicitly on a predefined sequence of*

scheduled stops CS_j*, known at the current time t when vehicle j's position is* v_j. Thus, the variables defining the status and availability of vehicle j at a particular scheduled stop k will be functions of the path followed by such a vehicle between its current position and k. In addition, the variables defined over a segment $(k, k+1)$ will also depend on the predefined sequence CS_j.

As briefly mentioned in Section 4.2, the cardinality of the sequence CS_j will be $N_j + 4$ (from 0 to $N_j + 3$), however, in some cases, either stop $N_j + 1$ or $N_j + 2$ could be skipped depending upon the next hub to be visited by the vehicle while moving in its reroutable portion of the route. In this notation, the last three stops in CS_j are always

$$\begin{aligned} a_j(N_j+1) &= h(HO_j) \\ a_j(N_j+2) &= \underset{[x,y]}{Min}\left\{tSN(a_j(N_j),[x,y]) + tTN([x,y], a_j(N_j+3))\right\} \\ a_j(N_j+3) &= h(HD_j) \end{aligned} \qquad (4.3)$$

If at time t there is at least one passenger on board whose destination is different from that of vehicle j, vehicle j will be forced to stop at HO_j in order to drop those passengers allowing another vehicle to take them towards their final destination. In that case, stop $N_j + 2$ is skipped since vehicle travels straight from home to the visit-hub along the trunk stretch (HO_j, HD_j). Otherwise, that is, at time t all passengers on board are traveling towards the same destination as that of the vehicle, the vehicle doesn't stop at its home hub and enters the trunk network at the optimal point (loosely called stop $N_j + 2$) according to formula (4.3). Notice that $a_j(N_j + 2)$ is not a real stop. That point

is more a reference point, but since the point depends on the location of the last scheduled stop N_j, it becomes part of the sequence.

If the boarding and alighting times are added to the local network travel time expression, the vehicle j's base travel time between any two stops of its sequence CS_j, say between stops k and k+1, can be computed as follows

$$tS_j(k,k+1) = \begin{cases} tSN(a_j(k), a_j(k+1)) + E[tPHome(HO_j)] & \text{if } a_j(k+1) \text{ is a pick-up} \\ tSN(a_j(k), a_j(k+1)) + E[tDHome(HO_j)] & \text{if } a_j(k+1) \text{ is a delivery} \\ tSN(a_j(k), a_j(k+1)) & \text{otherwise} \end{cases} \tag{4.4}$$

The previous expression depends on the network travel time (assumed known at any time), including the boarding or alighting time, according to the operation required at stop k+1. If stop k+1 is just a reference point (for example, a hub location), such operation times are not considered. Notice that expression (4.4) depends on the hub area where the pick-up or delivery operation is taking place. In case of hub stops, the boarding and alighting times are treated separately, as shown ahead in this section.

In the incremental cost formulation, three different components are included. The operator cost is assumed to depend only on the time the vehicle has been circulating, while user cost can be split into two pieces - waiting and travel time components.

The goal here is to formulate an incremental cost expression in order to compute the additional cost incurred by all the passengers scheduled and being served by vehicle j, due to a new pick-up insertion of group Z_l into the current vehicle j route.

4.3.1.1 Total waiting time cost calculation

First, let us compute a general expression for the waiting time cost, associated with a group of customers Z_r (could be the new request Z_l, or any group of customers already assigned to the vehicle but not picked-up yet) when vehicle j travels from stop k to stop k+1 of its sequence. The service request time is $tW0_r$. The concept of time windows is incorporated here in order to take care of the user "impatience" generated by an exaggerated waiting time at the pick-up location. In this formulation, the time window constraint enters as a soft constraint, penalizing with respect to the amount of time by which the pick-up of a customer can deviate from a desired pick-up interval. This is shown in Figure 4.6.

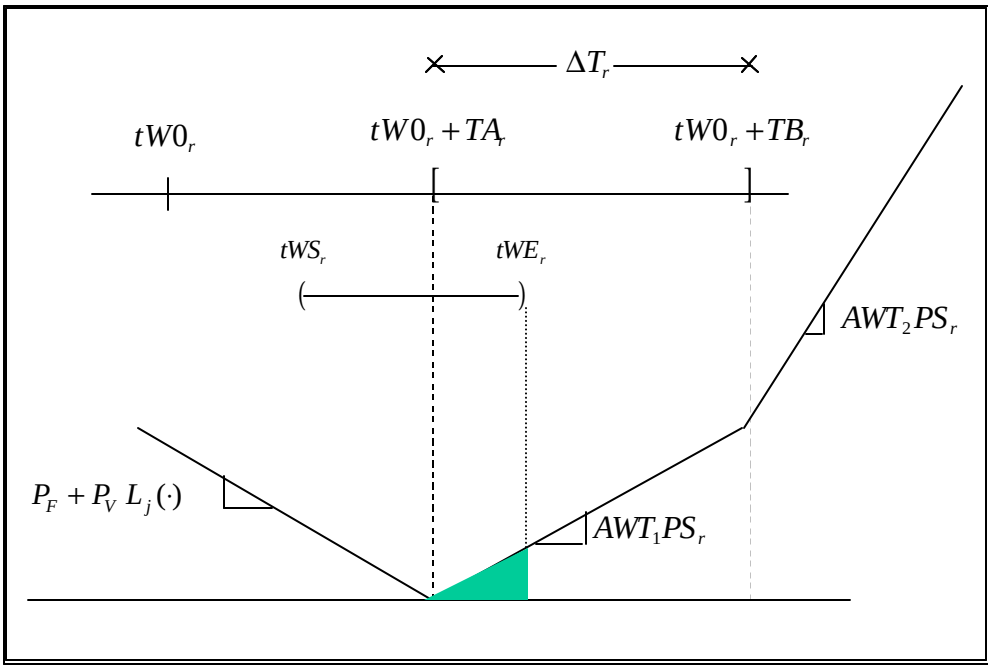

Figure 4.6 Waiting time cost component (total for all passengers on board and waiting at the stop)

In other words, $tW0_r$ is the time at which group Z_r starts waiting for service. TA_r, TB_r are defined respectively as lower and upper bounds of the desired pick-up interval measured with respect to $tW0_r$. For simplicity, only two different values of waiting time perceived by the customer are considered, depending on the interval where the service time falls. In addition, if the vehicle arrives early to pick-up these customers, say before the lower bound limit, it is assumed that passengers will not be ready for being picked-up, which will yields a penalty cost depending on the current vehicle load as shown in the figure. Thus, during interval $[tW0_r + TA_r, tW0_r + TB_r)$ AWT_1 will be the value of waiting time experienced by group of customers Z_r, while during $[tW0_r + TB_r, \infty)$, a greater value AWT_2 is assumed, both in $[US\$/\min]$. The upper bound of the interval defines the point from which customers really start getting impatient.

The purpose of this formulation is to compute the waiting time cost $CW(r)$, experienced by a group of individuals Z_r over the time interval $[t, t + \Delta t]$. In this context, that is to assume that vehicle j leaves stop k at t and arrives at next scheduled stop k+1 at $t + \Delta t$. First of all, let us define two auxiliary variables in order to simplify the notation.

$$tWS_r = t - tW0_r \tag{4.5}$$

$$tWE_r = tWS_r + \Delta t = t + \Delta t - tW0_r \tag{4.6}$$

From Figure 4.6, five possible cases can be distinguished depending on when both limits tWS_r and tWE_r fall along the time dimension. Notice also that $tWS_r < tWE_r$ and $tWE_r - tWS_r = \Delta t$. Analytically,

Case1: $(tWE_r < TA_r)$

$$CW(r) = [TA_r - tWE_r]\{P_F + P_V L_j(z_j^{-1}(r)\}$$

Case 2: $(tWS_r < TA_r) \wedge (TA_r \le TWE_r < TB_r)$

$$CW(r) = [tWE_r - TA_r]\, PS_r\, AWT_r$$

Case 3: $(tWS_r < TA_r) \wedge (tWE_r \ge TB_r)$

$$CW(r) = \{(TB_r - TA_r)AWT_1 + [tWE_r - TB_r]AWT_2\}PS_r$$

$$= [\Delta T_r\, AWT_1 + (tWE_r - TB_r)AWT_2]\, PS_r$$

Case 4: $(TA_r \le TWS_r < TB_r) \wedge (TWE_r < TB_r)$

$$CW(r) = [tWE_r - tWS_r]\, PS_r\, AWT_1 = \Delta t\, PS_r\, AWT_1$$

Case 5: $(TA_r \le TWS_r < TB_r) \wedge (TWE_r \ge TB_r)$

$$CW(r) = \{(TB_r - TWS_r)AWT_1 + [tWE_r - TB_r]AWT_2\}PS_r$$

Case 6: $(TWS_r \ge TB_r)$

$$CW(r) = [tWE_r - tWS_r]PS_r\, AWT_2 = \Delta t\, PS_r\, AWT_2$$

where P_F and P_V represent a fixed and variable penalty for arriving early at the pick-up point. Notice that in Case 1 the cost affects the vehicle and the passengers on board when vehicle reaches stop associated to group r. $L_j(z_j^{-1}(r))$ is the vehicle load after leaving stop $z_j^{-1}(r)$. The superscript –1 denotes the inverse function of $z_j(\cdot)$ defined in Section 4.2. In addition, $\Delta T_r = TB_r - TA_r$ and PS_r denotes the number of passengers comprising group Z_r.

Generalizing all cases in one expression, we get:

$$CW(r) = \boldsymbol{j}_r \left[\boldsymbol{q}_{Ar} TA_r + \boldsymbol{q}_{Br} TB_r + \boldsymbol{q}_{Sr} tWS_r + \boldsymbol{q}_{Er} tWE_r \right] \tag{4.7}$$

where

Case 1: $\boldsymbol{j}_r = 1$

$$\boldsymbol{q}_{Ar} = -\boldsymbol{q}_{Er} = P_F + P_V E[L_j(z_j^{-1}(r))]$$

$$\boldsymbol{q}_{Br} = \boldsymbol{q}_{Sr} = 0$$

Case 2: $\boldsymbol{j}_r = PS_r$

$$\boldsymbol{q}_{Er} = -\boldsymbol{q}_{Ar} = AWT_1$$

$$\boldsymbol{q}_{Br} = \boldsymbol{q}_{Sr} = 0$$

Case 3: $\boldsymbol{j}_r = PS_r$

$$\boldsymbol{q}_{Ar} = -AWT_1; \boldsymbol{q}_{Br} = AWT_1 - AWT_2; \boldsymbol{q}_{Sr} = 0; \boldsymbol{q}_{Er} = AWT_2$$

Case 4: $\boldsymbol{j}_r = PS_r$

$$\boldsymbol{q}_{Ar} = 0; \boldsymbol{q}_{Br} = 0; \boldsymbol{q}_{Sr} = -AWT_1; \boldsymbol{q}_{Er} = AWT_1$$

Case 5: $\boldsymbol{j}_r = PS_r$

$$\boldsymbol{q}_{Ar} = 0_1; \boldsymbol{q}_{Br} = AWT_1 - AWT_2; \boldsymbol{q}_{Sr} = -AWT_1; \boldsymbol{q}_{Er} = AWT_2$$

Case 6: $\boldsymbol{j}_r = PS_r$

$$\boldsymbol{q}_{Ar} = 0; \boldsymbol{q}_{Br} = 0; \boldsymbol{q}_{Sr} = -AWT_2; \boldsymbol{q}_{Er} = AWT_2$$

4.3.1.2 Passenger group conditions for cost calculations

Recalling the original problem of estimating the incremental cost of inserting new group of customers Z_l into vehicle j's original sequence CS_j, let us first define some useful conditions regarding all known users that are somewhat related to vehicle j at decision time t, when vehicle j reaches position v_j. The subsequent cost calculations depend on the passenger group conditions defined here. Each condition is represented by a (0,1) variable according to the following expressions:

- On-board passengers condition ($CONDB_j$): will be equal to one if and only if, all passengers on board with their destination at a neighboring hub, are traveling towards the same hub as the vehicle's visit hub, that is

$$\begin{aligned} &CONDB_j = 1 \quad iff \quad HZBD_j(m) = HD_j \quad \forall m \in ZBP_j \\ &CONDB_j = 0 \quad \text{otherwise} \end{aligned} \tag{4.8}$$

- New group of passengers condition ($CONDN_j(l)$): is defined in the same way as condition (4.8), but referred to all new customers that belong to Z_l with destination at some neighboring hub (external trips), analytically

$$\begin{aligned} &CONDN_j(l) = 1 \quad iff \quad HZD_l(k) = HD_j \quad \forall k \in ZE_l \\ &CONDN_j(l) = 0 \quad \text{otherwise} \end{aligned} \tag{4.9}$$

- Assigned customers condition ($CONDA_j$): follows the same definition, but now referred to all future pick-up requests scheduled to be served by vehicle j, thus,

$$\begin{aligned} &CONDA_j = 1 \quad iff \quad HZD_r(k) = HD_j \quad \forall k \in Z_r, \forall Z_r \subset ZA_j(0) \\ &CONDA_j = 0 \quad \text{otherwise} \end{aligned} \tag{4.10}$$

- The fourth condition $CONDC_j$ is a logical combination of (4.8), (4.10) as follows

$$CONDC_j = CONDB_j \cdot CONDA_j \tag{4.11}$$

Basically, if this condition were true and there were no further insertion, the vehicle should enter the trunk network and reach HD_j directly, skipping the stop at HO_j. But, on the other hand, if the condition were false, independently of the insertion, the vehicle should deviate towards HO_j before reaching HD_j.

- The fifth condition $CONDC_j(l)$ is also a logical combination of (4.9) and (4.11). It can be interpreted in the same way as (4.11) but adding the new insertion group as part of the assigned customers set. That is,

$$CONDC_j(l) = CONDC_j \cdot CONDN_j(l) \tag{4.12}$$

- The sixth and last condition $CONDH_j(l)$ is also a logical combination of (4.9) and (4.11), and shows the case in which the vehicle is deviated towards the stop at HO_j exclusively due to the new insertion. That is,

$$CONDH_j(l) = CONDC_j \cdot \left(1 - CONDN_j(l)\right) \tag{4.13}$$

4.3.1.3 Incremental cost

Once all these conditions have been defined, let us formulate the incremental cost problem resulting from any feasible insertion of group Z_l into vehicle j route. Feasible in two senses: (1) the insertion has to observe the vehicle capacity constraint at any segment along the vehicle route and (2) the insertion has to be consistent with the precedence constraint, i.e. all internal deliveries associated to group Z_l have to be inserted after the pick-up stop at ao_l. These two feasibility conditions are easily checked when running the proposed insertion algorithm (see Section 4.7.1), therefore, in the formulation they will not explicitly appear.

The concept of incremental cost implies the definition of two possible scenarios. The non-insertion case, representing the case in which Z_l is not inserted into vehicle j's route, and the insertion case, where Z_l is scheduled to be inserted into vehicle j's route. The former is conditioned by the original sequence CS_j while the latter is defined by a new sequence $CS_j(l) = \{0,1,\ldots,N_j(l),\ldots,N_j(l)+3\}$.

Notice that $CS_j(l)$ represents any feasible insertion of Z_l into the original sequence CS_j. The objective of this section is to compute the incremental cost associated to such an arbitrary feasible insertion alternative with respect to the original cost associated to the fixed sequence CS_j. The algorithm of Section 4.7.1 is designed to choose the best insertion sequence (the one resulting in the minimum incremental cost) among all feasible insertions for a fixed vehicle j. The dispatching module then will compare the best insertion option for all available vehicles within the hub region of the origin request, assigning the service to the vehicle with the minimum incremental cost. In order to make the comparison more efficient, the modeler could decide to restrict the insertion to a subset of vehicles instead, according to a proximity criterion.

To analytically express the cost function, its components need to be formulated in order. The components considered here are:

Waiting time component

With regard to the waiting time component, let us define the clock time as the absolute time at which vehicle j reaches an arbitrary stop k. Analytically, summing the travel times over the segments,

$$tCL_j(k) = t + \sum_{r=0}^{k-1} tS_j(r, r+1) \tag{4.14}$$

where t is the decision time defined above. Clearly, the previous expression can be generalized to any stop belonging to any sequence. Thus, for start and end times of vehicle travel over the segment,

$$\begin{aligned} tWS_r &= tCL_j(k) - tW0_r \\ tWE_r &= tWS_r + tS_j(k, k+1) \end{aligned} \tag{4.15}$$

From (4.15) all parameters in the cost function expression can be determined from the waiting time methodology summarized by expression (4.7).

Travel time component

With regard to the travel time component, the cost function expression comprises two major pieces. One part proportional to the vehicle load at stop k, and other accounting for the cumulative travel-time experienced by all passengers on board $TR_j(k)$.

The vehicle load has been split into pick-up load, internal and external deliveries load. Analytically, the load related components could be easily computed as follows

$$
\begin{aligned}
LP_j(k) &= LP_j(0) + \sum_{r=1}^{k} RS_j(r)\, ES_{z_j(r)} \\
LDI_j(k) &= LDI_j(0) - \sum_{r=1}^{k} \left\{ \left(1 - RS_j(r)\right) RD_j(r) + RS_j(r)\, IS_{z_j(r)} \right\} \\
LDE_j(k) &= LDE_j(0) - \sum_{r=1}^{k} \left\{ \left(1 - RD_j(r)\right)\left(1 - RS_j(r)\right) \right\}
\end{aligned}
\tag{4.16}
$$

where

$$
RS_j(k) = \begin{cases} 1 & \text{if the } k^{th} \text{ stop in vehicle } j \text{ sequence is a pick-up} \\ 0 & \text{if the } k^{th} \text{ stop of vehicle } j \text{ sequence is a delivery} \end{cases}
$$

and

$$
RD_j(k) = \begin{cases} 1 & \text{if the } k^{th} \text{ stop in vehicle } j \text{ sequence is an internal delivery} \\ 0 & \text{if the } k^{th} \text{ stop of vehicle } j \text{ sequence is an external delivery} \end{cases}
$$

Additionally, the assignment passenger set $ZA_j(k)$ can be updated from the original assigned pick-up requests as follows

$$
ZA_j(k) = ZA_j(0) \setminus \left\{ \bigcup_r Z_r : \exists n \le k \ s.t. \ z_j(n) = r \right\} \tag{4.17}
$$

In words, the assignment set of customers is shrunk every time a group of passengers is supposed to be picked-up by the vehicle according to sequence $CS_j(l)$.

With regard to the component associated with the cumulative travel time experienced by all passengers onboard vehicle j $TR_j(k)$, it turns out to be very straightforward from a conceptual standpoint. In fact, this variable only includes the cumulative travel-time of the passengers on this specific vehicle (basically, while they have been traveling on the last reroutable leg of their trip).

It was decided to incorporate this variable into the optimization procedure as a way to internalize the non-linear effects of travel time cost, and to relax the assumption of having a constant parameter for the value of travel time in the formulation. Moreover, this piece should somehow weigh the impatience of passengers in the vehicle as the trip get longer.

The analytical expression for $TR_j(k)$ is quite complex. The details of its computation can be found in Appendix to Chapter 4.

Insertion scenario

With regard to the insertion scenario cost, it must be noticed that the general expression will be associated to the insertion sequence $CS_j(l) = \{0,1,\ldots,N_j(l),\ldots,N_j(l)+3\}$. Remember that this sequence represents just one possible feasible insertion of group Z_l into the original vehicle sequence CS_j. Analytically, the cost associated to candidate insertion sequence $CS_j(l)$, can be computed by summing the costs on the sequence segments and adding the cost of the last segment to reach a hub. This computation yields

$$C_j(l) = \sum_{r=1}^{N_j(l)} C_j(r-1,r) + CR_j \tag{4.18}$$

where the cost associated with segment $(k,k+1)$ with $k+1 \le N_j(l)$, is computed as follows:

$$C_j(k,k+1) = tS_j(k,k+1)\left[OC + \left\{ LP_j(k) + LDI_j(k) \right\} ATT_1 + LDE_j(k)\, ATT_2 \right] + \\ + TR_j(k)\, ACT + \sum_{Z_r \subset ZA_j(k)} \boldsymbol{j}_r [\boldsymbol{q}_{Ar} T_{Ar} + \boldsymbol{q}_{Br} TB_r + \boldsymbol{q}_{Sr} tWS_r + \boldsymbol{q}_{Er} tWE_r] \quad (4.19)$$

Expression (4.19) includes both user and operator cost. The former considers those passengers known in the system at time t and somehow related to vehicle j hypothetical schedule $CS_j(l)$. The last summation computes the waiting time cost experienced by all scheduled pick-up groups still waiting at their origin location when vehicle j leaves stop k, according to expression (4.7).

The remaining part of the cost CR_j of expression (4.19) accounts for the transition between the reroutable and the non-reroutable portions of the vehicle route, and requires a careful look. Let us focus on the ultimate goal of this section, which is to estimate the additional cost of inserting a group of customers into a predefined vehicle route. Therefore, the first consideration to take into account is to compare different sequences (insertion versus non-insertion cases) reaching the same destination hub point. A second consideration to be taken into account is the fact that ultimately the insertion has to be assigned to the vehicle experiencing the minimum cost due to such an insertion. Thus the cost comparison is made across vehicles. Taking that fact into account, the remaining cost term has to measure the additional cost of taking all customers in ZE_l towards their final destination hub, in order to make this comparison consistent.

Let us first define the individual access cost $AC_j(i)$ as the cost of taking a single passenger onboard vehicle j to hub terminal i. Analytically,

$$AC_j(i) = \begin{cases} \left\{ \left[\dfrac{HT(HO_j)}{2} + tTN(ah(HO_j), ah(i)) \right\} ATT_1 + \right. & \\ + E[WTHub(HO_j)] \cdot AWT_h & \text{if } i \neq HD_j \\ \left\{ HT(HO_j) + tTN(ah(HO_j), ah(HD_j)) \right\} ATT_1 & \text{otherwise} \end{cases} \quad (4.20)$$

where AWT_h is the assumed value of the waiting time when the customer is waiting at a hub station. In addition, $HT(h)$ represents the average vehicle' stop time at hub h, and it could be computed as follows

$$HT(h) = Max\left\{ \left(E[D(h)] \cdot E[tDHub(h)] + E[P(h)] \cdot E[tPHub(h)] \right), ST \right\} \quad (4.21)$$

In (4.21), ST denotes the minimum vehicle stop time at a hub. Expression (4.26) takes special care of situation where passengers have to transfer at the origin hub (case when $i \neq HD_j$), adding the extra cost incurred by the system in taking them to their final destination on a different vehicle.

Thus, CR_j is computed as follows

$$CR_j = \begin{cases} C_j(N_j(l), N_j(l)+1) + \sum_{r=IS_l+1}^{PS_l} AC_j(HZD_l(r)) & \text{if } CONDC_j = 0 \\ C_j(N_j(l), N_j(l)+1) + \sum_{r=1}^{L_j(N_j(l)+1)} AC_j(HZBD_j(r)) & \text{if } CONDH_j(l) = 1 \\ C_j(N_j(l), N_j(l)+3) & \text{otherwise} \end{cases} \quad (4.22)$$

The first term of the first and second expressions in (4.22) is computed as before in equation (4.19), but excluding the waiting time component. However, the third expression needs additional treatment since there is no physical segment in the reroutable portion from stop $N_j(l)$ to stop $N_j(l)+3$. Considering this fact, the second expression becomes

$$C_j(N_j(l), N_j(l)+3) = tC_j(N_j(l), N_j(l)+3)\left[OC + LP_j(N_j(l))\, ATT_1\right] + \\ + TR_j(N_j(l))\, ACT \tag{4.23}$$

where

$$tC_j(N_j(l), N_j(l)+3) = tS_j(N_j(l), N_j(l)+2) + tTN(a_j(N_j(l)+2), a_j(N_j(l)+3)) \tag{4.24}$$

The $N_j(l)+2$ stop denotes the vehicle entrance point into the trunk line, from the position of the last scheduled stop $N_j(l)$. This last travel time expression is a combination of travel time computation in both route portions. Notice finally that the only remaining part of the vehicle load is the one that is going towards a neighboring hub (pick-up load).

Non-insertion scenario

The cost associated with the non-insertion scenario $C0_j$ is computed in the same way, but referred to the original scheduled vehicle sequence CS_j. In short,

$$C0_j = \sum_{r=1}^{N_j} C0_j(r-1, r) + CR0_j \tag{4.25}$$

The remaining portion is much simpler in this case than equation (4.22). As mentioned above, this last expression is needed when computing the incremental cost of inserting group l in order to compare consistent scenarios (for both the insertion and no insertion cases). Analytically,

$$CR0_j = \begin{cases} C0_j(N_j, N_j + 1) & \text{if } CONDC_j = 0 \\ C0_j(N_j, N_j + 3) & \text{otherwise} \end{cases} \tag{4.26}$$

By observing expressions (4.22) and (4.26), one can see that under each insertion condition, the comparison between both scenarios is consistent. On the one hand, all passengers on board at the decision time are taken to the same point in both cases. On the other hand, in expression (4.22) the cost of taken each member of group Z_l to their final destination hub is accounted.

For example, in the second case of expression (4.22), originally the vehicle is scheduled to access the trunk network without stopping at its home hub. However, after the insertion, the vehicle is required to stop first at its home hub before going towards its visit hub, since there is at least one passenger that belongs to group l who has his/her destination within a different hub area. For that case, the summation of equation (4.22) accounts for such an extra cost of taking everybody on the vehicle at home hub (stop

$N_j(l)+1)$ towards their final destination, some of them on the same vehicle, others transferring another vehicle according to their travel requirements.

Analytical computation of the incremental cost

Finally, the incremental cost expression for such an insertion can be computed by subtracting equation (4.25) from (4.18). That is

$$IC_j(l) = C_j(l) - C0_j \tag{4.27}$$

For a practical application of incremental cost based routing, it may be useful to add an extra factor $\chi_j(l)$ as a cost based on passenger type component of the insertion cost. Analytically,

$$\chi_j(l) = \begin{cases} \chi_1 & \text{if } CONDH_j(l) = 1 \\ \chi_2 & \text{else if } (1 - CONDC_j(l)) \cdot CONDN_j(l) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.28}$$

This term is purely heuristic and is added in order to prevent transfers from happening at the home hub, since vehicles then tend to pick up passengers whose destination is different from that of the vehicle. The operation is inefficient since the vehicle cannot access the trunk portion of its route at the optimal point. In addition, waiting time at hub increases because customers have a less chance of taking the right vehicle.

Thus, expression (4.27) becomes

$$IC_j(l) = C_j(l) - C0_j + \boldsymbol{x}_j(l) \tag{4.29}$$

Notice that expression (4.29) has to be evaluated for each vehicle and each feasible insertion choosing the minimum incremental cost among all feasible insertions and vehicles. An efficient manner of finding the best feasible insertion option for a specific vehicle is shown in Section 4.7.1.

Conceptually, the decision process can be seen as the process of comparing the incremental cost resulting from three different interrelated sources. A segment based cost and its components, then a total sequence cost and finally a component based on passenger type, for each insertion decision.

In the next section, expression (4.29) is generalized to the stochastic case introduced in Section 4.1.

4.3.2 Stochastic case

As introduced in Section 4.1, it has been claimed that the real travel time between two scheduled stops of an arbitrary vehicle route should change because of possible future pick-up insertions decided by the dispatching module dynamically over time. This premise is very reasonable when assuming dynamic and stochastic demand.

The objective of this section is to generalize expression (4.29) in order to capture the stochastic nature of vehicle travel time due to such unexpected rerouting and rescheduling decisions taken dynamically. The idea is to formulate the incremental cost function in the most realistic manner, that is, first compute the real cost over customers

and operator because of such an extra travel time delay, and second properly define the scenarios (insertion and non-insertion cases) and agents involved in the cost comparison process.

4.3.2.1 Expected vehicle travel time components

First, a general expression is proposed for the expected travel time experienced by vehicle j for traveling from k to k+1, both belonging to a scheduled sequence CS_j, as an additive linear function of the deterministic travel time defined in equation (4.4) and the expected number of insertions into the vehicle route as follows

$$E[tS_j(k,k+1)] = tS_j(k,k+1) + \Psi\left\{E[I_j(k,k+1)]\right\} \tag{4.30}$$

where

$$E[I_j(k,k+1)] = E[PI_j(k,k+1)] + E[DI_j(k,k+1)] \tag{4.31}$$

$E[PI_j(k,k+1)]$ and $E[DI_j(k,k+1)]$ represent the expected number of pick-up and non-scheduled delivery insertions respectively. These are non-scheduled, i.e. they come from the expected accumulated pick-up insertions unknown to the modeler at decision time t. Moreover, Ψ is a coefficient to be calibrated, representing the additional time caused by one extra insertion into the original route. The additive form of (4.31) could be modified depending on special the conditions of each specific case. In some scenarios, it could be more appropriate to define a multiplicative functional form for such an expression; however in principle an additive behavior seems reasonable.

In Chapter 5, a complete methodology for estimating the expected number of insertions is shown at length. In this section, it is assumed that the components in expressions (4.30) and (4.31) are known and measurable.

Before computing the corresponding incremental cost expressions, the expected vehicle load is split and analytically formulated, based on recursive functions from the current vehicle position where all conditions are deterministic.

4.3.2.2 Expected vehicle load components

Based on the components of the vehicle load in expression (4.16), let us compute the different components of the expected vehicle load on the segment $(k, k+1)$ recursively using the following formulas:

- For k =1:

$$\begin{aligned} E[LP_j(1)] &= LP_j(0) + RS_j(1)\, ES_{z_j(1)} + a_j(0,1) \\ E[LDI_j(1)] &= LDI_j(0) - \left(1 - RS_j(1)\right) RD_j(1) + RS_j(1)\, IS_{z_j(1)} + b_j(0,1) \\ E[LDE_j(1)] &= LDE_j(0) - \left(1 - RD_j(1)\right)\left(1 - RS_j(1)\right) \end{aligned} \tag{4.32}$$

- For k +1 given that conditions at k are known:

$$\begin{aligned} E[LP_j(k+1)] &= LP_j(k) + RS_j(k+1)\, ES_{z_j(k+1)} + a_j(k, k+1) \\ E[LDI_j(k+1)] &= LDI_j(k) - \left(1 - RS_j(k+1)\right) RD_j(k+1) + RS_j(k+1)\, IS_{z_j(k+1)} + \\ &\quad + b_j(k, k+1) \\ E[LDE_j(k+1)] &= LDE_j(k) - \left(1 - RD_j(k+1)\right)\left(1 - RS_j(k+1)\right) \end{aligned} \tag{4.33}$$

where

$$a_j(k, k+1) = E[ES]\, E[PI_j(k, k+1)] \tag{4.34}$$

$$b_j(k,k+1) = E[IS]E[PI_j(k,k+1)] - E[DI_j(k,k+1)] \tag{4.35}$$

and $RS_j(m), RD_j(m)$ having been already defined in Section 4.3.1. In equation (4.34) is assumed for simplicity that the pool sizes are uncorrelated to the number of pick-up points, an assumption that can be relaxed later.

Each term in expressions (4.32) and (4.33) is restricted to be non-negative in order to maintain physical consistency. As analytically shown in the next chapter, the expected load at any stop is bounded by zero from below and by the vehicle capacity from above to maintain consistency as well.

As defined in Section 4.2, parameters $ES_{z_j(k+1)}$ and $IS_{z_j(k+1)}$ denote the number of external and internal trips associated to group of customers $z_j(k+1)$ respectively, obviously valid only if $z_j(k+1)$ is a pick-up. In addition, expressions (4.32) and (4.33) are also functions of the expected number of external and internal trips per pick-up stop respectively, $E[ES], E[IS]$. In reality, these terms should take discrete values; however, in the current formulation it does not matter since they are used only to compute expected travel times, which are continuous.

Thus, the expected vehicle load at point $k+1$ can be calculated by summing the three terms in expression (4.33), that is

$$E[L_j(k+1)] = E[LP_j(k+1)] + E[LDI_j(k+1)] + LDE_j(k+1) \tag{4.36}$$

Notice that $LDE_j(m) \quad \forall m \in CS_j$, is not an expected value because the external deliveries are all wholly determined in advance at any stop. Equivalently, the expected vehicle load can be defined recursively as follows

$$E[L_j(k+1)] = E[L_j(k)] + RS_j(k+1)PS_{z_j(k+1)} - \left(1 - RS_j(k+1)\right) + c_j(k,k+1) \qquad (4.37)$$

where

$$c_j(k,k+1) = a_j(k,k+1) + b_j(k,k+1) = E[PS]E[PI_j(k,k+1)] - E[DI_j(k,k+1)] \quad (4.38)$$

and $E[PS] = E[IS] + E[ES]$.

The general expression for the expected load on any stretch in vehicle j's route computed in (4.37) can be easily decomposed in into two pieces. First, we can distinguish a known portion, where all its components can be calculated with certainty just by knowing the conditions at the current vehicle's position v_j. Second, we can visualize the other part, which is unknown, and a function of uncertain system factors at the current vehicle position v_j. Thus, equation (4.33) can be disaggregated and re-written as follows:

$$\begin{aligned}
&E[LP_j(k+1)] = LPK_j(k+1) + LPU_j(k+1) \quad \text{where} \\
&LPK_j(k+1) = LPK_j(k) + RS_j(k+1)\,ES_{z_j(k+1)} \quad \text{and} \quad LPU_j(k+1) = LPU_j(k) + a_j(k,k+1) \\
&E[LDI_j(k+1)] = LDIK_j(k+1) + LDIU_j(k+1) \quad \text{where} \\
&LDIK_j(k+1) = LDIK_j(k) - \left(1 - RS_j(k+1)\right)RD_j(k+1) + RS_j(k+1)\,IS_{z_j(k+1)} \quad \text{and} \\
&LDIU_j(k+1) = LDIU_j(k) + b_j(k,k+1) \\
&E[LDE_j(k+1)] = LDEK_j(k+1) = LDEK_j(k) - \left(1 - RD_j(k+1)\right)\left(1 - RS_j(k+1)\right)
\end{aligned} \qquad (4.39)$$

The first term of each expression in (4.39) represents the known part, while the last term includes all the assumed expected values. At location v_j, all the unknown expressions are null, since all the system conditions are supposed to be known at that point. Finally, with regard to the expected external deliveries, notice that there is no unknown component associated with them, since they are all known in advance at any physical point. Notice that the known portion of the load components matches equation (4.16).

All previous recursive formulas are valid for any stop $k+1 \leq N_j$. The remaining portion of the route needs a special treatment as in Section 4.3.1. The details of the expected load calculations for stops associated with the trunk portion of the vehicle route are presented in Appendix to Chapter 4.

4.3.2.3 Incremental cost formulation

Let us now write the incremental cost formulation in the same way it was done in the previous section, but adding the probabilistic component in both the expected travel time and expected load calculations. The incremental cost can be split again into a known and an unknown component.

The known part in both the insertion and non-insertion case (defined as $CK_j(l)$ and $CK0_j$ respectively) considers user and operator cost. For the user portion, only those passengers known in the system at time t and somehow related to vehicle j schedule are considered. The operator component is treated in the same way as in Section 4.3.1. The significant difference of this case with respect to the previous formulation in the last section relies on the travel time components, which in this case are stochastic even though the cost component is deterministic.

The unknown part in both scenarios as above (defined as $CU_j(l)$ and $CU0_j$ in the same way), accounts for the cost associated with those users not known at decision time *t*, but expected to be assigned to vehicle *j* and hence involved in the system optimization in a future time. Since, they are not known, there are some assumptions made in the calculations of these components as shown next in this section.

Insertion scenario

As in the deterministic case, the insertion scenario is related to a possible insertion sequence $CS_j(l)$ corresponding to a feasible insertion of group of customers Z_l into the original vehicle sequence CS_j. Analytically

$$C_j(l) = \sum_{r=1}^{N_j(l)} \left\{ CK_j(r-1,r) + CU_j(r-1,r) \right\} + CR_j \tag{4.40}$$

The cost associated with segment $(k,k+1)$ with $k+1 \le N_j(l)$, is computed as follows.

$$CK_j(k,k+1) = E[tS_j(k,k+1)] \left[OC + \left\{ LPK_j(k) + LDIK_j(k) \right\} ATT_1 + LDE_j(k)\, ATT_2 \right] + \\ + TRK_j(k)\, ACT + \sum_{Z_r \subset ZA_j^l(k)} \boldsymbol{j}_r [\boldsymbol{q}_{Ar} T_{Ar} + \boldsymbol{q}_{Br} TB_r + \boldsymbol{q}_{Sr} tWS_r + \boldsymbol{q}_{Er} tWE_r] \tag{4.41}$$

$$CU_j(k,k+1) = E[tS_j(k,k+1)] \left\{ LPU_j(k) + LDIU_j(k) + \frac{c_j(k,k+1)}{2} \right\} ATT_1 + \\ + TRU_j(k)\, ACT + CWU_j(k,k+1) \tag{4.42}$$

As defined in (4.38), $c_j(k,k+1)$ is the additional load from the segment due to pick-up and delivery insertions within the segment. Half of this is assumed as the average additional load in the segment. Thus, the term inside the bracket parenthesis in (4.4.2) represents the expected probabilistic load along segment $(k,k+1)$.

Notice that the operator cost only is included in expression (4.41). With regard to the waiting time components, in the specific case of equation (4.41) the only difference with respect to the deterministic case is the estimation of the limits tWS_r, tWE_r defined here as a function of both the expected clock time at which vehicle *j* reaches an arbitrary stop *k* and the expected vehicle travel time (in equation (4.30)). Analytically

$$E[tCL_j(k)] = t + \sum_{r=0}^{k-1} E[tS_j(r,r+1)] \tag{4.43}$$

Thus,

$$\begin{aligned} tWS_r &= E[tCL_j(k)] - tW0_r \\ tWE_r &= tWS_r + E[tS_j(k,k+1)] \end{aligned} \tag{4.44}$$

Regarding expression (4.42), the unknown waiting time component $CWU_j(k,k+1)$ is estimated assuming average values from the system. Analytically

$$CWU_j(k,k+1) = \begin{cases} E[WTHome(H0_j)] \cdot E[PS] \cdot E[PI_j(k,k+1)] \cdot AWT_1 & if\ E[WTHome(H0_j)] < \overline{\Delta T} \\ \{\overline{\Delta T} \cdot AWT_1 + (E[WTHome(H0_j)] - \overline{\Delta T})\} \cdot E[PS] \cdot E[PI_j(k,k+1)] & otherwise \end{cases}$$

where $\overline{\Delta T}$ is the average customer time window measured at the system. The other variables in the previous calculation have been previously defined. Note that this component starts becoming relevant when the observed customer waiting time becomes bigger than the observed patience time (acceptance time window), and the importance of this factor is assumed to be proportional to the expected number of pick-up insertions on the specific vehicle segment times the expected pick-up size, given the vehicle and system conditions.

The cumulative travel time of passengers on board $TR_j(k)$ has been split into a known and an unknown component, similar to all other variables. The details of these calculations are found in Appendix to Chapter 4.

The remaining portion CR_j associated to the transition to the trunk network is computed as before. However, in this case it is advisable to split it into known and unknown for a better understanding of the final process. The known part is almost the same as that of Section 4.3.1. In equations

$$CRK_j^l = \begin{cases} CK_j(N_j(l), N_j(l)+1) + \sum_{r=IS_l+1}^{PS_l} AC_j(HZD_l(r)) & \text{if } CONDC_j = 0 \\ CK_j(N_j(l), N_j(l)+1) + \sum_{r=1}^{L_j(N_j(l)+1)} AC_j(HZBD_j(r)) & \text{else if } CONDH_j(l) = 1 \\ CK_j(N_j(l), N_j(l)+3) & \text{otherwise} \end{cases} \tag{4.45}$$

where $AC_j(i)$ is computed using expression (4.20). The unknown remaining portion on the other hand is computed as follows

$$CRU_j^l = \begin{cases} CU_j(N_j(l), N_j(l)+1) & \text{if } CONDC_j = 0 \\ CU_j(N_j(l), N_j(l)+1) + LU_j(N_j(l)+1)\cdot ACU_j(l) & \text{else if } CONDH_j(l) = 1 \\ CU_j(N_j(l), N_j(l)+3) + (1-g)\cdot LU_j(N_j(l)+1)\cdot ACU_j(l) & \text{otherwise} \end{cases} \quad (4.46)$$

where g is a factor that measures the probability that the vehicle is deviated towards its home hub given that originally it was scheduled to go straight towards the visit hub by entering the trunk network at the optimal entrance location. An iterative method is needed in order to compute g as shown in Appendix to Chapter 4. $LU_j(k)$ denotes the unknown part of the load at stop k and $ACU_j(l)$ is the individual access cost for an unknown customer, which can be computed as follows

$$ACU_j(l) = \left(\left\{ \frac{HT(HO_j)}{2} + tTA_h(HO_j) \right\} ATT_1 + E[WTHub(HO_j)]\cdot AWT_h \right) \left(1 - PM_l(HD_j)\right) + \\ + \left(\left\{ HT(HO_j) + tTN(ah(HO_j), ah(HD_j)) \right\} ATT_1 \right) PM_l(HD_j) \quad (4.47)$$

where

$$PM_l(HD_j) = \text{Prob}\left(HZD_l(k) = HD_j, \forall k \in ZE_l \right) \quad (4.48)$$

represents the probability that an external trip destination hub matches the vehicle destination hub. In addition, $tTA_h(i)$ is defined as the average travel time from hub i to any of its neighboring hubs. Analytically

$$tTA_h(i) = \frac{\sum_{r=1}^{N_H} l_h(i,r)\cdot tTN(ah(i), ah(r))}{\sum_{r=1}^{N_H} l_h(i,r)} \quad (4.49)$$

where

$$\lambda_h(i,r) = \begin{cases} 1 & \text{if hub } r \text{ can be accessed from hub } i \text{ and } i \neq r \\ 0 & \text{otherwise} \end{cases}$$

In this case, unlike the deterministic formulation, both expected travel time calculations require special treatment, whether the vehicle has to proceed towards the home or the visit hub. The first and second expressions of (4.45) depend on $E[tS_j(N_j, N_j+1)]$, which could be computed directly using (4.30). However, in this case, the problem is not so simple. There is an operational rule that could add an additional source of stochasticity, which is defined as the "default path rule" (see Section 4.6.2.1 for details). Roughly, if the vehicle is not assigned to serve a new pick-up request, the dispatching module could eventually delay the decision of sending it to the trunk network in view of the current system conditions. In that case, the vehicle would be assigned to a "cell" or "group of cells" needing vehicle supply, but still within the reroutable portion. Each cell would have attached a "default path" where the vehicle would move towards, before accessing the trunk line. This extra delay in both $E[tS_j(N_j, N_j+1)]$ and $E[tS_j(N_j, N_j+3)]$ should also be treated as an extra unknown component. For simplicity in this formulation this extra factor is assumed not to be significant. This assumption can be relaxed later.

The third expression of (4.45), on the other hand, is a function of $E[tS_j(N_j, N_j+3)]$, which also is affected by the aforementioned rule. Moreover, there is an additional difference with the deterministic case. Here, even though the vehicle is scheduled to proceed straight towards its visit hub (without stopping at its home hub), there is still a probability that the vehicle may stop at its origin hub before going towards its visit hub, according to its possible future assignments.

All the details regarding the analytical treatment for computing both $E[tS_j(N_j, N_j+1)]$ and $E[tS_j(N_j, N_j+3)]$ are presented in Appendix to Chapter 4.

Apart from the expected travel time calculation, expressions (4.45) and (4.46) can be computed directly by applying (4.41) and (4.42). Known deliveries and known waiting time cost component are not part of this last stretch cost as well as the formulation of Section 4.3.1.

Non-insertion scenario

As in the deterministic case, the non-insertion scenario can be expressed in a more compact way than the insertion one, particularly regarding the remaining cost component. Analytically, the cost components described above can be computed as follows. For the non-insertion case

$$C0_j = \sum_{r=1}^{N_j} \left\{ CK_j(r-1, r) + CU_j(r-1, r) \right\} + CR0_j \tag{4.50}$$

In this case, the remaining portion is computed as follows

$$CR0_j = \begin{cases} CK_j(N_j, N_j+1) + CU_j(N_j, N_j+1) & \text{if } CONDC_j = 0 \\ CK_j(N_j, N_j+3) + CU_j(N_j, N_j+3) & \text{otherwise} \end{cases} \tag{4.51}$$

Analytical computation of the incremental cost

The ultimate incremental cost expression becomes

$$IC_j(l) = C_j(l) - C0_j + \xi_j(l) \tag{4.52}$$

To be precise about the segment cost calculations, it should be noted that only costs for potential additional pickups are considered. It may be argued that there is indeed a cost involved in *not* picking up a passenger. Technically this implies that for every vehicle segment under consideration, there are potential costs from not picking up all other passengers in the whole area (even outside the region of influence of the segment). Strictly speaking, expression (4.52) should also include an extra term in order to minimize the error coming from the lack of information and knowledge about future reassignments. An analytical expression for such a factor could become very complex and based upon a lot of assumptions. Therefore, so far it is considered negligible.

The heuristic term $\xi_j(l)$ is computed using (4.28) as before.

Remember that this expression has to be evaluated for each vehicle and each feasible insertion choosing the minimum incremental cost among all feasible insertions and vehicles.

In the next subsection, the algorithm utilized in order to chose the best feasible insertion of group Z_l according to the cost expressions discussed above is presented. The heuristics is very simple and efficient. It is also numerically shown that the algorithm performs well enough in most of cases.

4.3.3 Insertion heuristics: description and performance of the algorithm

The proposed insertion algorithm is a simplified version of a "branch-and-bound" process, and it is constructed starting from the original scheduled sequence of the studied vehicle. Thus, if the scheduled sequence of vehicle j is originally CS_j, the algorithm finds the best insertion of group of customers Z_l into that sequence, based on the cost criteria as in (4.52).

The idea of the algorithm is the following: first, insert the origin position of the group in all feasible places along sequence CS_j and compute the incremental cost of such an insertion. Second, for each origin insertion position ("branch"), find the best possible feasible insertion of all internal deliveries associated to Z_l in all feasible positions succeeding the origin insertion (to observe precedence condition). These insertions will not be taken into account ("bound") in cases where the computed cost of inserting just the origin spot results greater than the incremental cost of any previously computed entire sequence. This last condition assumes that the insertion of any internal delivery should somehow always increase the overall sequence insertion cost.

After finding the best insertion sequence, a swapping improvement process is implemented around the new insertions (of course observing precedence as well as feasibility conditions) in order to improve cases in which the new insertion could eventually favor a different order of two consecutive stops of the original sequence CS_j.

Analytically, let $ICo_j^{(s)}(l)$ be the incremental cost of inserting the origin spot of group Z_l after position s of sequence CS_j. In addition, $IC_j^{(s,m)}(l)$ will denote the

incremental cost of the m^{th} feasible combination of internal deliveries associated to the insertion of the origin spot after position *s*. Both terms are computed using either expression (4.29) or (4.52) depending the case.

Thus,

Step 0: Set *s*:= 1 and $L = \infty$.

Step 1: If $s > N_j$, **stop**. Otherwise, insert origin spot of Z_l at position *s*. Compute $ICo_j^{(s)}(l)$ and go to *Step 2*.

Step 2: If $s > 1$, compute $ICo_j^{(s)}(l)$ and check if $ICo_j^{(s)}(l) > IC_j^{(r,m)}(l)$ for some feasible combination of internal deliveries *m* associated to origin insertion at position $r, \forall r : 1,..., s-1$. If so, let $s := s+1$ and go back to *Step 1*. Otherwise, go to *Step 3*.

Step 3: Compute the cost of all feasible combinations of internal deliveries $IC_j^{(s,m)}(l)$. Find the minimum among them (say for $m = m^*$) and compare it with *L*. If $IC_j^{(s,m^*)}(l) < L$, set $L := IC_j^{(s,m^*)}(l)$. Go back to *Step 1*.

After termination, the sequence associated to the minimum value *L* will be the chosen sequence. A swapping improvement heuristics is run over the best insertion sequence previously found. The idea is to check for possible improvements (cost reductions) due to potential changes of the order of original sequence CS_j. The procedure consists in swapping stops belonging to CS_j next to any new insertion, whether it is the origin of the call or any of the internal deliveries associated, and accept the swap if, first the swap is feasible and also if there is an improvement due to such a

modification. The objective of this swapping procedure is to correct the constraint of taking CS_j as a base solution, and in some cases could considerably improve the performance of the heuristics. For clarity, consider the example in the following figure:

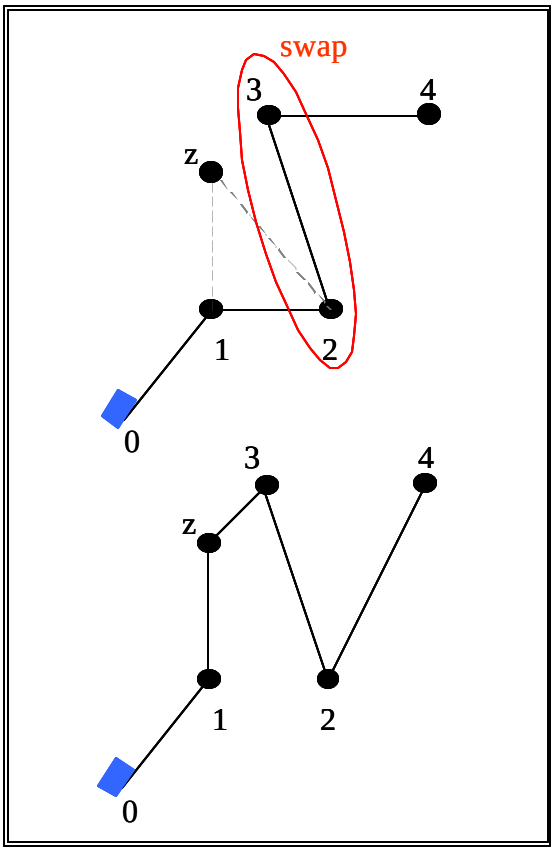

Figure 4.7 Insertion heuristics example: swapping improvement procedure

In the example, suppose that the insertion of spot *z* is part of the solution of the algorithm over sequence CS_j, represented by consecutive stops 1,2,3 and 4. For example, if the cost were directly proportional to the distance, it would be reasonable from the figure to swap stops 2 and 3, resulting in the final sequence 0-1-z-3-2-4, which should be better than 0-1-z-2-3-4 from the original algorithm.

The whole algorithm, including the swapping procedure performs very well. It is simple and very efficient considering the size of the analyzed problems (about 8 to 10 points in the worst case). In addition, the algorithm compares costs computed based on complete sequences, which is a relevant issue considering the cost structure introduced in Sections 4.3.1 and 4.3.2, in which the cost of succeeding stops depend upon the order of the preceding ones. Finally, the algorithm uses previous optimization solutions embedded in the definition of the original sequence CS_j, which is a helpful feature.

In the next table and figure, some interesting statistics are shown, comparing the performance as well as the computation time of this algorithm against the optimal solution obtained from the complete enumeration of alternatives. This is performed overa thousand different insertion cases for a random original sequence pattern.

Table 4.1 Insertion algorithm performance

Algorithm v/s benchmark comparison	***Value***	***(%)***
Total cases	1000	100
Perfect match	726	72.6
Average difference (w.r.t. not match cases)	0.04624	4.624
Average difference (w.r.t. total cases)	0.01267	1.267

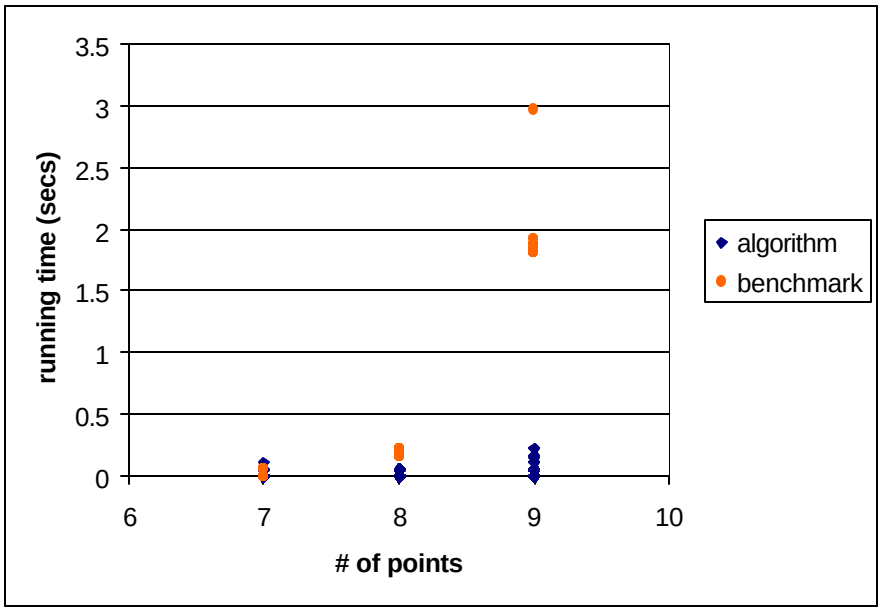

Figure 4.8 Computational time comparison

Notice from Figure 4.8 that computational time remains almost invariant regardless of the number of sequence points unlike the optimal solution method in which computational time go rapidly to infinity. Moreover, note from Table 4.1 that the percentage error in those cases in which the algorithm did not match exactly the optimal solution was always very low (about 4.624 % on average), showing that even though the algorithm solution is not optimal in 27.4 % of the cases, it is also very good and similar to the optimal one.

In the next section an additional improvement is discussed regarding the rescheduling of vehicles in real time accounting for potential benefits of reassigning already scheduled customers to different vehicles that have changed its route due to new unexpected insertions. This additional adjustment has not been implemented in the context of this dissertation, however in the next section some conceptual issues as well

as a simple solution algorithm are introduced and left for future improvements of this methodology.

4.4 Scheduling-routing rules: adjustment of vehicle schedules

In Section 4.3, a method to assign a new group of customers (previously defined as a pick-up request) to be served by a specific vehicle, operating within the reroutable portion of its route is mathematically formulated. Such a pick-up decision could eventually impact the current status of the system at a hub area level, since the chosen vehicle could be a good candidate to serve other pending pick-up requests assigned to different vehicles at a lower cost. If so, those vehicles could potentially change their schedules as well as their routes, resulting in a series of vehicle route adjustments until no further improvements occur.

In this section a method to adjust current vehicles routes and schedules is described in order to deal with this local imbalance, without adding excessive complexity to the formulations. The method is not numerically implemented as part of this dissertation, because it does not seem to result in a very significant impact on numerical results in the context of the *HCPPT* scheme. In a future publication, it will be improved, tested and implemented for cases in which the impact of rescheduling could become relevant.

The idea behind these adjustments is the following: if vehicle j's route changes due to a new insertion into its current route, it could be better (speaking in the economic sense as described in previous sections) to assign some already scheduled customers to vehicle j. If so, and if we decide to modify some other vehicle $k \neq j$ route, and

reassign, say one customer previously scheduled to vehicle k, to vehicle j, both vehicle j's and vehicle k's routes would change. The former because of the extra insertion, and the latter due to the pick-up exclusion from its original schedule. At this point, strictly speaking, possible improvements should be considered, considering a reschedule on vehicle k route as well, and so on. One nice feature of our system design is that we are able to restrict the analysis to the hub area associated to the reroutable portion of vehicle j's route only.

The incremental cost over the system for switching insertions from one route to another will be based on the same expressions formulated and computed in Section 4.3. In theory, one should analyze all possible improvements from any vehicle's route change within the studied region, however from a practical standpoint, the modifications should be restricted only to the major detected improvements due to assignment's changes (considering a maximum of one insertion change per vehicle's route). That should capture most of the significant improvements with respect to the previous reassignment solution. Hence, a simple adjustment procedure is proposed for being implemented in a further experiment, as follows:

Consider the adjustments resulting from a change in vehicle j's route due to the insertion of group of calls Z_l into its original sequence of scheduled stops. Z_l is now part of the set of scheduled calls to be served by vehicle j, $ZA_j(0)$. The original route of vehicle j has been modified, so it could be a good alternative to serve an already scheduled set of customers at a lower cost.

The hub region associated to vehicle j is $r = HO_j$, for some $r \in \{1,..,N_H\}$. Let us define the set of vehicles assigned to hub region r currently in the reroutable portion

of their route as $VR_r = \{1,2,..,MR_r\}$. Notice that this last set changes in real time according to the operation of the vehicles.

Then, the adjustment procedure for vehicle *j*, should be as follows:

Step 0: Set $m := j$, $iter := 1$ and $VR_r^{iter} = VR_r$.

Compute the adjustment extra cost of switching group of customers $Z_s \subset ZA_k(0)$ from assigned vehicle *k* to vehicle *j*. Analytically

$$AD_{mk}(s) = IC_m(s) - IB_k(s) \qquad \forall k \in \underset{k \neq m}{VR(r)}, \forall\ Z_s \subset ZA_k(0) \tag{4.53}$$

where

$$IC_m(s) = C_m(s) - C0_m + \boldsymbol{x}_m(s) \tag{4.54}$$

$$IB_k(s) = C0_k - C_k^e(s) - \boldsymbol{x}_k(s) \tag{4.55}$$

Expression (4.54) is just like (4.52), but referred to inserting group of customers Z_s into vehicle's *m* route, originally scheduled to vehicle *k*. Expression (4.55) is slightly different. Here, the idea is to compute the benefit of taking the group of customers Z_s out from vehicle's *k* route. The first term represents the cost incurred by the system in following the current vehicle *k's* sequence (including all stops associated to Z_s). The second term on the other hand, which is distinguished by a superscript *e*, represents the cost incurred by the system in following the optimal sequence after excluding Z_s from the original one. The last two terms in (4.55) are then computed accordingly.

Step 1: Find the best swapping

- If $AD_{mk}(s) \geq 0 \qquad \forall k \in \underset{k \neq m}{VR_r^{iter}}$, $\forall Z_k \subset ZA_k(0) \quad \Rightarrow \quad STOP$ (no improvement is possible from vehicle's *m* route)
- Else, find $\underset{k,s}{Min}\{AD_{mk}(s)\} \equiv AD_{mk^*}(s^*)$

  Assign Z_{s^*} to vehicle *m* and exclude it from original assigned vehicle k^*

  Set $iter := iter + 1$, $VR_r^{iter} = VR_r^{iter-1} - \{m\}$ and $m := k^*$

Go back to *Step 1*.

Notice that the maximum number of iterations of this adjustment procedure will be equal to the cardinality of the candidate vehicles VR_r, which is always not significant in terms of computational resources. In Chapter 8 (Further research and Conclusions) some insights are provided for possible numerical applications in order to quantify the real impact of these proposed adjustments in the original schedules under different routing patterns, considering the significance of the improvements as well as the disturbance on the scheduled routes every time the procedure is called.

Note also that in a more refined scheme, the adjustment criterion can also be utilized to assign a customer already scheduled to a certain vehicle to a different vehicle, which has just become free of all future tasks within the reroutable portion of its route. In that case, the manager has to check the economic benefit of assigning an additional pick-up request to it before it leaves the hub region moving towards the trunk network.

In the next section, a brief introduction to the initial *TSP* delivery tour construction is presented and discussed for deciding the initial sequence of stops when

vehicles pick-up passengers at hubs for delivering. Since this is only an initial solution, the *TSP* algorithm will be based upon the deterministic travel time between stops $tS_j(k,k+1)$. This assumption makes deliveries undistinguishable by the dispatching module when taking the distribution decision. Since the tour is built only for deliveries, neither precedence nor capacity constraints are required. The way in which deliveries are grouped for a specific vehicle is discussed in section 4.6.1.1. further in this chapter.

4.5 Scheduling-routing rules: initial delivery tour construction

In this section, the original delivery tour construction is treated. Every time a vehicle *j* arrives to a hub and picks up passengers for being delivered to their final destination within a hub region, the dispatching module has to take a decision and construct a sequence of stops $CS_j = \{0,1,\ldots,N_j,N_j+1\}$. Position zero represents the current vehicle position (in all cases it should be a hub location), N_j is the number of deliveries (bounded from above by the vehicle capacity L_{MAX}) and a last position N_j+1 that would be a desirable future location of the vehicle after finishing the distribution. So far, the origin hub location HO_j seems to be a good option to finish the tour. In addition, the tour is not closed, since the origin and final position of the vehicle will be in most cases different. The travel time of a segment from any location to the initial vehicle position is assumed to be zero (in order not to close the tour).

Two very simple algorithms were tested, the *nearest neighbor* and *GRASP* (Rosenkrantz, 1977). After repeated experiments, it was found that the difference in performance is not significant for problems of such a small size (7 to 8 points at most).

In order to check the quality of these algorithms, the exact *TSP* algorithm was coded (based on enumeration of all the alternatives). From the comparison, it was checked that both algorithms reach the optimal *TSP* solution in almost every case, according to the characteristics of the customer distribution, the symmetry of the hub region and the amount of point to be visited (7 to 8 points).

For the sake of simplicity, the *nearest neighbor* algorithm was chosen and implemented as part of the final simulations in this dissertation.

Notice that, in this case, since travel times will be computed from a simulated physical network (see Chapter 6 and 7 for details), in general the segment travel times are not symmetric (i.e., for a pair of stops k and l, $tS_j(k,l) \neq tS_j(l,k)$); therefore, when implementing the algorithm the order of the stops has to be considered in the cost comparison. Regardless of such a detail, the algorithm, in general terms, performs very well and runs very fast. It is also very simple to code and implement.

In the next section, some additional heuristic rules based the notion of optimality and common sense, are introduced in order to improve both the *HCPPT* system operation and performance.

4.6 Heuristics rules for improving system performance

The objective of this section is to show some additional rules designed to improve system performance - to distribute and balance location of vehicles within its area of influence according to the system requirements and manage terminal and general operations when vehicles are moving along the non-reroutable portion of their routes. In

Chapter 7 the impact of each of the proposed heuristics in system performance is tested and numerically quantified under different system designs and conditions.

Before presenting any rule, let us re-define the vehicle state in a simpler manner that will be also extended in Chapter 6 when explaining the final simulation issues. In Chapter 3 (Section 3.3), ten different vehicle states were defined in order to split the discrete event simulation into independent pieces. In what follows, only three vehicle states are needed for modeling the whole process. Thus,

- *State 1*: Vehicle is moving within the reroutable portion of its route, either assigned to serve a sequence of stops or moving towards its default path. It also includes cases where vehicles are moving back from its visit hub and they are assigned to leave the trunk corridor before getting to the home hub in order to start distributing passengers.
- *State 2:* Vehicle is either moving towards the trunk network or it is on the trunk corridor moving towards its visit hub.
- *State 3*: Vehicle is moving back from its visit hub along the trunk corridor but it has been assigned to stop at its home hub before starting dropping passengers at their destinations.

Graphically, recalling Figure 3.1:

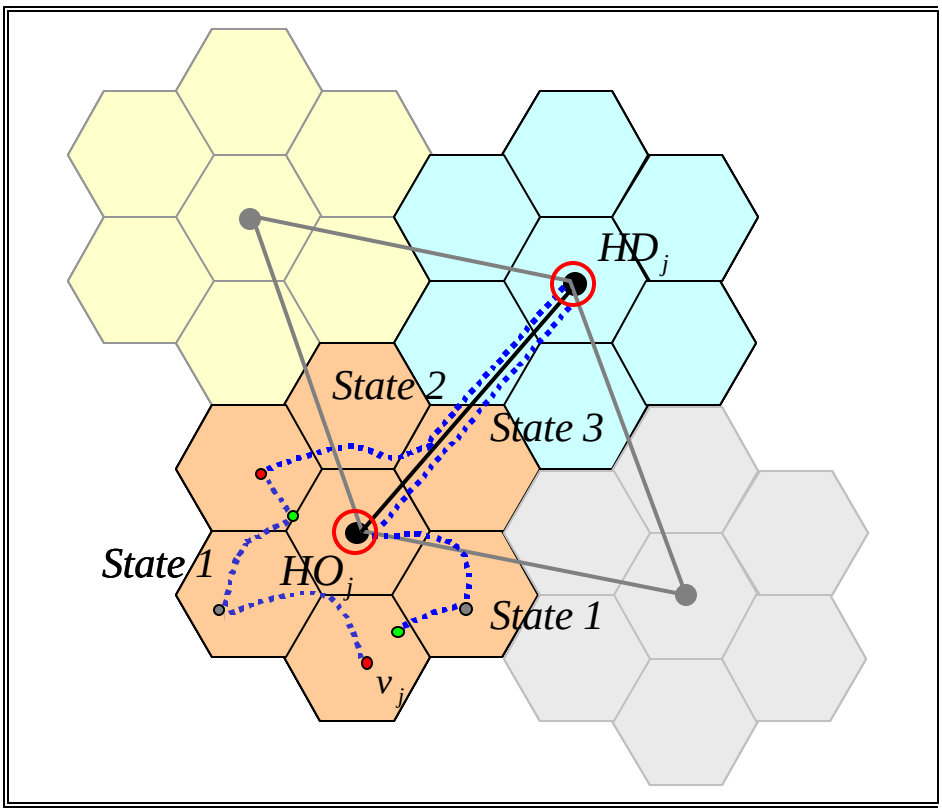

Figure 4.9 Vehicle schedule representation

The section has been divided into two subsections: the former describing rules for improving system operation at hub terminals and when vehicles are moving through the non-reroutable of their routes, and the latter describing some additional heuristics associated to the reroutable portion of vehicle routes complementing the cost expressions shown in Sections 4.3-4.5.

4.6.1 Non-reroutable portion of vehicle routes: heuristics for improving performance

4.6.1.1 Queuing strategy at hub terminals

In the case of delivery tours, grouping passengers at hubs on the basis of the proximity of their destinations may improve the efficiency of the system. This implies forming separate groups of passengers destined to different subareas within the hub zone.

Heuristics can be developed to form groups (loosely called queues) and this is expected to both minimize waiting time at the hub and reduce the length of initial *TSP* vehicle tours. Therefore, the strategy will take into account two different factors for giving pick-up priority to the passengers waiting at hub: passenger waiting time at hub and destination closeness among members of each queue as well as with respect to vehicle on-board passengers.

Queues conceptually will be lines formed at each hub. Each hub i will have $N_H(i)+1$ groups of lines, with $N_H(i)$ defined as the number of accessible hubs from hub i. Notice that passengers have not to be formed in line while waiting. The terminal operator just has to keep track of the minimum necessary information regarding each passenger when he (she) arrives at the hub. That is, arrival time and destination location (for the heuristic purposes, destination "cell" and "hub" will be enough).

First, let us show the strategy of forming these groups of queues while passengers get to the hub. Let us define for any given hub i the following two sets:

$$Q(i) = \{1,2,\ldots, NQ(i)\} \tag{4.56}$$

$$Q(i,s) = \{1,2,\ldots, NQ(i,s)\} \tag{4.57}$$

where $Q(i)$ denotes the group of queues of passengers formed at hub i with destination inside hub area i. $Q(i,s)$ on the other hand, denotes the group of queues of passengers waiting at hub i but with destination within hub area s. $NQ(i)$, $NQ(i,s)$ denote the number of lines that belong to $Q(i)$ and $Q(i,s)$ respectively.

The r^{th} queue belonging to either $Q(i)$ or $Q(i,s)$ defined as q_r^i (or q_r^{is} for the latter set) is composed by nq_r^i (nq_r^{is}) single passengers, that is

$$\begin{aligned} q_r^i &= \{1,2,\ldots, nq_r^i\} \\ q_r^{is} &= \{1,2,\ldots, nq_r^{is}\} \end{aligned} \tag{4.58}$$

where the m^{th} element of queue q_r^i (q_r^{is}) is wholly identified trough a one-to-one mapping queue-customer, similar to that defined in Section 4.2. Analytically

$$\forall m \in q_r^i \begin{cases} zq_r^i(m) = l \\ zqp_r^i(m) = n \end{cases} \tag{4.59}$$

$$\forall m \in q_r^{is} \begin{cases} zq_r^{is}(m) = l \\ zqp_r^{is}(m) = n \end{cases} \tag{4.60}$$

In words, the m^{th} member of the queue (either q_r^i or q_r^{is}) is the n^{th} customer associated to request group Z_l. Notice that at this stage of the trip, all members of group Z_l will be

distributed in different queues (and eventually in different hubs), depending on their final destination.

Let us describe the queue construction procedure based on a basic grouping criterion: closeness of the member destination. In a formal optimization problem, most probably a traditional clustering algorithm could have been useful to group customers according to their destination location. In the context of the *HCPPT*, a simpler clustering algorithm is developed, taking advantage of the hexagonal tiling used as well as the symmetry form of each hub area. Let us take a look to an arbitrary hub zone, composed by seven hexagonal cells numbered from 1 to 7 as in the next figure:

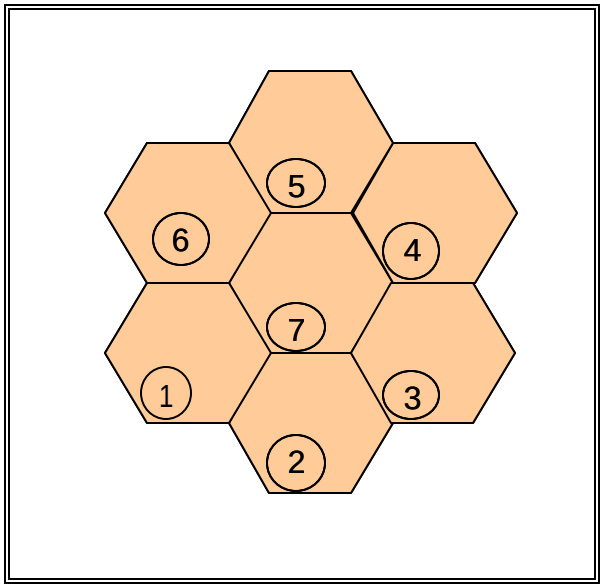

Figure 4.10 Hub area distribution of hexagonal cells

Obviously the central cell (7) is adjacent to every other cell within the area. All other cells are adjacent to three other cells, including the central one. With a total of seven cells, at most six clusters of adjacent cells (in groups of three cells) can be formed, say 1-2-7, 2-3-7, 3-4-7, 4-5-7, 5-6-7 and 1-6-7.

Therefore, six will be the maximum number of elements of a specific queue if the following constraint is imposed in the queue construction: every customer in the queue has to belong to a cell adjacent to any other queue member's cell.

Then, a very simple algorithm is used in order to group customers such that the previous constraint is fulfilled for any single queue. The algorithm is called every time a new customer arrives to a hub terminal, and is as follows:

Event: Customer $Z_l(n)$ arrives at stop hub i. If the customer destination $ad_l(n) \in HA(i)$ go to *Step 1*, else go to *Step 2*. $HA(i)$ denotes the area associated to hub i.

- *Step 1*: Find $q_r^i \in Q(i)$ such that the destination cell of the new customer $CZD_l(n)$ is adjacent to all other customers in that queue. If such a queue exists and $nq_r^i < L_{MAX}$, add $Z_l(n)$ to q_r^i. Otherwise, add a new queue in $Q(i)$ and put $Z_l(n)$ on it. Stop.
- *Step 2*: Let s be the destination hub of $Z_l(n)$. Find $q_r^{is} \in Q(i,s)$ such that the destination cell of the new customer $CZD_l(n)$ is adjacent to all other customers in that queue. If such a queue exists and $nq_r^{is} < L_{MAX}$, add $Z_l(n)$ to q_r^{is}. Otherwise, add a new queue in $Q(i,s)$ and put $Z_l(n)$ on it. Stop.

This algorithm ensures that each single queue contains only customers traveling to an area comprised by at most three adjacent cells. This distribution of customers is a nice way to somehow reduce the length of the initial distribution tours. The additional constraint regarding the capacity of the vehicles is for practical purposes since each

vehicle can accommodate at most L_{MAX} passengers, therefore queues longer than L_{MAX} are not really needed.

In what follows, the complete assignment procedure is described, based upon the described queue structure. Let us concentrate on the following scheme, regarding all passengers whose destination is within $HA(i)$. By focusing on neighboring hubs i and s, the analysis will be concentrated on those vehicles whose home hub is i and visit hub is s. In notation, vehicle $j \in VO_i$ means that $HO_j = i$, and similarly vehicle $j \in VD_s$ means that $HD_j = s$.

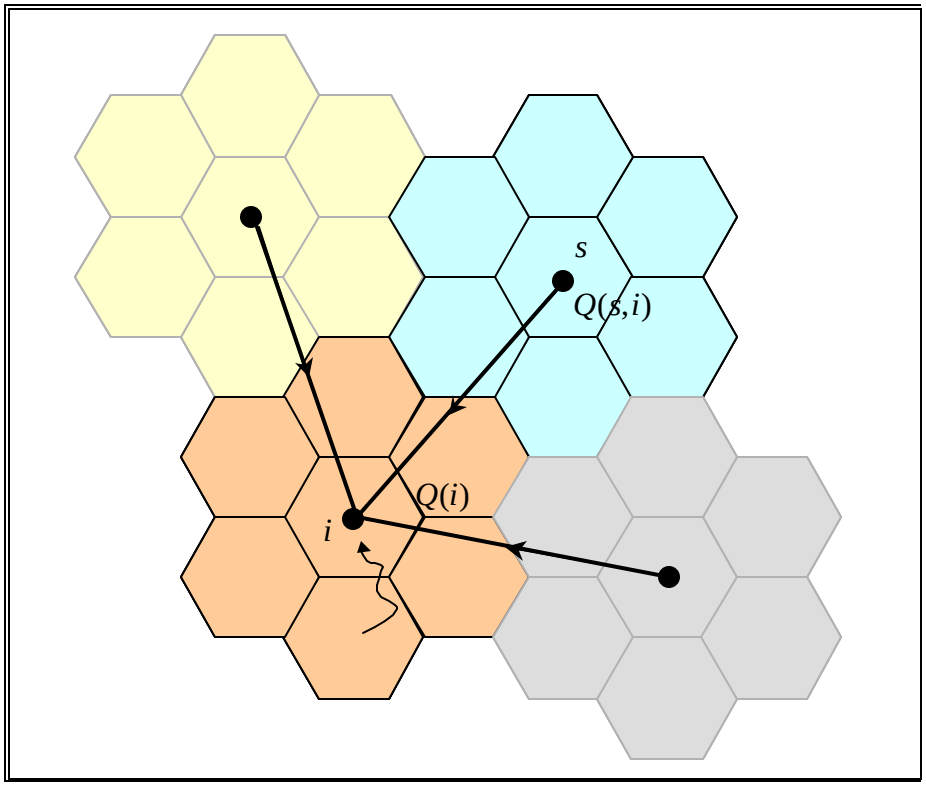

Figure 4.11 Terminal passenger assignment representation

Moreover, let us define the size of the assigned load at current vehicle j position as

$$LA_j = L_j(0) + NZA_j(0) \tag{4.61}$$

Notice that at this stage of the vehicle trip, all variables are deterministic.

The analysis will be performed for assigning passengers belonging to either $Q(i)$ or $Q(s,i)$ since they are reroutable inside $HA(i)$. All other pair of neighboring hubs will be treated in the same manner. Vehicle j for example, will have two potential stopping hubs $HO_j = i$ and $HD_j = s$, when moving on the way back from the trunk network to its reroutable portion, after dropping all passengers on board picked up inside $HA(i)$ and going to a neighboring hub area (external trips).

Before showing the final vehicle assignment algorithm, let us define two fundamental variables measuring the performance of any queue assignment according to the criteria and the queue structures stated above, focusing only on $Q(i)$ and $Q(s,i)$.

$FHA_{si}^{j}(m)$: is a variable designed to measure the attractiveness of the assignment of vehicle j to pick-up customers waiting at the m^{th} queue in set $Q(s,i)$. In this case, the position of vehicle j should effectively be at hub s. In addition, notice that depending on the vehicle assignment it could be possible to whether assign the whole queue to vehicle j or just a portion of it.

$FHAN_{i}^{j}(r)$: is a variable designed to measure the attractiveness of the assignment of vehicle j to pick-up customers waiting at the r^{th} queue in set $Q(i)$. In this case, the position of vehicle j could be anywhere on the way from hub s to hub i along the trunk corridor. The last comment for the previous variable regarding the capacity constraint is valid here too.

The greater the performance measure (either $FHA_{si}^{j}(m)$ or $FHAN_{i}^{j}(r)$), the higher the priority given to candidate vehicle j for picking up members of that queue.

Analytically, if $L_{MAX} - LA_j \geq nq_m^{si}$ (no capacity constraint, which means that the whole queue can be accommodated into the vehicle) then

$$FHA_{si}^{j}(m) = \Gamma_A \, ADJF_{si}^{j}(m) + \Gamma_T \, TH(q_m^{si}) \quad \forall m \in Q(s,i) \tag{4.62}$$

Expression (4.62) is a composite function, involving two different objectives (adjacency level and waiting time) weighted by arbitrary parameters Γ_A, Γ_T.

$ADJF_{si}^{j}(m)$ is the variable measuring the adjacency level between passengers assigned (and on board) vehicle j and the members of queue q_m^{si}. In equations:

$$\begin{aligned} ADJF_{si}^{j}(m) = \sum_{w \in q_m^{si}} \Bigg(& \sum_{v \in ZB_j(0)} \left\{ ADJ\left[CZD_{zq_m^{si}(w)}(zqp_m^{si}(w)), CZD_{zb_j(v)}(zbp_j(v)) \right] \right\} + \\ & + \sum_{Z_l(n) \subset ZA_j(0)} \left\{ ADJ\left[CZD_{zq_m^{si}(w)}(zqp_m^{si}(w)), CZD_l(n) \right] \right\} \Bigg) \end{aligned} \tag{4.63}$$

where $ADJ[c_1, c_2]$ measures an arbitrary level of adjacency between two cells c_1 and c_2, according to the modeler criteria. In the context of this dissertation, a simple weight is used for differentiating adjacent cells from non-adjacent ones. Thus,

$$ADJ[c_1, c_2] = \begin{cases} 2 & \text{if } c_1 = c_2 \\ 1 & \text{else if } c_1 \text{ is adjacent to } c_2 \text{ and } c_1 \neq c_2 \\ 0 & \text{otherwise} \end{cases} \quad (4.64)$$

$TH(q_m^{si})$ on the other hand, represents the total waiting time associated with queue q_m^{si}, and is computed as follows

$$TH(q_m^{si}) = \sum_{r=1}^{nq_m^{si}} \Delta tWh_{zq_m^{si}(r)}^{zqp_m^{si}(r)} \quad (4.65)$$

where ΔtWh_l^n denotes the time that the n^{th} member of group of customers Z_l has been waiting at its transfer hub, i.e. $\Delta tWh_l^n = t - tWh0_l^n$, where $tWh0_l^n$ represents the absolute time the customer starts waiting at his(her) transfer hub. Notice that expression (4.65) can be generalized to any queue.

The second variable $FHAN_i^j(r)$ is defined similarly. That is, for the capacity constraint case ($L_{MAX} - LA_j \geq nq_r^i$)

$$FHAN_i^j(r) = \Gamma_A \, ADJFN_i^j(r) + \Gamma_T \left\{ TH(q_r^i) + nq_r^i \, tTN(v_j, ah(i)) \right\} \quad \forall r \in Q(i) \quad (4.66)$$

The difference with expression (4.62) is the assumed extra waiting time of all members in the queue since vehicle j could be currently traveling from its visit hub to hub i. If the vehicle is at hub i the contribution will be zero. $ADJFN_i^j(r)$ is computed accordingly. That is

$$ADJFN_i^j(r) = \sum_{w \in q_r^i} \left(\sum_{v \in ZB_j(0)} \left\{ ADJ\left[CZD_{zq_r^i(w)}(zqp_r^i(w)), CZD_{zb_j(v)}(zbp_j(v)) \right] \right\} + \right.$$
$$\left. + \sum_{Z_l(n) \subset ZA_j(0)} \left\{ ADJ\left[CZD_{zq_r^i(w)}(zqp_r^i(w)), CZD_l(n) \right] \right\} \right) \tag{4.67}$$

A more complicated case is when the vehicle is not able to accommodate all members of the queue. In such a case the idea is to accommodate those passengers that maximize the appropriate performance measure. That is, for computing $FHA_{si}^j(m)$ in case of having $L_{MAX} - LA_j < nq_m^{si}$, sort q_m^{si} in a descendent order according to the following criterion:

$$wq_m^{si}(w)_j = \Gamma_A \left[\sum_{v \in ZB_j(0)} \left\{ ADJ\left[CZD_{zq_m^{si}(w)}(zqp_m^{si}(w)), CZD_{zb_j(v)}(zbp_j(v)) \right] \right\} + \right.$$
$$\left. + \sum_{Z_l(n) \subset ZA_j(0)} \left\{ ADJ\left[CZD_{zq_m^{si}(w)}(zqp_m^{si}(w)), CZD_l(n) \right] \right\} \right] + \Gamma_T \, \Delta tWh_{zq_m^{si}(w)}^{zqp_m^{si}(w)} \tag{4.68}$$

From this operation, a new queue structure $q_m^{si*} = \left\{ s(1), \ldots s(nq_m^{si}) \right\}$ is generated. Then, $FHA_{si}^j(m)$ can be computed as follows

$$FHA_{si}^j(m) = \sum_{p=1}^{L_{MAX} - LA_j} wq_m^{si}(s(p))_j \tag{4.69}$$

The other variable $FHAN_i^j(r)$ can be estimated in the same manner when $L_{MAX} - LA_j < nq_r^i$. Analytically, sort q_r^i in a descendent order according to the following criterion:

$$wq_r^i(w)_j = \Gamma_A \left[\sum_{v \in ZB_j(0)} \left\{ ADJ\left[CZD_{zq_r^i(w)}(zqp_r^i(w)), CZD_{zb_j(v)}(zbp_j(v)) \right] \right\} + \right.$$

$$\left. + \sum_{Z_l(n) \subset ZA_j(0)} \left\{ ADJ\left[CZD_{zq_r^i(w)}(zqp_r^i(w)), CZD_l(n) \right] \right\} \right] + \Gamma_T \left\{ \Delta tWh_{zq_r^i(w)}^{zqp_r^i(w)} + tTN(v_j, ah(i)) \right\} \tag{4.70}$$

Then, the sorted queue $q_r^{i*} = \left\{ s(1), \ldots s(nq_r^i) \right\}$ is generated. Finally, $FHAN_i^j(r)$ is computed as follows

$$FHAN_i^j(r) = \sum_{p=1}^{L_{MAX} - LA_j} wq_r^i(s(p))_j \tag{4.71}$$

Reverting to the scheme in Figure 4.11, there will be two major events that will require taking an assignment passenger-vehicle decision. Thus, every time a vehicle arrives to the visit hub (or to the home hub but changing from *State 3* to *1*), it should be assigned to pick-up potential customers waiting there for distribution (event 1), and every time a vehicle in *State 2* arrives to its home hub, it will leave passengers there to be picked-up by another vehicle (event 2). Event 2 is important because sometimes a vehicle in *State* 1 returning from its visit hub, still in the non-reroutable portion of its route but ready to start distributing passengers, turns out to be the best option to pick-up the new set of passengers dropped at its home hub. In such a case, the dispatching module could eventually take the decision of changing the state of that vehicle from 1 to 3, and force it to stay on the trunk corridor till its home hub for picking up these passengers before either start distributing or simply joining the surface network (see algorithm 4.3 next).

In both events, the assignment decision will be taken based upon the following algorithms:

Algorithm 4.1:

Event 1, case 1: Vehicle j stops at its home hub i. In this case, only passengers from set $Q(i)$ can be loaded into the vehicle.

Iter := 0

While ($LA_j < L_{MAX}$)

{

Iter := *Iter* + 1

Step 1: Find $r^* = \arg \underset{r \in Q(i)}{Max}\ FHAN_i^j(r)$

Step 2: Load $q_{r^*}^i$ (or the subset determined in (4.71)) to the vehicle

Step 3: Update $Q(i)$ and LA_j according to the previous load operation

}

Algorithm 4.2:

Event 1, case 2: Vehicle j stops at its visit hub s. In this case, passengers from both sets $Q(i)$ and $Q(s,i)$ can be loaded into (or assigned to) the vehicle.

Iter := 0

While ($LA_j < L_{MAX}$)

{

Iter := *Iter* + 1

Step 1: Compute $r^* = \arg \underset{r \in Q(i)}{Max}\ FHAN_i^j(r)$ and $m^* = \arg \underset{m \in Q(s,i)}{Max}\ FHA_{si}^j(m)$

Step 2: Find $u = \arg \underset{(r^*, m^*)}{Max} \left\{ FHAN_i^j(r^*), FHA_{si}^j(m^*) \right\}$

Step 3: Define $qA_j(Iter) = \begin{cases} q_u^i & if\ u = r^* \\ q_u^{si} & otherwise \end{cases}$

Step 4: Load (or assign) $qA_j(Iter)$ (or the subset determined in (4.69) or (4.71) depending on the case) to the vehicle

Step 5: Update $Q(i)$ (or $Q(s,i)$) and LA_j according to the previous load operation.

}

Notice that updating a set such as $Q(i)$ or $Q(s,i)$ implies deleting all customers that are either assigned to or loaded into any vehicle.

Algorithm 4.2 is more general than algorithm 4.1 since depending on the conditions the dispatching module could eventually decide to reserve space for a future assignment at the vehicle home hub instead of loading passengers directly at its visit hub while the vehicle is stopped there. In addition, if that happens, the vehicle is forced to stop at its home hub before starting distributing passengers (*State* 3).

Next, the other decision event is presented, when a vehicle delivers passengers at its home hub location the dispatching module could decide to assign them to be picked up by a vehicle returning from its visit hub, depending on the available vehicles moving along their assigned trunk corridors.

Algorithm 4.3:

Event 2, case 1: Vehicle $k \in VD_i$ delivers its passengers at hub *i*.

Step 0: At event time, define *VFA*: as the set of all vehicles $j \in VO_i$ in either *State* 3 or *State* 1 (the last ones should still be on the trunk corridor), satisfying $LA_j < L_{MAX}$.

Step1: Find $r^* = \arg \underset{r \in Q(i)}{Max} \left\{ TH(q_r^i) \right\}$

Step2: Assign $q_{r^*}^i$ to vehicle $j^* = \underset{j \in VFA}{Max} \left\{ FHAN_i^j(r^*) + PS1(j) \right\}$

Step3: Update vehicle j^* assignment and $Q(i)$

Step4: If either $Q(i) = \{\boldsymbol{f}\}$ or $LA_j = L_{MAX} \quad \forall j \in VFA \quad \Rightarrow$ **Stop**

Else go to step 1

Algorithm 4.3 complements the first two ones. The control of this event is a manner of deciding the exit location of vehicles from its non-reroutable portion when returning from its visit hub. In general, as discussed in Chapter 3, given the operating rules and the embedded design of *HCPPT*, it seems convenient to make vehicles stop at both hubs when returning from the visit hub due to the larger number of expected passengers waiting at their home hubs for being delivered within the same area. Note that this rule is consistent with the last heuristic term in expressions (4.29) and (4.52).

In addition, $PS1(j)$ is a penalty associated to rerouting vehicles already on *State* 1 (ready for starting delivery) to *State* 3, and it is assumed proportional to the assumed affected passengers on board as follows

$$PS1(j) \quad \begin{cases} \boldsymbol{q}_{P1} \cdot L_j(0) & \text{if } veh\ j \text{ in status } 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.72}$$

where $\boldsymbol{q}_{P1}$ is an arbitrary negative parameter.

The queuing strategy expounded in the current section is mostly heuristic and depends on clustering concepts, taking advantage of the symmetric structure of the cluster area and the natural division in cells as part of the original design. The real impact of applying such rules will be quantified as part of Chapter 7.

In the next section, a very simple rule associated to the decision of holding vehicles at hub stops when other vehicles are about to arrive is developed.

4.6.1.2 Holding vehicles at hub stops

This rule is very simple. In some cases, it will be reasonable to hold vehicles at hub stops, waiting for other vehicles that are coming to the same hub in order to transfer passengers. This procedure in most cases should improve the productivity values as well as reduce the passenger waiting times at hubs.

Let us consider a vehicle $j \in VO_i$ and $j \in VD_s$. The vehicle is stopped at visit hub s and the dispatching module considers the possibility of holding the vehicle at the stop for a while, if the following conditions are all satisfied:

- $NQ(i) + NQ(s,i) < MINAQ$
- $(1 - \mathrm{Prh}(s))MO_2(s) + MD_2(s) > MINVH$

where $MINAQ$ and $MINVH$ are suitable fixed parameters. In addition,

$\mathrm{Prh}(s)$: proportion of vehicles transferring at visit hub with home hub *s*.

$MO_p(s)$: number of vehicles that belong to VO_s in state *p*.

$MD_p(s)$: number of vehicles that belong to VD_s in state *p*.

On the other hand, if vehicle j is stopped at home hub *i*, the dispatching module could decide to stop the vehicle there for certain time, delaying the passenger assignment decision, if all the following conditions are satisfied:

- $NQ(i) < MINAQ$
- $MD_2(i) > MINVH$
- $LA_j < MINAP$

where *MINAP* is also an arbitrary parameter decided by the modeler. The holding rule is purely heuristics and it has to be evaluated via simulation as shown in Chapter 7.

4.6.2 Reroutable portion of vehicle routes: heuristics for improving performance

4.6.2.1 Default vehicle repositioning rule

As discussed in 4.3, sometimes it is advisable to maintain a certain balance between reroutable and non-reroutable vehicles at any time.

Roughly, if the vehicle is not assigned to serve any new pick-up request, the dispatching module could eventually delay the decision of sending it to the trunk network in view of the current system conditions. The particular decision of whether to send or not send a vehicle to the trunk network is discussed in Section 4.6.2.2. In this section, it is assumed that the decision of keeping the vehicle within its reroutable portion has been taken. The point here is where to send it.

The idea is to assign the vehicle to a "cell" or "group of cells" with low temporal vehicle supply, but still within the reroutable portion. Each cell would have attached a "default path" where the vehicle would move towards, before accessing the trunk line.

As mentioned in Section 4.3, this extra delay in both $E[tS_j(N_j, N_j+1)]$ and $E[tS_j(N_j, N_j+3)]$ is considered negligible so far.

At current time t we have to decide where to send vehicle j within hub area i. For each cell c that belongs to hub area i, the available cell capacity is defined as follows:

$$CAPC(c,i) = \sum_{u \in VO_i} D_u(c)\, SPK_u(0,1) \tag{4.73}$$

where

$$SPK_u(0,1) = L_{MAX} - \left(L_u(0) + [LRES_u(1)]^+\right) \tag{4.74}$$

denotes the known available space on vehicle u on its current segment, as a function of the vehicle load and the positive part of the reserved space for future scheduled insertions (see analytical definition in Appendix to Chapter 4).

In addition,

$$D_u(c) = \begin{cases} 1 & \text{if vehicle } u \text{ is within cell } c \text{ at the current time} \\ 0 & \text{otherwise} \end{cases}$$

Therefore, if $m_{PD}(c)$ represents the passenger demand per time unit within cell c (measured from the system in real time), then let us define the indicator demand-capacity for cell c that belongs to hub i, as follows

$$ICD(c,i) = \frac{m_{PD}(c)}{CAPC(c,i)} \quad \left[\frac{pax}{seats - TU}\right] \tag{4.75}$$

TU denotes time units. Finally, the rule is to send vehicle j to cell c^* belonging to hub i such that $c^* = \arg \underset{c \in Hub\ i}{Max}\{ICD(c,i)\}$. This rule is based on a clustering concept but taking advantage of the symmetry of hub areas and on how hub areas have been split into homogeneously distributed cells. The idea is to dynamically assign vehicles to different cells depending on the actual distribution of demand and supply on time and space.

In the next subsection, an additional rule is proposed for sending vehicles to the trunk network after being on the default path, determined by expression (4.75), for certain time.

4.6.2.2 Transition from reroutable to non-reroutable portion rule

As introduced above, the objective of this rule is to decide when to change vehicle state from 1 to 2 according to the states defined above in Section 4.6. This rule will be applied to all vehicles on their reroutable portion, without further stop assignments within the service area.

Let us first define the temporal load factor associated to hub *i* as follows

$$LFH(i) = \frac{DHD(i) + E_{NP} \cdot \sum_{s \in NH(i)} MO_2(s)}{L_{MAX} \cdot MONR(i)} \tag{4.76}$$

where $NH(i)$ denotes the set of hub i neighboring hubs and E_{NP} is the expected number of passengers per vehicles transferring at any single hub.

In addition,

$$DHD(i) = \sum_{s \in NH(i)} NQ(s,i) + NQ(i) \tag{4.77}$$

and

$$MONR(i) = MO_2(i) + MO_3(i) \tag{4.78}$$

Expression (4.77) accounts for the number of people waiting at any hub for being distributed within zone *i*. Expression (4.78) on the other hand, represents the number of vehicles on the non-reroutable portion of their trip with origin at hub *i*. All these variables are defined at time *t* and eventually can change over time.

The decision of sending a particular vehicle *j* to its trunk line will be based mainly on two variables: the hub load factor $LFH(i)$ and the cumulative time associated to external trips on board $TR_j^E(0)$. In this case, internal trips are not considered when taken this decision because they have to be delivered at their final destinations before joining the trunk corridor regardless of their current state. The strict definition of $TR_j^E(0)$ is provided in Appendix to Chapter 4.

In this section, two alternative rules are defined in order to decide whether vehicle *j* change status and start moving towards the trunk corridor or vehicle *j* stays for a while waiting for a new pick-up to come up (and it moves according to 4.6.2.1). The two criteria are as follows:

Rule 1:

If $TR_j^E(0) > TLIM$: change vehicle state from 1 to 2.

Otherwise check the following

If $LFH(i) > MINLF$: change vehicle state from 1 to 2.

Otherwise, send the vehicle to default path according to rule 4.6.2.1.

Rule 2:

If $TR_j^E(0) > TLIM$: change vehicle state from 1 to 2.

Otherwise check the following

If $L_j(0) = 0$ or $VRR(i) > CVRR$

If $LFH(i) > MINLF$: change vehicle state from 1 to 2.

Otherwise, send the vehicle to default path according to rule 4.6.2.1.

In case of Rule 2, $VRR(i)$ represents the rate between vehicles assigned to hub i within the reroutable portion of their route in State 1, versus those in States 2 or 3, at any time. *TLIM*, *MINLF* and *CVRR* are modeling parameters. Note that the cumulative travel time of people on board has higher priority than the load factor (vehicle distribution) issue. In addition, note that this rules are purely heuristic, and they are simply a more elaborate version of the C_{MAX} rule of the feasibility study in Chapter 3.

Rule 2 is stricter than Rule 1. In case of Rule 2, the load factor condition is checked only if temporarily vehicles in State 1 are too many compared to vehicles in other States (that is, when the rate between $MO_1(i)$ and $\left(MO_2(i) + MO_3(i)\right)$ is greater than certain threshold *CVRR*). Vehicles without passengers are checked anyway at this point.

The manner in which this rule is coded in the simulation scheme is explained in Chapter 6. Once again, the performance of this distribution decision has to be evaluated via simulation as shown in Chapter 7.

4.7 Real network *HCPPT* routing algorithms

The objective of this section is to briefly discuss some technological and practical issues regarding the feasibility of the implementation of a system such as *HCPPT* in the real world. In fact, there are some important aspects to be mentioned in this respect:

- As mentioned in Chapter 3, and considering the necessity of information in real time at any time for running all rules and processes discussed in this chapter, it is clear that the implementation of such a system relies on the use of updated information technology in order to feed all processes and inputs needed by the routing and scheduling algorithms. Thus, real time information regarding vehicle location (AVL), current state and occupancy, call attributes (group size, origin and destination accurate location, pick-up time request, etc.). In addition, a complex logistic system is requires at terminal level in order to properly manage and organize the queuing structure described in Section 4.6.1.1.

  In this sense, and thinking of implementing the *HCPPT* on the field, it is important to mention that the required technology is available nowadays, and currently is being tested in Corpus Christi, Texas, where the Regional Transportation Authority has been exploring ways to use technology to provide more responsive and efficient transit systems. Autonomous Dial-A-Ride (ADART) is a system currently being implemented there, developed in partnership with the Volpe Center. This system relies on a network of on-board computers that communicate with each other. In fact, all dispatching, routing and scheduling decisions are made by these

computers on board each vehicle. These on-board computers assign trips and plan routes optimally among themselves.

ADART technology encompasses a high level of automation, consolidating scheduling, fare collection, credit verification, and vehicle routing into a single automated system. There is no dispatcher, and the driver's only job is to obey instructions from the vehicle's computer. Consequently, an ADART fleet covers a large service area without any centralized supervision.

Unlike *HCPPT*, ADART has not been designed for a large-scale operation; however, the technology can be easily adapted to the *HCPPT* scheme.

- Notice that all cost expression, whether they are deterministic or stochastic, depend upon network travel time information, which also should be obtained somehow from the real network system. In addition, an efficient point-to-point shortest path algorithm has to be implemented in order to compute segment costs as well as to plan schedules in real time. In the simulation experiment (described in Chapter 6), a point- to-point shortest path algorithm is implemented based on microscopic simulated travel time data. In the reality, system information can be obtained from agency data, self reported data, loop data, and also from network status information collected by the transit vehicles while traveling trough the network system.

4.8 Final remarks

In this chapter, the central objective is to review all the rules and considerations regarding vehicle assignment and scheduling decisions taken in real time by a dispatching module, in the context of the *HCPPT* system introduced in Chapter 3.

In Chapter 6, the actual implementation of all the rules and algorithms into the simulation scheme is explained and presented at length. Some of the criteria are based on optimization fundaments while others are purely heuristics and have to be evaluated through simulation.

Most of the aforementioned rules depend on arbitrary parameters that could be calibrated or guessed based upon theoretical fundaments or simply common sense and intuition.

In other cases, some parameters have to be tested and analyzed via simulation.

This approach makes the system very flexible and controllable.

In the next chapter, an adaptive predictive control approach is developed in order to estimate and calibrate the expected number of insertions and expected vehicle travel times for a general pick-up and delivery problem scheme. This theory is needed in order to properly evaluate all rules previously described in Section 4.3.2.

5 ANALYTICAL MODELING OF STOCHASTIC REROUTING DELAYS FOR DYNAMIC MULTI-VEHICLE PICK-UP AND DELIVERY PROBLEMS.

5.1 Introduction

In the previous chapter, all heuristic rules for scheduling-routing *HCPPT* were presented from an analytical standpoint. As shown there, the design of the system geared towards a decomposed solution that achieve vehicle selection and route planning in a manner that ensures efficiency and quasi-optimality when applying sophisticated routing rules, terminal management heuristics and the appropriate operational schemes. However, there are some restrictions behind the design of *HCPPT*, which under certain unfavorable conditions could yield sub-optimal solutions.

All previous formulations incorporate both user (loosely called customer of the transit system) and operator cost components, implying an implicit trade-off between two agent objectives that conflict to some extent. In Chapter 7, several simulations will be carried out, quantifying the performance of the system under different weights for both cost components. There is no analytical way to measure the real impact on the transportation system of implementing specific policies obtained from weighting differently the cost function components. However, a detailed simulation of the real-world operation as in Chapter 7 will allow the modeler to study these different policies from a quantitative measure of the performance of the simulated system.

At this point, there is one assumption behind most of the scheduling-routing rules described in Sections 4.3-4.6, which has not been clarified yet. Basically, in the previous formulation, a fundamental premise is that there exists an analytical way to estimate the expected travel time of, say vehicle j, between two scheduled stops, assuming that all conditions and attributes of vehicle j are known by the dispatching module before it arrives at the upstream stop position of such a segment. Since the analysis is based on fulfilling the travel necessities of unknown customers appearing in real-time, it is assumed that most of the predefined routes and schedules should dynamically change as a new call enters the system (see the illustrative example of Section 4.1). Thus, one important issue to be highlighted is that, when computing an analytical expression for any kind of decision cost function (whether it affects the user or the operator), there is an additional source of stochasticity on the calculation of the expected travel time between two points on a transportation network, apart from the typical non-recurrent congestion of traffic affecting all vehicle travel times, which is the "*stochastic rerouting delay for dynamic vehicle routing (SRDDVR)*".

In other words, for all the already-scheduled users, the expected travel (or waiting) time that they will incur during their trip will be strongly affected by any future reassignment of the vehicle assigned to pick them up at their origin spot, or drop them at their destination in case they are already on the vehicle. The major idea is to introduce such estimation into the routing rules already described in Chapter 4 in order to incorporate a more realistic measure of travel (waiting) time experienced by the users as well as the operator into the decision cost formulation, which eventually could change

some of the dispatching module decisions, resulting in better solutions closer to the desired dynamic social optimal equilibrium.

As discussed in Chapter 2, there are not many publications in the specialized literature that solve "dynamic vehicle routing problems", especially in case of large-scale systems. In addition, none of them have considered the stochasticity in travel time from dynamic reassignment as part of the objective function formulation, considerably limiting the chance of reaching a good and realistic solution.

The objective of this chapter is to introduce an approach designed to properly estimate the expected travel time of a transit vehicle to any of its scheduled stops under stochastic demand, generated dynamically over time without any previous knowledge of these new requests by the dispatching module at the time of the call. It is important to point out that the general methodology computed in the next three sections of this chapter, generally apply to all dynamic real-time pick-up and delivery problems (*PD*). Therefore, first a general framework will be developed (Sections 5.2-5.6), and next (section 5.7) all necessary extensions of the methodology and additional notation will be presented. The focus in Section 5.7 is in the optimization schemes applied within the "reroutable" portion of *HCPPT*.

The methodology is based upon real-time update of probabilities, in order to predict the expected vehicle's travel time between two already scheduled stops. As will be discussed in Sections 5.3-5.4, conceptually this problem could be considered as a form of quasi-optimal adaptive predictive control. In the next section, notation adjustments with respect to the original definition section 4.2 are addressed. Next, in Section 5.3 some of the basic elements of adaptive and predictive control theory are

briefly discussed, then, the "*stochastic rerouting delay for dynamic vehicle routing*" approach is formulated as an adaptive predictive control problem in Section 5.4, and in Section 5.5, a methodology to solve the specific control problem in real time from a branched process approach (*BPA*) is proposed. In Section 5.6, the data collection as well as model calibration issues are addressed. Finally, in Section 5.7, some final remarks are pointed out and all necessary extensions for modeling *HCCPT* under the same scheme are described.

5.2 Notation adjustments

As pointed out above, the scheme presented in this chapter is in principle applicable to any dynamic vehicle routing scheme; however, the notation should be consistent with that of Chapter 4 for the analysis of *HCPPT* schemes. In fact, the notation specified in Section 4.2 remains the same, observing some small exceptions:

1) Regarding the definition of the vehicle's sequence of stops CS_j, in this case the sequence will be defined only for the "reroutable" portion of vehicle's j route (there is no other portion in the general pick-up and delivery formulation), discarding any connection with the trunk network stated in the definition of the *HCPPT* system. Analytically

   - CS_j: sequence of stops scheduled for vehicle j from the current position v_j (position 0 in CS_j) till the last stop scheduled N_j. In set notation $CS_j = \{0,1,2,\ldots,N_j\}$.

- Correspondence between stop sequence numeration and physical position (Cartesian coordinates) of any stop that belongs to CS_j:

$$a_j(s) = \left(x_j(s), y_j(s)\right) \qquad \forall s \in CS_j$$

thus, $a_j(0) = \left(x_j(0), y_j(0)\right) \equiv v_j = \left(xv_j, yv_j\right)$

2) In the general case, since there are no transfers, there is no difference between internal or external deliveries. In fact, all customers have to be delivered to their final destinations by the same vehicle that picked them up from their origin spot. Basically, for an arbitrary group of customers Z_l, $PS_l = IS_l$ and $ES_l = 0$, i.e. all trips are internal deliveries.

3) The notation associated to hub correspondence is removed. For example, $E[tPHome(i)]$ which represented the expected pick up time per passenger at home (Hub area $i \in H$), now is simply $E[tPHome]$.

5.3 Elements of adaptive predictive control theory

According to Sunan *et.al* (2001), predictive control, as the name implies, is a form of control, which incorporates the prediction of a system behavior into its formulation. The prediction serves to estimate the future values of a variable based on the available system information. The more representative the information is of the system, the better is the accuracy of the prediction. The estimate of the future system variables can then be used in the design of control laws to achieve a good control performance, which is usually to drive or maintain the output to a desired set-point. This class of control

methods, which incorporates information and assumptions pertaining to the future values of the system output, are generally referred to as predictive control.

Predictive control is usually considered when a better performance than that achievable by non-predictive control is required. This includes systems whose future behavior can be quite different from that perceived of the present one, such as time-delay systems, high-order systems and poorly damped and non-minimum phase systems.

An early predicted control method was suggested by Smith (1959). He introduced the idea of using a predictor to overcome the problem encountered in controlling a system with dead time.

There are many predictive controller formulations based on the same common ideas, amongst which can be included: Identification and Command (IDCOM), Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC), Internal Model Control (IMC), Predictor Based Self-Tuning Control, Extended Horizon Adaptive Control (EHAC), Extended Predictive Self Adaptive Control (EPSAC), Generalized Predictive Control (GPC), Multistep Multi-variable Adaptive Control (MUSMAR), Multipredictor Receding Horizon Adaptive Control (MURHAC), Predictive Functional Control (PFC), Unified Predictive Control (UPC), Con-strained Receding-horizon Predictive Control (CRHPC), Stabilizing I/O Receding Horizon Control (SIORHC) (Soeterboek, 1992; Betmead *et.al*, 1991; Camacho and Bordons, 1995; Richalet *et al*, 1978; Cutler and Ramaker, 1980; Clarke *et al*, 1987a, 1987b).

All these methods have certain features in common which distinguish them from other design philosophies - the solution of optimization problem at each time instant implemented by using receding horizon concept, the incorporation of plant output

predictions and the provision of a small number of design parameters connected to various degrees with the closed loop dynamics (Betmead *et al*, 1991).

With regard to the concept of adaptive control, as defined by Saridis (1977), an adaptive control system must provide continuous information about the present state of the plant that is to *identify* the process; it must compare present system performance to the desired or optimum performance and make a *decision* to adapt the system so as to tend toward optimum performance; and finally it must initiate a proper *modification* so as to drive the system towards the optimum. These three functions are inherent in an adaptive system.

As Kanjilal (1995) points out, an adaptive control scheme can be broadly considered as a combination of an on-line estimation method for the process parameters, and a controlled design procedure. There are two widely used configurations on adaptive controllers: (1) *Self-organizing adaptive controllers* recursively identify the process, and formulate the control strategy aiming at optimal performance. (2) *Model reference adaptive controllers* try to achieve a closed-loop system performance similar to that of a reference model by recursive adaptation of the controller parameters.

From the fusion of both schemes, the concept of Adaptive Predictive Control methods (APC) has been introduced over the last decade. Basically, if a good model is available *a priori* to describe the dynamic relationship between the system input and outputs, it may be used as the predictive model. However, in most cases it is difficult to obtain precise information about the system a priori. Moreover, even if it were available, the system may frequently vary its dynamics in its evolution over time. The purpose of adding an adaptive component to the predictive control system is to achieve, in an

uncertain and time-varying environment, the results that would otherwise be obtained by the predictive control system only if the system dynamics were known (Sunan *et al*, 2001).

Many works on APC methods with guaranteed stability and with impressive results obtained by using robust control design approaches published in nineties (Camacho and Bordons, 1995). Most predictive controllers presented in the literature are adaptive predictive controllers (GPC, EHAC, EPSAC, MUSMAR, MURHAC, CRHPC, SIORHC). GPC is the most popular APC method (Clarke *et al*, 1987a and 1987b).

The number of publications on design and applications of Non-linear APC is increased in nineties (Katende and Jutan, 1995). There are applications of APC used neural nets and Fuzzy Logic (Camacho and Bordons, 1995). Methods of Model Predictive Control are widely using in Business and Economics (Hanke and Reitsch, 1995). Conventional control is applicable only to well-structured problems. For problems which poorly understood and described only in natural language terms fuzzy logic can play a role either by quantifying imprecise natural language and/or by converting human experience to systematic but fuzzy if-then rules (Ho, 1999).

First, let us present a very general state space formulation of the model predictive control by borrowing some general ideas from Morari (1994).

The general structure is presented in figure 5.1. A modeler utilizes knowledge of the process (plant) inputs u and measurements y to arrive at a state estimate $\hat{x}$. Starting from the current state estimate $\hat{x}$, one can employ classic prediction algorithms to predict the behavior of the process outputs over some output horizon H_p when the manipulated inputs u are changed over some input horizon H_c (see Figure 5.2).

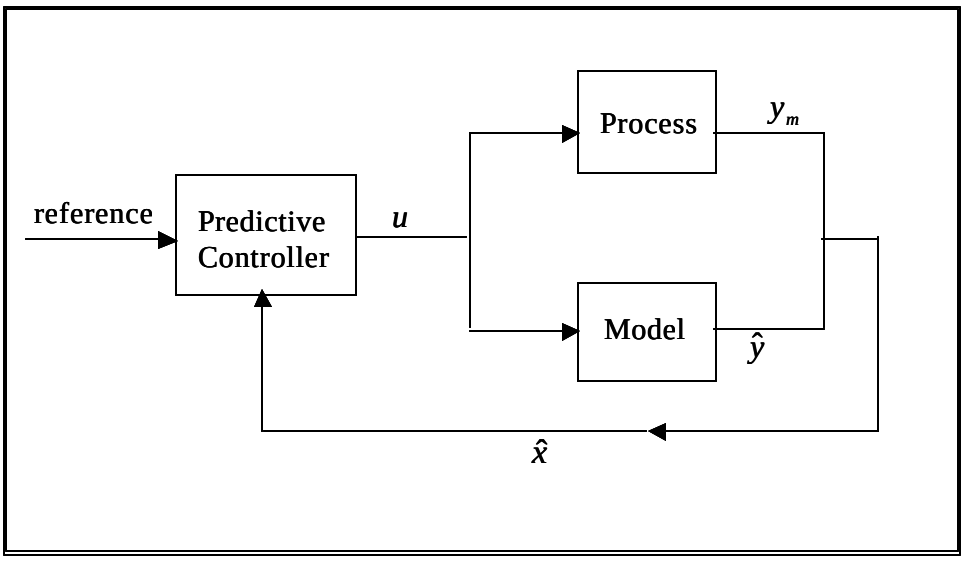

Figure 5.1 Overall block diagram of a predictive control system

The task of the predictive controller is to compute the present and future manipulated variable moves $u(s),\ldots,u(s+H_c-1)$ such that the predicted outputs follow the reference in a desirable manner. The predictive controller takes into account constraints on the inputs and outputs, which may be present.

Only $u(s)$, the first one of the sequence of optimal control moves is implemented on the real plant. At time s +1 another measurement $y_m(s+1)$ and another state estimate $\hat{x}(s+1)$ are obtained, the horizons are shifted forward by one step, and another optimization is carried out. The procedure results in a *moving horizon* or *receding horizon* strategy. A key feature of the technique is that the input and output horizons (H_c and H_p) are generally finite. The problem definition as presented allows one to treat with equal ease multivariate problems with an unequal number of inputs and outputs, non-minimum phase systems and systems subject to constraints.

Assume that the system is described by

$$x(s) = f(x(s-1), u(s-1), w(s-1)) = f_s(x(s-1), u(s-1)) + f_e(w(s-1)) \quad (5.1)$$

$$y_m(s) = g(x(s), v(s)) = g_s(x(s)) + g_e(v(s)) \quad (5.2)$$

where the customary nomenclature has been employed. The vector of manipulated variables is u, y_m is the vector of process measurements, w is the state disturbance and v the measurement noise. The disturbance and the noise could be of a deterministic or a stochastic nature. The theory of output prediction is summarized through the following steps:

$$\hat{x}(s/s-1) = f_s(x(s-1/s-1), u(s-1)) \quad (5.3)$$

$$\hat{y}(s/s-1) = g_s(\hat{x}(s/s-1)) \quad (5.4)$$

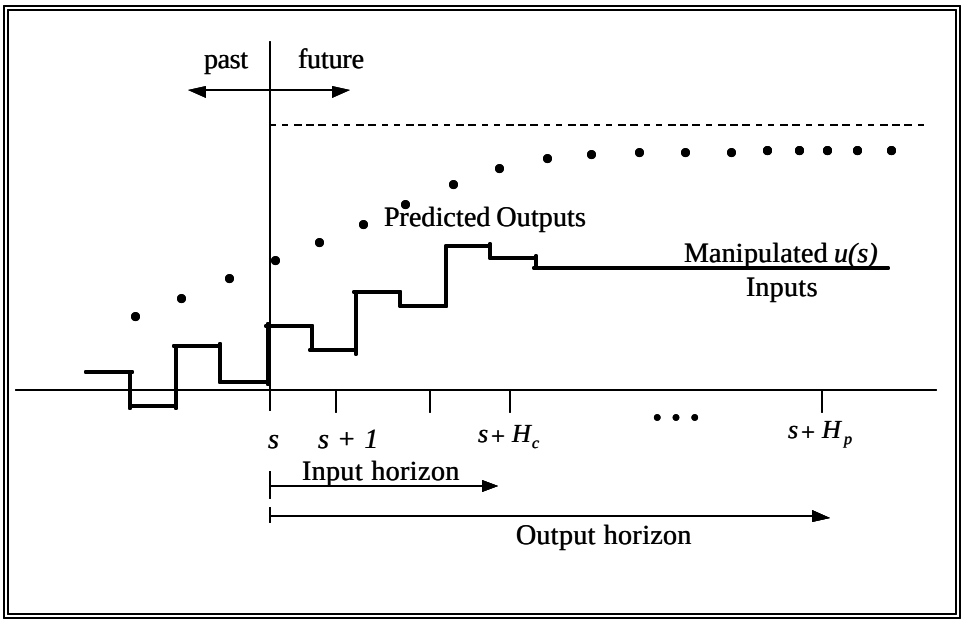

Figure 5.2 Definition of the optimization problem for model predictive control

Correction based on measurements:

$$x(s/s) = \Gamma(\hat{x}(s/s-1), [y_m(s) - \hat{y}(s/s-1)], \beta) = \hat{x}(s/s-1) + \beta\,(y_m(s) - \hat{y}(s/s-1)) \tag{5.5}$$

Prediction

$$\hat{x}(s+1/s) = f_s(x(s/s), u(s)) \tag{5.6}$$

$$\hat{y}(s+1/s) = g_s(\hat{x}(s+1/s)) \tag{5.7}$$

Prediction for more than one step ahead can be obtained by applying the prediction equations (5.6 and 5.7) recursively. The present and future control actions are found from the solution of the following optimization problem:

$$\underset{u(s),u(s+1),\ldots}{Min}\; J = J_1(\hat{x}(s+1/s), \hat{x}(s+2/s), \ldots, \hat{x}(s+H_p/s)) + J_2(u(s), u(s+1), \ldots, u(s+H_c-1)) \tag{5.8}$$

In this formulation, several assumptions have been made, only for the sake of clarity in the presentation of the formulation. For example, the fact of assuming that equations (5.1), (5.3) and (5.8) are separable is not straightforward. In addition, the second equality of equation (5.5) is a quite simple representation of the correction resulting from the measurement of the output, and it is not necessarily valid for all models.

In the same equation, notice that the parameter β (or set of parameters B) is (are) fixed. An adaptive predictive controller APC would be obtained by combining parameter identification and predictive control algorithms, as shown in Figure 5.3. The predictive model will change at each sampling instant *s*. In the adaptive system, the system model gives an estimation of the system output at instant *s* using model

parameters $\hat{\mathrm{B}}$ also estimated at instant *s*. In the predictive controller, the estimated model is used to formulate the predictive model at instant *s* and to derive the control law.

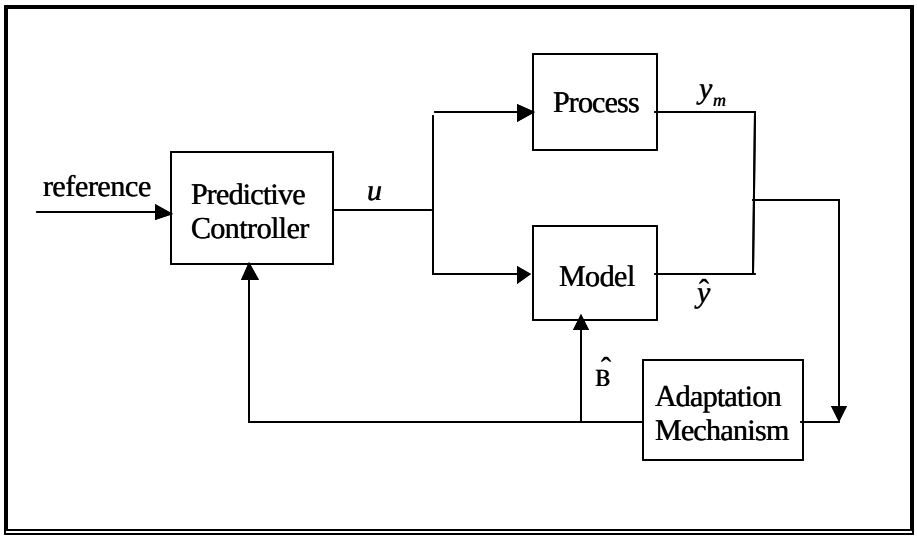

Figure 5.3 Overall block diagram of an adaptive predictive control system

In the previous formulation, the correction based on measurements step (equation 5.5) would be replaced by the following expression in order to obtain an APC scheme instead of just a predictive model:

$$x(s/s) = \Gamma(\hat{x}(s/s-1), [y_m(s) - \hat{y}(s/s-1)], \hat{\mathrm{B}}(s)) \tag{5.9}$$

where

$$\hat{\mathrm{B}}(s) = \begin{cases} \hat{\mathrm{B}}(s-1) & \text{if } y_m(s) \approx \hat{y}(s/s-1) \\ \hat{\mathrm{B}}(y_m(s), y_m(s-1), \ldots, y_m(s-H_e)) & \text{otherwise} \end{cases} \tag{5.10}$$

In words, the set of parameters should be adjusted (within the adaptation mechanism) whenever needed. Now, at time *s*, with an additional set of data available, the objective

is to determine the updated parameters $\hat{B}(s)$. H_e represents the number of sample periods backwards the modeler will consider representative for estimating the set of parameters $\hat{B}(s)$ in order to obtain a reasonably good predictive model Γ.

In the next section, the stochastic rerouting delay for dynamic vehicle routing *SRDDVR* problem will be formulated in terms of the schemes presented above, highlighting the similarities, differences and restrictions of designing this particular problem as an APC, or simply as a predictive control model. Next, in Section 5.5-5.6, a recursive family of algorithms for solving the *SRDDVR* problem in real time is proposed. Finally, in Section 5.7, the *SRDDVR* scheme will be adapted to the conditions of the pick-up decision rules taken within the "reroutable" portion of the vehicle's routes in the context of the *HCPPT* system.

5.4 Stochastic rerouting delay for dynamic vehicle routing: an adaptive predictive control approach

The proposed routing-scheduling rules presented in chapter 4 were designed to account for fundamental issues such as the dynamic nature of the system. Since calls are generated dynamically and the pick-up and delivery decisions are taken in real time based on system information at the decision time, decision rules are developed that depend on the *expected number of future insertions* into pre-established vehicle routes. Note that passenger assignments could change over time because of changes in system conditions. Hence, the proposed *SRDDVR* scheme essentially considers stochastic travel times to be expected for future assignments. These travel times can be calibrated online based on historical information regarding the performance of the system, just as in APC

systems. This capability of fine-tuning the system is what essentially brings out the quasi-optimal nature of the solutions.

Assume an influence area *A*, with a transit service network of length *D* in distance units [DU]. Moreover, a fleet of transit *VF* vehicles of size *NF* is currently in operation traveling within the area according to the routing rules already specified in chapter 4 for the "reroutable" portion of *HCPPT* vehicle' routes. The demand for service is unknown and is generated dynamically in real-time (assume a rate μ_{CD} in calls per time units [call/TU]). Routing and scheduling decisions have to be taken in real time, in order to accommodate such a demand into the available vehicles.

The predictive controller of Figure 5.3 is in this case represented by the dispatching module, who takes routing decisions in real time based on the information he has from the system (process) and the expected values for travel times and attributes of its vehicle fleet (model).

Time steps are not directly applicable in this particular case, since the predictive controller is taking routing decisions every time a call enters the system. In other words, the control mechanism is activated whenever a new request comes in. Hence, instead of defining a fixed time step, in this scheme it is necessary to formulate the problem in terms of "epochs" $e(s)$, where $e(s)$ will represent the time interval between requests $s-1$ and s.

The state of the system at instant s (i.e, when request s enters the system) will be determined by the attributes of each vehicle within the area *A* at that moment (say, $ATTR_j(k,k+1)$ for vehicle j, segment $(k,k+1)$). Vehicle attributes include all vehicle features when leaving the k^{th} stop of its sequence, these are clock time of arrival

$tCL_j(k)$, vehicle load $L_j(k)$ (that could be replaced by the available space for accommodating new requests on the stretch), the cumulative travel-time experienced by all passengers on board after leaving stop k, $TR_j(k)$, and a measure of the available space for potential passengers in the proximity of segment $(k, k+1)$, $WSP_j(k, k+1)$.

This last variable is introduced here and represents the system's available capacity (measured in vehicle seats) within the "catchment area" of segment $(k, k+1)$ when vehicle j is expected to cross such a route segment. The catchment area concept is discussed at length in Section 5.5 and Appendix to Chapter 5. The analytical expression for $WSP_j(k, k+1)$ is attached in Appendix to Chapter 5.

Moreover, the control $u(s)$ can be visualized as the effect of the dispatching module routing decisions, represented by the set of sequences assigned to every vehicle before $e(s)$. Analytically,

$$x(s) = \left\{ ATTR_j(k, k+1) \right\}\Big|_{e(s)} \qquad \forall\ j: 1, \ldots, NF \tag{5.11}$$

$$u(s) = \left\{ CS_j \right\}\Big|_{e(s)} \qquad \forall\ j: 1, \ldots, NF \tag{5.12}$$

Notice that, sequences CS_j remain fixed during the whole interval $e(s)$, since during that time the controller does not realize any action. With regard to the output of the system, it is definitely associated to vehicle ride times onto pre-specified segments. In fact, the vector of process measurements y_m will be computed measuring the observed segment travel times $tS_j(k, k+1)$, occurring during the time interval $e(s)$ for all vehicles and for all pair of stops $(k, k+1)$. These pairs of stops are adjacent stops in the

current sequence CS_j. Notice that, in order to measure the expected travel time, it is required the vehicle to arrive at stop k +1 sometime within time interval $e(s)$. Analytically

$$y_m(s) = \left\{ tS_j(k,k+1) \right\}\Big|_{\substack{tCL_j(k+1)\in e(s) \\ (k,k+1)\in CS_j\big|_{e(r)<e(s)}}} \tag{5.13}$$

where $tC_l(k+1)$ represents the clock time at which vehicle j leaves stop $k+1 \in CS_j$, and $CS_j\big|_{e(r)<e(s)}$ denotes a vehicle j sequence being defined for an epoch $e(r)$ previous to $e(s)$. Figure 5.4 shows an example of a certain control strategy applied by the dispatching module at the time request s enters the system, for a hypothetical case with *NF = 5*.

In what follows, $\hat{x}(s), \hat{y}(s)$ will represent the system predictions based on modeling parameters calibrated before epoch $e(s)$. In equations,

$$\begin{aligned} &\hat{x}(s) = \left\{ E[ATTR_j(k,k+1)] \right\}\Big|_{e(s)} \\ &\hat{x}(s) = \left\{ E[tCL_j(k)], E[L_j(k)], TR_j(k), WSP_j(k,k+1) \right\}\Big|_{e(s)} \\ &\qquad \forall\, j, \forall k, k+1 \in CS_j \end{aligned} \tag{5.14}$$

$$\hat{y}(s) = \left\{ E[tS_j(k,k+1] \right\}\Big|_{e(s)} \qquad \forall\, j, \forall k, k+1 \in CS_j \tag{5.15}$$

Expressions (5.14) and (5.15) show an estimation of the system state, comprising vehicle features and expected travel times over predefined segments, the latter as a function of the expected number of insertions. The modeling procedure embedded into the

computation of the expected vehicle travel time, will remain invariant for a period predefined by the modeler. In case of applying just a predictive approach, the parameters of the model would be considered always fixed independent of the observed measurements y_m.

Note also that the prediction horizon H_p is not fixed in this formulation, and it can be computed as the latest expected stop time among all vehicle' sequences, that is

$$H_p(s) = \underset{j:1,\ldots,NF}{Max} \left\{ E[tCL_j(N_j)\} \right|_{e(s)} \tag{5.16}$$

In an APC scheme, the major difference would be to add an adaptation mechanism via the calibration of certain parameters associated to the model procedure, whenever needed. In other words, after a period of evaluation of the performance of the model, it could be necessary to recalibrate some procedures and update certain observed expected values, in order to better reproduce the observed values measured in real time from the operation of the system. This test can be carried out every H_c epochs, with $H_c < H_p$ (receding horizon concept). See Figure 5.5 for a representation of the whole APC process for Stochastic Rerouting Delay.

Analytically, the expected travel time will depend on the expected number of non-scheduled insertions (see Section 5.5 for a detailed treatment of the modeling process), which also depends on the conditions of the system, the demand rate, and a set of parameters $\hat{B}(s)$ describing the probability of vehicle-call assignment given the corresponding state conditions.

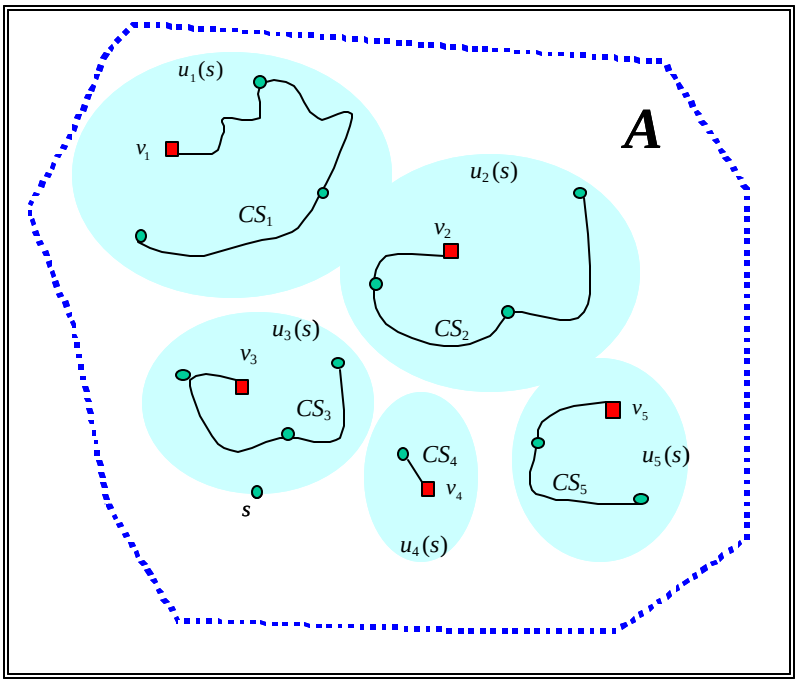

Figure 5.4 Graphical representation of controller actions u(s)

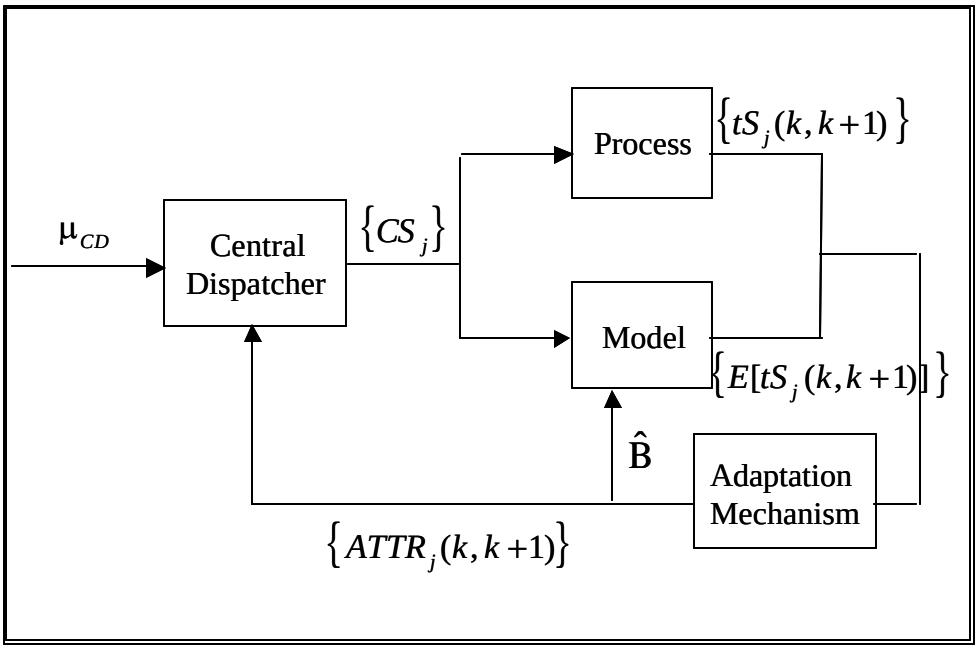

Figure 5.5 Overall block diagram of an APA approach for computing Stochastic Rerouting Delay

These parameters have been calibrated before $e(s)$ and are assumed to be representative of the system behavior during $e(s)$. Analytically,

$$E[tS_j(k,k+1)]\Big|_{e(s)} = F\left(tSN(a_j(k),a_j(k+1)), \overline{\Psi}, E[I_j(k,k+1)\right)\Big|_{e(s)} \tag{5.17}$$

where

$$E[I_j(k,k+1)]\Big|_{e(s)} = G\left(E[ATTR_j(k)], \mu_{CD}, \hat{\mathrm{B}}(s)\right)\Big|_{e(s)} \tag{5.18}$$

hence,

$$E[tS_j(k,k+1)]\Big|_{e(s)} = F\left(tSN(a_j(k),a_j(k+1)), \overline{\Psi}, G\left(E[ATTR_j(k,k+1)], \mu_{CD}, \hat{\mathrm{B}}(s)\right)\right)\Big|_{e(s)} \tag{5.19}$$

$\overline{\Psi}$ is a measured average of the expected additional travel time incurred by a vehicle due to an extra insertion into its original route, and it can be updated from system information every update time H_c as well as the set of parameters $\hat{}(s)$. Functions F and G do not seem to have a closed form, as will be explained in Section 5.5. Thus is why it is difficult to apply conventional APC model approaches in order to properly estimate (5.18) and (5.19).

Following the adaptation mechanism, every H_c the estimation quality is checked. Thus

$$\hat{\mathrm{B}}(s+1) = \begin{cases} \hat{\mathrm{B}}(s) & \text{if } tS_j(k,k+1)\Big|_{e(s)} \approx E[tS_j(k,k+1)]\Big|_{e(s)} \\ \hat{\mathrm{B}}(x(s-r), y_m(s-r)) \text{ for } r: 0,\ldots,H_e & \text{otherwise} \end{cases}$$

The estimation process $\hat{B}(\cdot)$ requires data obtained from the performance of the system for a time period H_e measured backward since the occurrence of event s. This parameter H_e is defined arbitrarily by the modeler in order to perform a robust and representative estimation of the stochastic model embedded in the definition of the expected number of pick-up insertions as will be discussed in the next section.

In summary, in this section the APC general structure was applied to the specific problem of computing the delays from rerouting of transit vehicles in real time, under stochastic and unknown demand generated dynamically. Provided the unknown internal relations between components of the system, and considering that the general expressions found in this formulation don't seems to have a closed form, it was not possible to accommodate the system to the conventional adaptive predictive modeling techniques, broadly described in Section 5.3. In the next section, a very detailed modeling approach is developed in order to solve the problem, calibrate parameters and estimate expected vehicle travel times in real time, assuming random demand generated dynamically, and a set of optimized embedded decision rules applied by the dispatching module according to the theory summarized in Chapter 4.

5.5 Stochastic rerouting delay for dynamic vehicle routing: general formulation and solution algorithms

5.5.1 Generalities

In the previous section, an APC scheme was introduced to find the stochastic routing delays in pick-up and delivery. However, given the complexity of such a system, it is hard to conceive a manner of finding analytical expressions for the transfer-function models involved. Therefore, in this section a complementary analytical formulation along with a solution methodology is introduced and developed under certain assumptions regarding the operation of the pick-up and delivery system.

The ultimate objective is to compute vehicle j's expected travel time between any two known points, as a function of the vehicle and system features, and based on future insertion decisions. As mentioned in the previous sections, these calculations are based on system-wide features and performance measures, which can be updated in real time based on historical information obtained from the system in a previous time period (in the context of the APC formulation, this is H_e). In this vision, the routing rules are allowed to change dynamically as the characteristics of the system change, resulting in a novel, control-based strategy for optimizing any dynamic pick-up and delivery problem regardless of the specific details.

Let us assume that the traffic conditions on the physical network are known at any time. This assumption could be debatable, in the sense that network traffic conditions represents an additional source of stochasticity to the system, which strictly

speaking should be superposed to the unexpected insertions factor. However, the study of the impact on routing rules due to change in traffic conditions is out of the scope of this analytical formulation. This effect will be studied utilizing a hybrid approach based upon simulation of a network at a microscopic level along with the routing-scheduling decisions taken at a macroscopic level, as explained in Chapter 6.

As in Chapter 4, it is important to mention that *all the variables to be defined and analytically formulated hereafter will depend explicitly on a predefined sequence of scheduled stops* CS_j*, known at time t when vehicle j's position is* v_j. Thus, the variables defining the status and availability of vehicle j at a particular scheduled stop k will be functions of the path followed by such a vehicle between its current position and k. In addition, the variables defined over a segment $(k, k+1)$ will also depend on the predefined sequence CS_j.

Now, under the same conditions of Section 5.4, the vehicle j's base travel time between any two stops of its sequence CS_j can be computed using the surface travel time tSN (as in Chapter 4 Section 4.2).

$$tS_j(k,k+1) = \begin{cases} tSN(a_j(k), a_j(k+1)) + E[tPHome] & \text{if } a_j(k+1) \text{ is a pick-up} \\ tSN(a_j(k), a_j(k+1)) + E[tDHome] & \text{if } a_j(k+1) \text{ is a delivery} \\ tSN(a_j(k), a_j(k+1)) & \text{otherwise} \end{cases} \quad (5.20)$$

The previous expression depends on the network travel time (assumed known at any time), including the boarding or alighting time, according to the operation required at stop k+1. If stop k+1 is just a reference point, such operation times are not considered.

The main premise of this formulation is that the real travel time between two scheduled stops of an arbitrary vehicle route should change because of possible future pick-up insertions decided by the dispatching module dynamically over time. Hence, a general expression is proposed for the expected travel time experienced by vehicle j for traveling from k to k+1, as an additive linear function of the deterministic travel time defined above and the expected number of insertions into the vehicle route as follows

$$E[tS_j(k,k+1)] = tS_j(k,k+1) + \Psi\left\{E[I_j(k,k+1)]\right\} \tag{5.21}$$

where

$$E[I_j(k,k+1)] = E[PI_j(k,k+1)] + E[DI_j(k,k+1)] \tag{5.22}$$

As defined in Chapter 4, $E[PI_j(k,k+1)]$ and $E[DI_j(k,k+1)]$ represent the expected number of pick-up and non-scheduled delivery insertions respectively. These are non-scheduled, i.e. they come from the expected accumulated pick-up insertions unknown to the modeler at decision time t. Moreover, Ψ is a coefficient to be calibrated, representing the additional time caused by one extra insertion into the original route. The additive form of (5.21) could be modified depending on special conditions to be analyzed for each specific case. In some scenarios, it could be more appropriate to define a multiplicative functional form for such an expression, however in principle an additive behavior seems reasonable.

Moreover, an additional variable has to be introduced here in order to measure the effective capacity of vehicles at any time-position spot. In fact, when assigning new passengers to a specific vehicle, it is required to make sure that the available space over

time will be able to accommodate the pick-up requests already scheduled for that vehicle, even if some of them are finally assigned to a different vehicle. In order to simplify the formulation, the fact that in reality the number of insertions must take an integer value is not taken into account. It makes sense because the final objective is to compute expected travel times, so the expected number of insertions does not have a physical meaning by itself. Basically the effective vehicle capacity on segment $(k, k+1)$ can be quantified through the approximated available space $SP_j(k, k+1)$ (in vehicle seats), which is calculated as follows

$$SP_j(k, k+1) = \left[L_{MAX} + \frac{E[DIP_j(k, k+1)]}{2} - \left(E[L_j(k)] + [LRES_j(k+1)]^+ \right) \right]^+ \qquad (5.23)$$

In some very rare cases, the numerical value of the original expression in (5.23) could be negative, which does not make any sense from a logical standpoint. That is why in the modeling procedure, the function $[\cdot]^+$ is used instead.

The second term of the right-hand side of equation (5.23) is a measure of the average extra available seats on the vehicle-segment resulting from future delivery insertions. For such a calculation the expected number of delivery insertions $E[DI_j(k, k+1)]$ has been split into delivery insertions from previous segments $E[DIP_j(k, k+1)]$, and delivery insertions from the current segment $E[DIC_j(k, k+1)]$. In most cases the latter is assumed to be zero, and it is assumed not to affect equation (5.23) in any case.

With regard to the expression in brackets, $LRES_j(k+1)$ represents the reserved space. This space needs to account for the maximum load that are known to occur in the future segments. Analytically

$$LRES_j(k+1) = \underset{n}{Max}\left\{\sum_{m=k+1}^{n}\left[RS_j(m)\,PS_{z_j(m)} - \left(1 - RS_j(m)\right)\right]\right\} \qquad \forall n : k+1,\ldots,N_j \qquad (5.24)$$

with PS_l representing the pick-up pool size and $RS_j(k)$ is a dummy for identifying the nature of stop k. This dummy variable was already defined in Chapter 4 as follows

$$RS_j(k) = \begin{cases} 1 & \text{if the } k^{th} \text{ stop in vehicle } j \text{ sequence is a pick-up} \\ 0 & \text{if the } k^{th} \text{ stop of vehicle } j \text{ sequence is a delivery} \end{cases}$$

Therefore, all terms in the summation in (5.24) account for the net load interchange at stop m.

In all the rerouting schemes within the adaptive predictive control framework presented in Sections 5.3 and 5.4, a key element is predicting the number of future insertions for any vehicle. This depends on predicting the decisions by the dispatch system in assigning new calls to any future segment of any vehicle. A key focus in the following sections is on developing models of the dispatch decisions, since in real time what we need is to predict how the system will evolve based on the nature of the dispatch algorithm.

The type of dispatch decisions causes changes in vehicle routes, number of insertions, etc., and the core of the prediction process is in modeling the nature of the algorithm's decision making.

In this dissertation, only the possibility of insertions by the dispatching module is modeled. Other things like those issues discussed in Chapter 4 regarding for example the decision to send a vehicle to the trunk network, or holding vehicles at stops, etc., haven't been treated as part of this framework yet. These topics have to be added to this methodology in the future after understanding their role as part of the dispatch algorithm.

Before starting with the methodology, it is advisable to understand the dynamics of the problem. In the next section, the details of the system are discussed taking into account how the system works and how the dispatching module takes routing decisions in order to accommodate the dynamically generated customers according to certain optimization rules.

5.5.2 Problem description

Let us describe the case of vehicle j, which at time t is in position v_j and it has been assigned to follow a certain sequence of stops CS_j combining pick-ups and deliveries. Graphically

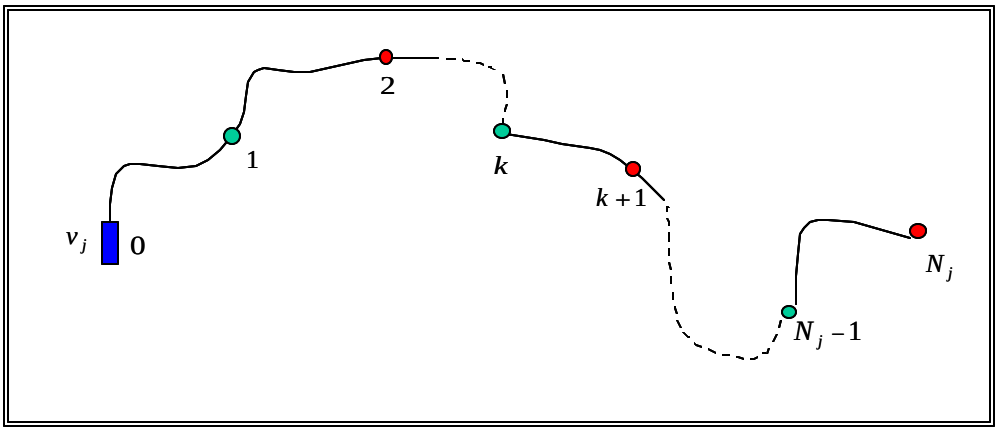

Figure 5.6 Graphical representation of a vehicle sequence of stops

The present time t is defined when vehicle is at position v_j. In addition, all the information at position v_j is assumed deterministic and known by the dispatching module. For any given future segment $(k, k+1)$, two different scenarios are distinguished regarding future insertions into the original route:

Scenario (A), in-advance (preinformed) reassignments: it includes possible future insertions in segment $(k, k+1)$ before vehicle j arrives to stop k. In other words, in this calculation, the estimation of potential insertions should account for all reassignments in $(k, k+1)$ occurring while vehicle j is moving from location 0 to stop k.

Scenario (S), sudden reassignments: after vehicle j leaves stop k, there is still a chance of having additional insertions on the same segment $(k, k+1)$, while vehicle j is traveling throughout the segment. This scenario considers that possibility.

Hereafter, the term vehicle-segment refers to a specific segment within a sequence of stops defined for a specific vehicle. It must be noticed that on the one hand, insertions in

(A) should occur more frequently that those having place in *(S)*, especially when the vehicle-segment under analysis is far from the current vehicle position, considering that there is more time for having new candidate calls entering the system before the vehicle arrives to that segment. However, on the other hand, users inserted in the context of scenario *(A)* will have to wait more at the pick-up location, since that transit vehicle will be normally farter than another candidate vehicle-segment analyzed under scenario *(S)* with respect to that call. The travel time component usually favor vehicle-segments in scenario *(A)*, while the waiting time component will definitely favor the insertion under the *(S)* option.

Regardless of the above description, in some cases the operational rules imposed by the transit provider could not allow insertions of type *(S)*. In fact, a basic operational restriction in order to keep stability and avoid infinite deferment of some users, could be not to allow insertions in segment $(k, k+1)$ once the vehicle entered the segment, that is, avoid modifying the imminent route.

Graphically, for both scenarios, reassignments should happen in the way as drawn in the following figure:

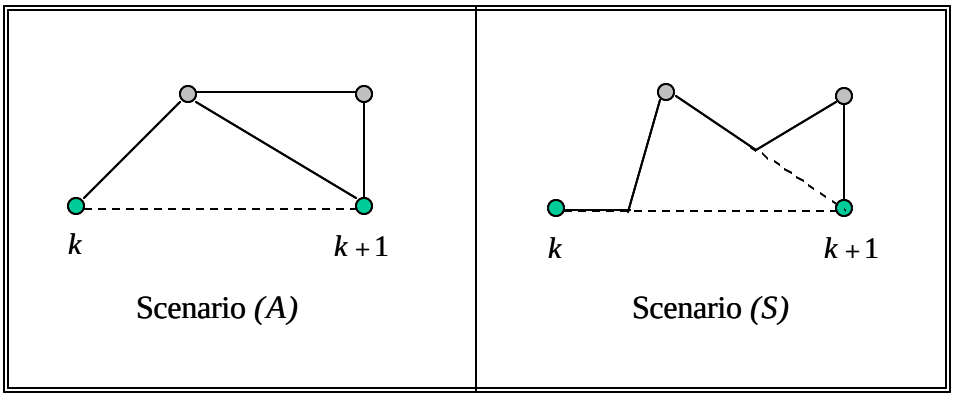

Figure 5.7 Possible reassignment of vehicle original route in both scenarios

Note that in *(A)* since insertions are scheduled before the vehicle arrives at stop k, they are assigned from the already decided stops, unlike scenario *(S)* where insertions are occurring whereas the vehicle is traveling towards the next scheduled stop. Conceptually, both schemes are different, which requires developing different analytical treatments for each case.

Finally, notice that the modeling of the expected number of insertions is a sequential process, from the conceptual nature of the potential insertion cases. In fact, scenario *(A)* has to be modeled first, and once it is already defined, case *(S)* has to be incorporated in the analysis provided that insertions coming from the modeling of *(A)* are now part of the system.

In summary, the final expected number of insertions will be the summation of those obtained from both scenarios, computed sequentially, according to the conceptual scheme described above. Analytically

$$E[I_j(k,k+1)] = E[I_j(k,k+1)]_A + E[I_j(k,k+1)]_S \tag{5.25}$$

5.5.3 Analytical formulation of the problem and solution methodology

5.5.3.1 In-advance reassignments: Scenario *(A)*

For the total area of influence A assume a total service network of length D in distance units [DU]. In addition, a rate of calls μ_C in [Calls/TU-DU] is assumed. Therefore, the rate of demand in [Calls/TU] can be computed approximately as $\mu_{CD} = D\mu_C$. In order to make the calculations simpler, it is assumed that the demand generation follows a

uniform distribution over the total length of the transit service network. In a more general case, it will be necessary to consider an empirical spatial distribution of the demand instead, however, this fact just complicates the calculation without adding any conceptual contribution on the formulation.

Let us first compute the value for the expected number of pick-ups insertions (in-advance reassignments) over segment $(k,k+1)$ in sequence CS_j of vehicle j, $E[PI_j(k,k+1)]_A$. Note that all intermediate variables defined in this section will depend upon segment $(k,k+1)$ associated to sequence CS_j, however this dependence will not be explicit on the notation only for the sake of simplicity and clarity.

Assuming that the demand rate represents suitably the dynamics of the system, the approximate number of candidate insertions for the advance scenario *CIA*, is computed as follows

$$\begin{aligned} CIA &= \mu_{CD} \sum_{r=0}^{k-1} E[tS_j(r,r+1)] \cdot \frac{AR_j(k,k+1)}{A} + E[DI_j(k,k+1)]_A \\ CIA &= CPA + E[DI_j(k,k+1)]_A \end{aligned} \tag{5.26}$$

where $AR_j(k,k+1)$ is the segment catchment area. At this point, another assumption is introduced. It will be assumed that segment $(k,k+1)$ has influence mainly over its surrounding area in space. In words, under normal operational conditions, and assuming that there are enough vehicles moving around waiting for a new pick-up assignment, it is reasonable to suppose that segment $(k,k+1)$ of vehicle j' route will be a real candidate to be considered as part of the pick-up decision only if the new request is generated within some predefined area (in space) around the segment, defined as the catchment

area $AR_j(k,k+1)$. Therefore, the probability for a call generated outside the catchment area of being inserted in such a segment is assumed to be negligible.

CIA is composed by two terms: *CPA*, which is the number of candidate pick-ups, and $E[DI_j(k,k+1)]_A$ representing the expected number of delivery insertions, both computed for the in-advance scenario. So far, they are assumed known and measurable. The value assigned to *CIA* does not have to be an integer number; therefore, a new variable is introduced, which is the ceiling of *CIA*, as $CCIA = \lceil CIA \rceil$. Later in the formulation the necessity of this computation will be addressed.

Conceptually, the catchment area could have any shape around segment $(k,k+1)$, but an elliptical form with extreme points determined by stops *k* and *k*+1 on the longest axis seems to accommodate very well to the problem requirements (see Figure 5.8). In addition, the eccentricity of the ellipse (*e*) is a parameter that could be calibrated based on the observed dispatch decisions, and eventually it could be part of the adaptive procedure as well. The details about the catchment area computations are treated in Appendix to Chapter 5.

Therefore, assuming knowledge of the expected number of delivery insertions, it is possible to compute the probability for a new insertion to be a pick-up P_P. The complement of P_P will represent the probability for a new insertion to be a delivery P_D. Analytically

$$P_P = \frac{CPA}{CIA} \qquad P_D = 1 - P_P \tag{5.27}$$

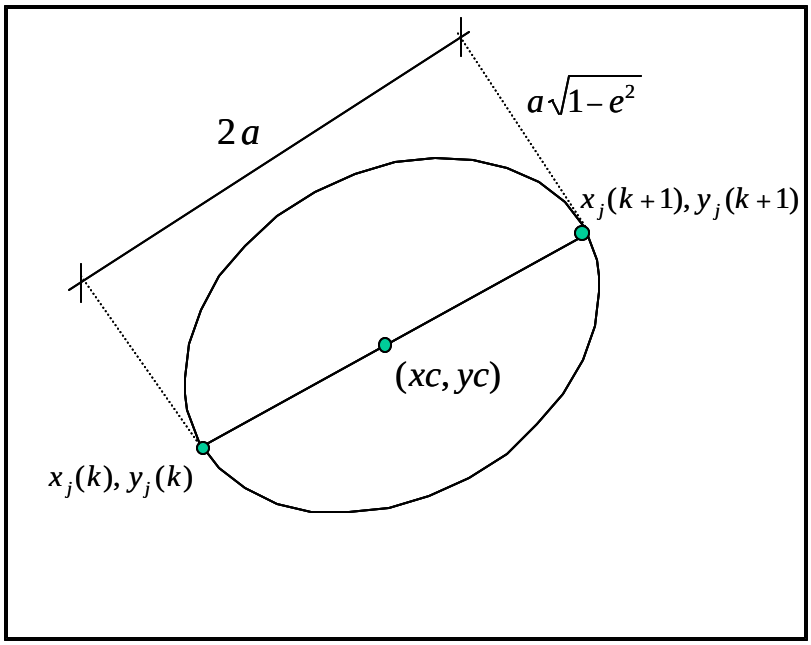

Figure 5.8 Segment $(k, k+1)$ catchment area

Another important decision factor, associated with the relative space-time position of the pick-up request with respect to the vehicle route, is the waiting time effect. This consideration motivates the inclusion of a new variable into the formulation $DCT(q)$, which is the difference in time between the expected arrival time of the vehicle to stop k and the expected time of generation of the new request q, where k is the upstream stop of segment $(k, k+1)$. Analytically

$$DCT(q) = E[tCL_j(k)] - EZt(q) \tag{5.28}$$

where $EZt(q)$ represents the expected call time corresponding to insertion q. This variable approximately measures the time the customer has been waiting for service before vehicle j enters the candidate insertion segment $(k, k+1)$ and it is defined only

for pick-up insertions. Notice that the $DCT(q)$ in expression (5.28) will always be zero for cases of sudden reassignment, as explained in Section 5.5.3.2.

Expressions (5.27) and (5.28) are needed in order to incorporate the dispatch decisions (measured quantitatively from the performance of the system) into the adaptive predictive modeling approach developed here.

At this point, in order to maintain consistency in the notation utilized to describe the insertion procedure, it is necessary to define an augmented sequence *CSA*, constructed from the real sequence CS_j, by incorporating all candidate insertions between elements *k* and *k* +1. Thus,

$$CSA = \{0,1,\ldots,k,k+1,\ldots k+L,\ldots,N_j+L\} \tag{5.29}$$

where $L = CCIA + 1$. Note that in expression (5.29), the ceiling of the candidate insertions is used instead of just *CIA*, since it is clear that the former is an integer number and therefore, it can be used as an index in the definition of *CSA*. Further on this chapter, a correction will be incorporated in order to account for this approximation. Hence, from stop *k* in both sequences, all remaining positions are shifted by *CCIA* in order to include the potential insertions as part of a virtual augmented vehicle's sequence. Thus, $a_j(k+1) = \overline{a}_j(k+L)$, where $\overline{a}_j(\cdot)$ denotes the same correspondence function between sequence position and physical position defined above, but now referred to the augmented sequence *CSA*. It has to be mentioned that the augmented sequence is not a real sequence of stops, it is just defined for clarifying the application of the methodology. In addition, henceforth all variables originally defined for describing

some element of CS_j will be written with a subscript AU when they are referred to the augmented sequence *CSA*.

The next definition is for the model in the adaptation mechanism, as defined in Sections 5.3 and 5.4, and it has to be calibrated using information obtained from the performance of the system. The model has been designed in order to estimate P_{A_j} the probability of inserting a new stop q between two stops r and s that belong to *CSA* and $k < r, s < k + L$. Analytically:

$$P_{A_j}(r, q, s; NI, PINS) = P_P \cdot P_{AS_j} + P_D \tag{5.30}$$

subject to expected load feasibility conditions defined below in (5.31).

$P_{AS_j} = P_{AS_j}(ATTR_j(r,s), DCT(q); NI, PINS)$ is the probability for any given call at any time in the catchment area of a vehicle's future segment to be inserted to that given segment. This probability depends on a few continuous variables. First, $DCT(q)$, where q is the new insertion stop as defined in (5.28).

P_{AS_j} also depends on *NI* and *PINS* because the segment conditions change depending on the previously assumed insertions. Thus, *NI* is a parameter that represents the number of accepted insertions between stops k and $k+ L$ both belonging to *CSA*, decided before q. *PINS* on the other hand, denotes a subset of *NI*, accounting for the number of accepted insertions decided before q, but now only between stops k and r in *CSA*.

Finally, P_{AS_j} depends on few factors dealing with the current vehicle as well as other vehicles that compete with it. These factors can be represented as $ATTR_j(r, s)$.

A graphic representation of such a decision is shown in Figure 5.9.

The set of attributes $ATTR_j(r,s) = \{SPA_j(r,s), TR_j^{AU}(r), WSP_j^{AU}(r,s)\}$ is composed by three components: the approximated available space for future pick-ups in the context of the in-advance scenario $SPA_j(r,s)$, the cumulative time of passengers on board when vehicle j leaves stop r, and the aforementioned available space for potential passengers in the proximity of segment $(k, k+L)$.

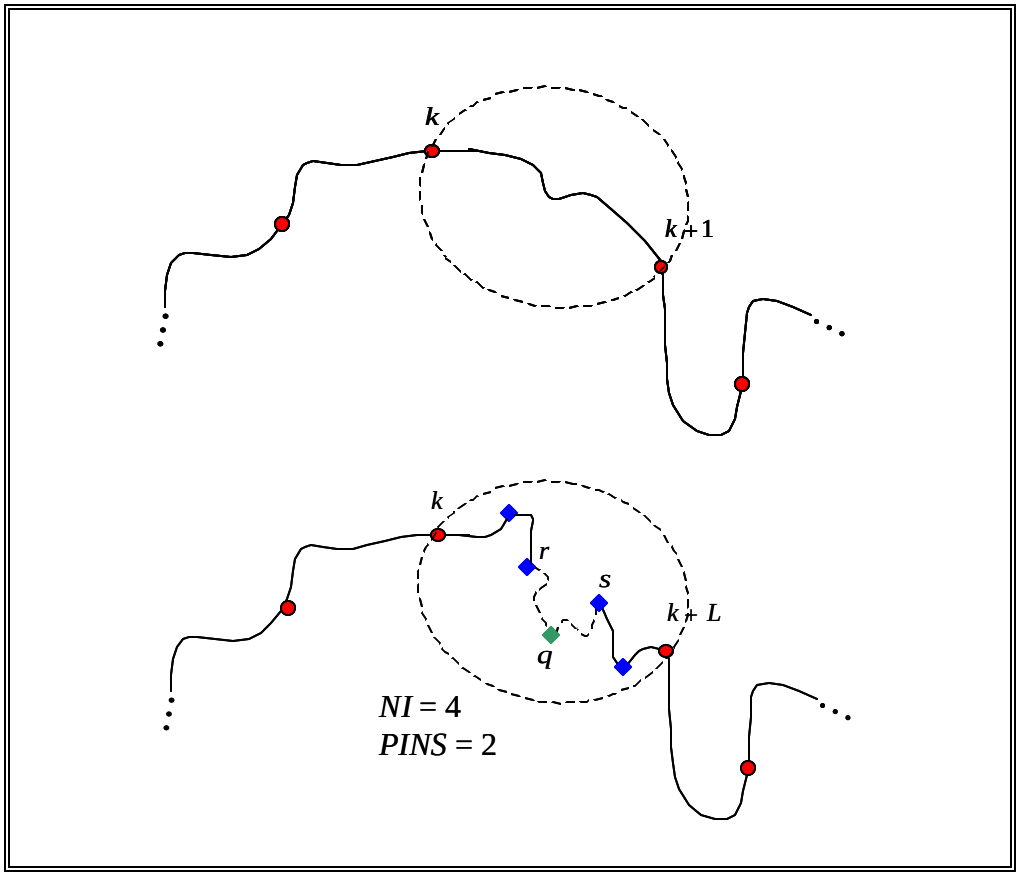

Figure 5.9 Insertion of new call *q* representation

The expected load feasibility condition constraining the assignment probability in (5.30) is based upon the space of the vehicle needed for future already scheduled insertions. Analytically,

$$
\begin{aligned}
& E[L_j^{AU}(r)] + P_P E[PS] - P_D \le SPA_j(r,s) \\
\Leftrightarrow\ & E[L_j^{AU}(r)] + P_P E[PS] - 1 + P_P \le SPA_j(r,s) \\
\Leftrightarrow\ & E[L_j^{AU}(r)] + P_P (E[PS] + 1) \le SPA_j(r,s) + 1
\end{aligned}
\tag{5.31}
$$

A second feasibility condition could be added to expression (5.30), more as an operational constraint. It could be required that the number of decided reassignments in a preceding position with respect to stop $k+L$ of the augmented sequence (originally $k+1$), defined as $NIT(k+L)$, be bounded by an arbitrary parameter *MRA* (maximum number of reassignments allowed), which measures the maximum number of insertions allowed into a predetermined vehicle's route. That is $NIT(k+L) < MRA$.

Conceptually, $NIT(k+L)$ represents the maximum number of reassignments experienced by a scheduled stop $k+L$, measured from the time stop $k+L$ was originally assigned to such a route. In fact, the dynamic nature of the proposed system justifies this additional constraint limiting the pick-up (or delivery) shift, in order to prevent indefinite deferment.

The model for effective pick-ups insertions can be calibrated using data taken from the system by observing the insertion decision taken by the dispatching module. In fact, $P_{AS_j}(ATTR_j(r,s), DCT(q); NI, PINS)$ is modeled as a binary discrete model, in which the dependent variable, say y, is binary. The way to model this behavior properly is by utilizing the *probit* analysis model introduced by Goldberger (1964), and nicely summarized by Maddala (1983). It is assumed that there is an underlying response variable y_i^* defined by the regression relationship

$$
y_i^* = \beta' x_i + \mu_i \tag{5.32}
$$

In practice, y_i^* is unobservable. What it is really observed is a dummy variable *y* defined by

$$\begin{aligned} y &= 1 \quad if \;\; y_i^* > 0 \\ y &= 0 \quad otherwise \end{aligned} \tag{5.33}$$

All the modeling framework taken from Maddala (1983), summarizing the theory and calibration techniques for *probit* and *logit* models are described in Appendix to Chapter 5. It is also appended a discussion about the similarities and differences between both approaches, emphasizing the advantages, disadvantages and constraints belonging to each modeling technique.

In this particular case, it is clear the modeled binary choice, either the dispatching module accepts the new pick-up request to be inserted into a specified segment (r,s) or he rejects it, given the conditions of the vehicle when getting to that segment. Thus, the deterministic portion of the right-hand side of equation (5.32) can be written as follows

$$\beta' x_i = \beta_0 + \beta_1 \frac{SPA_j(r,s)}{E[PS]} + \beta_2 TR_j^{AU}(r) + \beta_3 WSP_j^{AU}(k,k+L) + \beta_4 DCT \tag{5.34}$$

These variables are explained in detail next. The variables on the right-hand side of expression (5.34) are the ones that supposedly should influence the pick-up decision taken by the dispatching module under the cost function structure and formulation of Chapter 4. The kind of data collected from the system performance and the calibration methodology will be presented in much more detail in Section 5.6 and Appendix to Chapter 5.

With regard to the calculation of the indicator $DCT(q)$, it is necessary to visualize the new insertion q as part of the potential insertion sub-sequence $CSAT = \{k+1,\ldots,k+L-1\}$ containing all potential insertions added to CS_j. Thus, if $q = k + p, \quad q \in CSAT$, then

$$DCT(k+p) = \left(1 - \frac{p}{L}\right)\sum_{r=0}^{k-1} E[tS_j(r, r+1)] \tag{5.35}$$

assuming that the inter-arrival call time is uniformly distributed over the whole period of generation of candidate insertions.

The definition of the approximated available space for future pick-ups in the context of the in-advance scenario $SPA_j(r,s)$ is computed likewise equation (5.23) for the calculation of $SP_j(k,k+1)$, with the difference that in this case the expected additional insertions on the way between *r* and *s* have been incorporated. That is

$$\begin{aligned} SPA_j(r,s) = [&L_{MAX} + PD \cdot (NI - PINS) + \\ &- \left(E[L_j^{AU}(k)] + PINS\, E[PS] \cdot PP + [LRES_j^{AU}(k+L)]^+\right)]^+ \end{aligned} \tag{5.36}$$

that can also be written as

$$\begin{aligned} SPA_j(r,s) = [&L_{MAX} + PD \cdot NI + \\ &- \left(E[L_j^{AU}(k)] + PINS\left(E[PS] \cdot PP + PD\right) + [LRES_j^{AU}(k+L)]^+\right)]^+ \end{aligned} \tag{5.37}$$

In (5.36) and (5.37), $LRES_j^{AU}(k+L)$ is equal to expression $LRES_j(k+1)$ computed in (5.24). In (5.33), $PD(NI - PINS)$ measures the expected extra spaces resulting from the expected deliveries between r and $k+L$ while $PINS\,E[PS]\cdot PP$ considers the expected required spaces to accommodate the pick-ups happening between stops *k* and *r*.

Notice that if $E[PS]=1$, (5.37) turns out to be

$$SPA_j(r,s) = \left[L_{MAX} + PD\cdot NI - \left(E[L_j^{AU}(k)] + PINS + [LRES_j^{AU}(k+L)]^+\right)\right]^+ \tag{5.38}$$

The approximated measure of the available space for potential passengers in the proximity of segment $(k,k+L)$ is computed in Appendix to Chapter 5. This variable is assumed to remain almost invariant over the whole segment $(k,k+L)$ and since (r,s) belongs to $(k,k+L)$, then $WSP_j^{AU}(s,r) \approx WSP_j^{AU}(k,k+L)$ for all computation purposes. Finally, the cumulative travel-time experienced by all passengers on board when vehicle *j* leaves stop *r*, is calculated as

$$TR_j^{AU}(r) = TR_j^{AU}(k) + \frac{\alpha_A}{2} PINS^2\cdot E[PS]\cdot PP \tag{5.39}$$

where $TR_j^{AU}(k) = TR_j(k)$ as it was already computed in Chapter 4.

$\alpha_A = \left(\dfrac{tS_j^{AU}(k,k+L) + \overline{\Psi}\cdot NI}{NI+1}\right)$ is an approximated measure of the expected travel time experienced by the vehicle when traveling between two consecutive stops that belong to

segment $(k, k+1)$. $\overline{\Psi}$ is a parameter accounting for the average extra delay per insertion measured from the system. Basically, all pick-ups inserted on stretch (k, r) suffer an extra delay of $\frac{\alpha_A \cdot PINS}{2}$ on average. Note that $\alpha_A \cdot PINS$ is an approximated measure of the vehicle travel time on sub-segment (k, r) under the same conditions.

With regard to expression (5.30), it must be noticed that the operation of the system requires the expected delivery insertions to happen on the pre-assigned segment, with probability equal to 1. The policy in this sense is designed to maintain system consistency. Thus, once a new delivery is scheduled to happen on certain stretch, it is a certainty that it will be delivered on that stretch. This consideration explains why the effective assignment probability is defined exclusively for pick-ups, since non-scheduled deliveries have to be inserted anyway.

Scenario (*A*): Expected number of pick-up insertions calculation using a branched process approach (BPA)

In order to estimate the expected number of pick-up insertions occurring over certain segment $(k, k+L)$ associated to the augmented vehicle's sequence *CSA*, it is postulated as a branched process approach (BPA). The idea behind the BPA treatment is to create a tree with its root at stop k. The branches emerge initially from stop k and grow towards stop $k+L$. Each level of the tree represents the inclusion of a new potential insertion in the order defined by the potential insertion sub-sequence *CSAT* defined above in this section. Each branch of the tree, emerging from a node at level r for example, has an associated probability, and represents one possible insertion order over the previous

sequence corresponding to its parent branch computed at level *r*-1. The probability of each branch to happen is based upon all previous calculations and upon the probabilistic model of expression (5.30).

Analytically, the expected number of pick-up insertions on segment $(k, k+L)$ for the advance scenario can be computed through the following recursive function:

$$E[PI_j(k,k+1)]_A = E[PI_j^{AU}(k,k+L)]_A = \Lambda_j(k,k+L;k+1) \tag{5.40}$$

where

$$\Lambda_j(k,k+L;k+p) = \begin{cases} \Lambda B_j(k,k+L;k+p) & \text{if } p < L \\ 0 & \text{otherwise} \end{cases} \tag{5.41}$$

and

$$\begin{aligned} \Lambda B_j(k,k+L;k+p) = {} & P A_j(k,k+p,k+L;0,0)\Lambda_j(k,k+p,k+L;k+p+1) + \\ & + \{1 - P A_j(k,k+p,k+L;0,0)\}\Lambda_j(k,k+L;k+p+1) \end{aligned} \tag{5.42}$$

Expression (5.42) is written as a function of the branch object defined by $\Lambda_j(k,k+L;k+p)$, which represents the event of inserting the new request *k+p* between *k* and *k+L*. Notice that if there are no more potential insertions to be considered, expression (5.41) closes the recursion assigning a value equal to zero to the object (no insertions accepted). In expression (5.42), the basic object (5.41) is a function of $\Lambda_j(k,k+p,k+L;k+p+1)$, generating a new object characterizing the potential insertion of *k+p+1* somewhere on a predetermined sequence of new insertions (in this case containing just one element *k+p*).

Let us define $S(u,v) = \{s(u), s(u+1), s(u+2)\ldots, s(v)\}$ as a sequence of accepted insertions occurring between k and $k+L$ in some order determined by the definition of $s(i)$ and associated with some branch. Then, in the general case, assuming that a sequence $S(1,m)$ has been already accepted:

$$\Lambda_j(k, S(1,m), k+L; k+p) = \begin{cases} \Lambda B_j(k, S(1,m), k+L; k+p) & \text{if } p < L \\ mPP & \text{if } (p = L) \wedge (k+L-1 \notin S(1,m)) \\ \{m-1+P_{LAST}\}PP & \text{if } (p = L) \wedge (k+L-1 \in S(1,m)) \end{cases} \tag{5.43}$$

and

$$\begin{aligned} &\Lambda B_j(k, S(1,m), k+L; k+p) = \\ &= \frac{1}{m+1}\Big\{ PA_j(k, k+p, s(1); m, 0)\Lambda_j(k, k+p, S(1,m), k+L; k+p+1) + \\ &+ [1 - PA_j(k, k+p, s(1); m, 0)]\Lambda_j(k, S(1,m), k+L; k+p+1) + \\ &+ \sum_{r=1}^{m-1}\Big(PA_j(s(r), k+p, s(r+1); m, r)\Lambda_j(k, S(1,r), k+p, S(r+1,m), k+L; k+p+1) + \\ &+ [1 - PA_j(s(r), k+p, s(r+1); m, r)]\Lambda_j(k, S(1,m), k+L; k+p+1)\Big) + \\ &+ PA_j(s(m), k+p, k+L; m, m)\Lambda_j(k, S(1,m), k+p, k+L; k+p+1) + \\ &+ [1 - PA_j(s(m), k+p, k+L; m, m)]\Lambda_j(k, S(1,m), k+L; k+p+1)\Big\} \end{aligned} \tag{5.44}$$

Expression (5.43) has now two different options to close the recursion. The problem has been split into two cases in order to correct the fact that *CIA* not necessarily is an integer, motivating the definition of a new variable *CCIA* in order to generate a countable number of events in the BPA scheme. The way to correct this difference is by assuming that the first *L-2* candidate calls enters the system with probability equal to 1, however the last candidate request *k+L-1* could eventually not appear while vehicle is moving

along segment $(k, k+L)$. It is assumed it has a probability of occurrence equals to $P_{LAST} = 1 - CCIA + CIA$. In case that $CIA=CCIA$, $P_{LAST} = 1$.

Hence, the case $p=L$ and $k+L-1 \notin S(1,m)$ in expression (5.43), means that the last candidate insertion does not belong to the previous accepted insertion sequence $S(1,m)$, in which case the number of accepted pick-up insertions associated to that specific path defined for certain sequence of branches of the tree is equals to the number of insertions m times the probability for a insertion to be a pick-up P_P. On the other hand, if $p=L$ and $k+L-1 \in S(1,m)$, it is assumed that the expected number of insertions also has to incorporate the probability of insertion $k+L-1$ to happen, as in expression (5.43).

Expression (5.44) is only an elaborate recursive expression that captures the different chances of inserting $k+p+1$ after m previously accepted insertions happening in the order defined by sequence $S(1,m)$. In addition, there is no clear reason for assuming that certain sequence order should be better than another in the sense of optimizing the system performance, therefore, it has been assumed that all branches emerging from a common root at certain level, are equally weighted (by a factor equals to 1 over $m+1$ in expression (5.44)). Each one represents a potential sequence order with the new request, and the probability of insertion of it under that configuration is not the same for each configuration. In fact, it depends on the probability of assignment under each specific configuration conditions, as defined above in model (5.34).

Notice that if $m=1$, equation (5.44) still works. However, in such a case $S(1,m) = S(1,1) = s(1)$. In this case, the summation does not make any sense, and therefore, it has to be discarded from the calculations.

Let us illustrate the meaning and the calculations behind expressions (5.43) and (5.44) via a conceptual example. The numerical models and a real application of the BPA will be developed and explained in Chapter 7 for the different simulated scenarios. Suppose, there are not any expected delivery insertions for a sequence $(k, k+1)$ that belongs to certain vehicle j's route CS_j, i.e. $P_P = 1$. From the demand rate and all the segment conditions, it was found that only two candidate insertions could eventually happen. Assume also that *CIA*=*CCIA*, in order to simplify the analysis.

Thus, $CSAT = \{k+1, k+2\}$, and L=3. Customers are generated in the order as in *CSAT*, that is, k+1 enters the system first, and next k+2 appears. That means that the decision process is sequential in the sense that the insertion of k+1 is decided first, k+2 is analyzed next, and so on, as would happen in the real world where it has been assumed that routing decisions are taken sequentially, one by one, at least in the reroutable portion of the *HCPPT* vehicle' routes. Then,

$$E[PI_j^{AU}(k, k+3)]_A = \Lambda_j(k, k+3; k+1) \tag{5.45}$$

here (using 5.40))

$$\Lambda_j(k, k+3; k+1) = \Lambda B_j(k, k+3; k+1) = P_{A_j}(k, k+1, k+3; 0, 0)\,\Lambda_j(k, k+1, k+3; k+2) + \\ + \{1 - P_{A_j}(k, k+1, k+3; 0, 0)\}\Lambda_j(k, k+3; k+2) \tag{5.46}$$

Next, a closed expression for $\Lambda_j(k, k+1, k+3; k+2)$ and $\Lambda_j(k, k+3; k+2)$ has to be computed in the next step. The former through equation (5.44), while the latter using (5.42) as before. Analytically

$$\Lambda_j(k,k+1,k+3;k+2) = \Lambda B_j(k,k+1,k+3;k+2) \tag{5.47}$$

$$\Lambda_j(k,k+3;k+2) = \Lambda B_j(k,k+3;k+2) \tag{5.48}$$

where in (5.47), $S(1,1) = s(1) = k+1$ represents the previous insertion generated at the first tree level (equation (5.46)). Thus,

$$\begin{aligned}\Lambda B_j(k,k+1,k+3;k+2) = \frac{1}{2}\{ & P_{A_j}(k,k+2,k+1;1,0)\,\Lambda_j(k,k+2,k+1,k+3;k+3) + \\ & + [\,1 - P_{A_j}(k,k+2,k+1;1,0)\,]\,\Lambda_j(k,k+1,k+3;k+3) + \\ & + P_{A_j}(k+1,k+2,k+3;1,1)\,\Lambda_j(k,k+1,k+2,k+3;k+3) + \\ & + [\,1 - P_{A_j}(k+1,k+2,k+3;1,1)\,]\,\Lambda_j(k,k+1,k+3;k+3)\,\}\end{aligned} \tag{5.49}$$

and

$$\begin{aligned}\Lambda B_j(k,k+3;k+2) = & P_{A_j}(k,k+2,k+3;0,0)\,\Lambda_j(k,k+2,k+3;k+3) + \\ & + \{1 - P_{A_j}(k,k+2,k+3;0,0)\}\,\Lambda_j(k,k+3;k+3)\end{aligned} \tag{5.50}$$

The remaining objects close the recursion, since p=3=L. Therefore, from conditions (5.41) and (5.42):

$$\begin{aligned}&\Lambda_j(k,k+2,k+1,k+3;k+3) = 2 \\ &\Lambda_j(k,k+1,k+2,k+3;k+3) = 2 \\ &\Lambda_j(k,k+1,k+3;k+3) = 1\end{aligned} \tag{5.51}$$

By replacing (5.51) back into (5.49) and (5.50), and performing some analytical work, the following two expressions are obtained

$$\Lambda B_j(k,k+1,k+3;k+2) = 1 + \frac{1}{2}\{P_{A_j}(k,k+2,k+1;1,0) + P_{A_j}(k+1,k+2,k+3;1,1)\} \tag{5.52}$$

$$\Lambda B_j(k,k+3;k+2) = P_{A_j}(k,k+2,k+3;0,0) \tag{5.53}$$

Finally, by replacing (5.52) and (5.53) into (5.46), the expected number of pick-ups insertions can be computed as

$$
\begin{aligned}
&E[PI_j^{AU}(k,k+3)]_A = \\
&= PA_j(k,k+1,k+3;0,0)\left(1+\frac{1}{2}\{PA_j(k,k+2,k+1;1,0)+PA_j(k+1,k+2,k+3;1,1)\}\right)+ \\
&+\{1-PA_j(k,k+1,k+3;0,0)\}PA_j(k,k+2,k+3;0,0)
\end{aligned}
\tag{5.54}
$$

Note the simplicity of expression (5.54). Let us interpret such an expression in the context of this methodology. Each probability in (5.54) represents a specific insertion event. Assuming that all possible events are feasible, it is possible to interpret each event as a sequence of accepted insertions occurring with certain probability. Such probability expressions have to be computed using the discrete choice methodology explained analytically through expressions (5.30)-(5.39) above refereed as the pick-up insertion assignment probability.

In (5.46), the only possible cases are graphically represented in Figure 5.10. Note that, there are two cases in which the total number of insertions are 2, and other two cases with just one insertion. Additionally, there is a chance of having no insertions (not depicted in Figure 5.10). That case does not appear explicitly in (5.54) as well, since the associated term has been multiplied by zero in the final state of the BPA (see right hand condition of expression (5.41)).

The final tree generated by the BPA process is showed in Figure 5.11. In the figure, each branch of the tree has an associated insertion scenario, where all branches

that belong to the inferior level ("child") depend on the scenario of the corresponding branch at the superior level ("parent"). Figure 5.11 is also consistent with expression (5.54). The first term of the right hand side of equation (5.54) can be seen as the sequence of branches at the left side of the figure. The expression in parenthesis multiplied by $P_{A_j}(k, k+1, k+3; 0, 0)$ represents the likelihood associated to the "child" nest of the left side of the figure. In such an expression, the insertion of k+2 is assumed to have the same chance of occurring whether to the left or to the right position with respect to the previous insertion k+1. That is why the factor 0.5 appears within the parenthesis. The interpretation of the right sequence of branches is analogous.

Each path from the root node towards the bottom of the tree yields a pick-up insertions number *PI*, having certain probability according to the path sequence along the tree (from the top to the bottom). In this simple case, expression (5.54) summarized the final computation. In the general case, recursive equations (5.42) and (5.44) provide the right way to compute the expected number of pick-up insertions under all the aforementioned assumptions.

Let us apply some numbers. If for instance from the discrete choice model, it were known that

$$\begin{aligned} &P_{A_j}(k, k+1, k+3; 0, 0) = 0.5 \\ &P_{A_j}(k, k+2, k+1; 1, 0) = 0.25 \\ &P_{A_j}(k+1, k+2, k+3; 1, 1) = 0.25 \\ &P_{A_j}(k, k+2, k+3; 0, 0) = 0.5 \end{aligned} \tag{5.55}$$

then, expressions (5.54) and (5.21) would yield

$$\begin{aligned} &E[PI_j^{AU}(k, k+3)]_A = 0.5*1.25 + 0.5*0.5 = 0.875 \\ &E[tS_j^{AU}(k, k+3)]_A = tS_j^{AU}(k, k+3) + \overline{\Psi} * 0.875 = 15 + 4.375 = 19.375 \quad \text{(min)} \end{aligned} \tag{5.56}$$

considering a segment of 15 minutes and a measured extra time per insertion of 5 minutes. In this particular case, the expected travel time on that segment to be considered in the cost function formulation, affecting both operator and users, should be 20 minutes instead of 15 minutes used if the approach would not be applied. Though this case does not show a relevant difference it is important to account for such differences. For example, in case of a situation where insertions were far away one each other, and therefore, the calibrated factor $\overline{\Psi}$ were much greater than 5 minutes (say, around 10 to 15 minutes), the bias for not including such a difference could become considerable.

So far, for the advanced scenario, along the whole formulation it has been assumed that $E[DI_j(k,k+1)]_A$ is known for all segments that belong to CS_j. In the next sub-section a methodological and analytical approach for estimating the expected number of non-scheduled delivery insertions is presented, assuming known all features of previous segments, including the total expected number of pick-up insertions on those segments, that is, by knowing

$$E[PI_j(r,r+1)] = E[PI_j(r,r+1)]_A + E[PI_j(r,r+1)]_S,$$

for all segment $(r,r+1)$ such that $r+1 \leq k$. Notice that the second term of the right-hand side in the previous expression (corresponding to the sudden scenario *S*) will be computed analytically in Section 5.5.3.2.

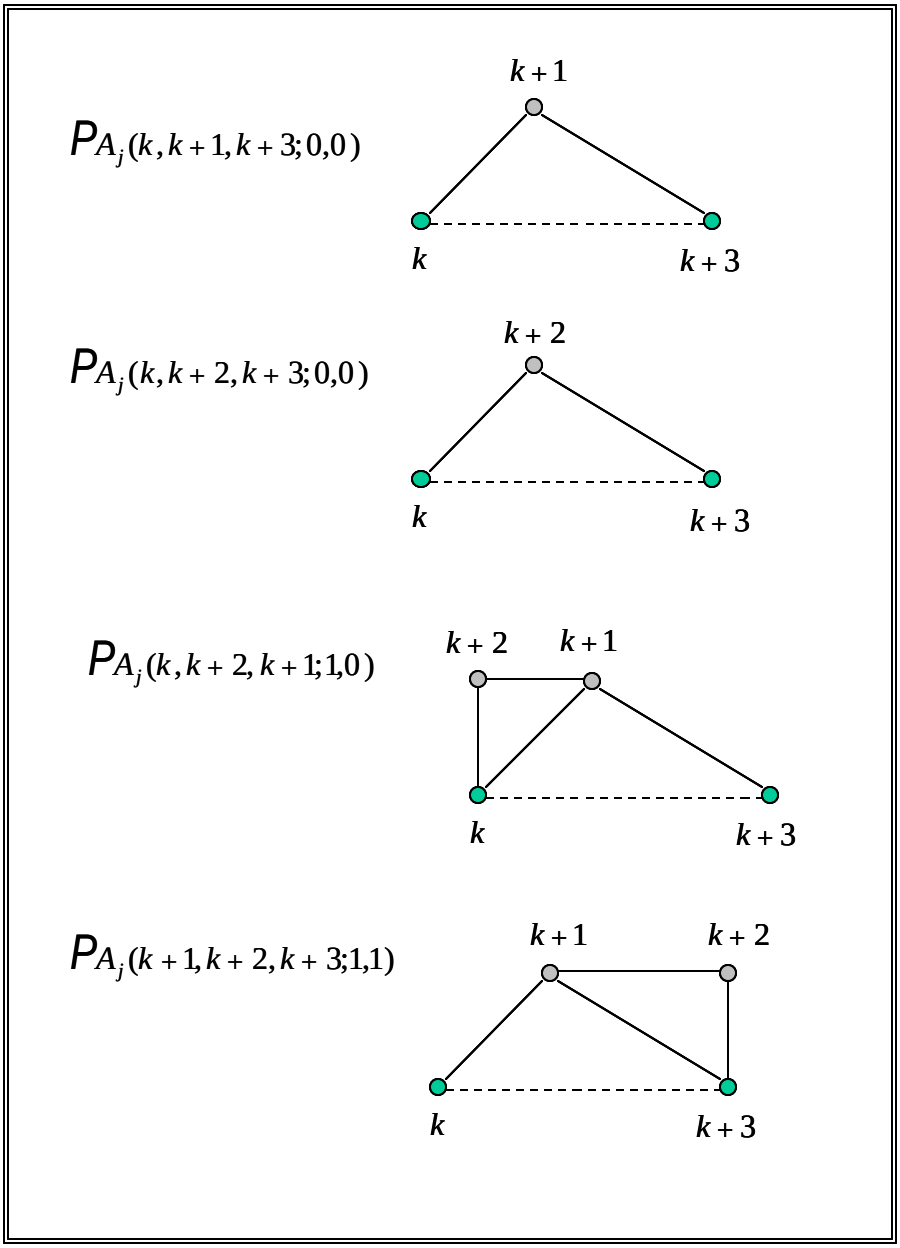

Figure 5.10 Interpreting the probability of pick-up assignments

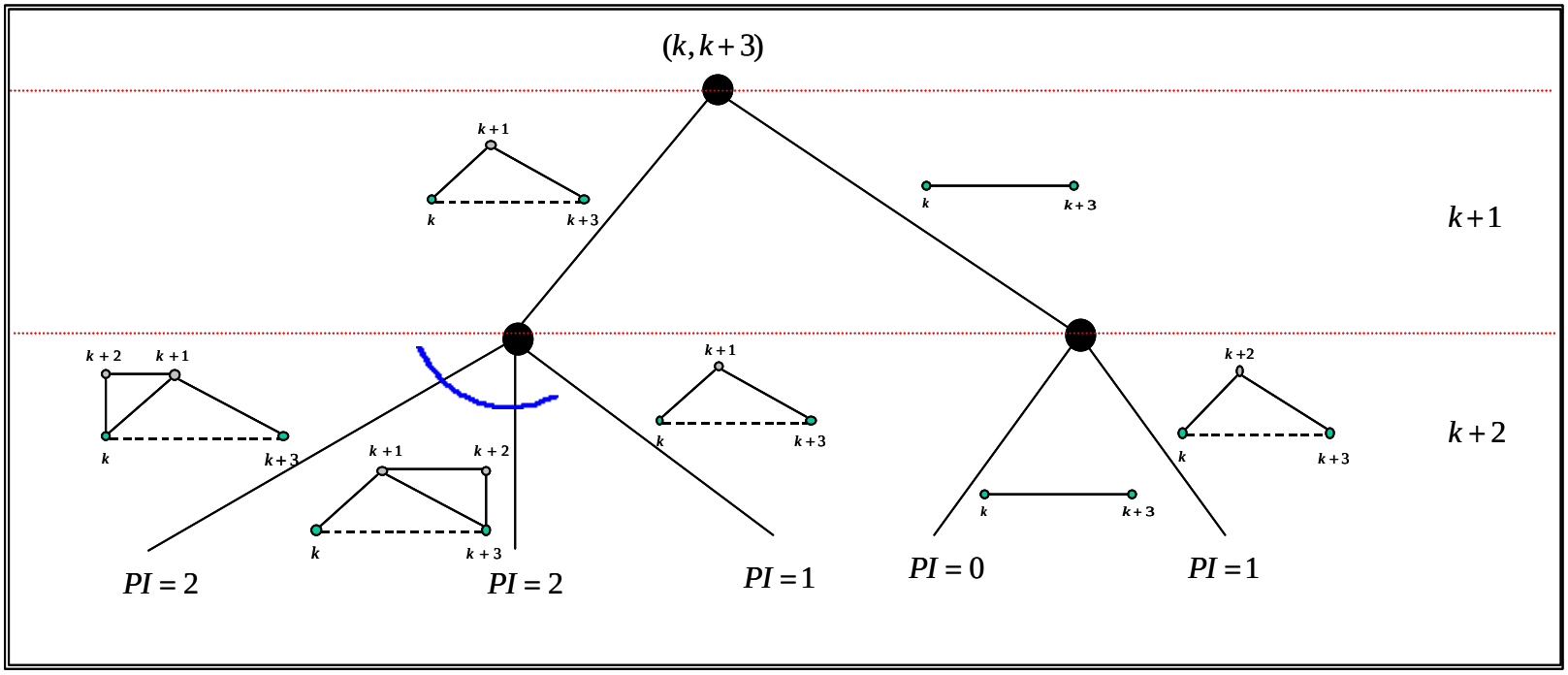

Figure 5.11 Branched process approach (BPA) tree example

Scenario (*A*): Expected number of delivery insertions calculation

For estimating the expected number of delivery non-scheduled insertions at decision time t for the advanced scenario (*A*), it is required to know the total number of expected pick-up insertions for all segments preceding segment $(k, k+1)$. However, the problem could eventually depend on non-scheduled deliveries on segment $(k, k+1)$ coming from a non-scheduled pick-up occurring on the same segment, generating a cumbersome iterative process (see expression (5.26)). This last phenomena is very unlikely to happen, only on long segments this effect could have a considerable effect on the travel time estimation. Even though, it seems not very practical to solve the iterative process in most cases, the algorithm is conceptually presented next in Figure 5.12.

In summary, all unexpected delivery insertions occurring on segment $(k, k+1)$ are caused by the expected number of pick-up insertions computed previously and sequentially segment by segment, from the current vehicle position (stop 0) till reaching stop $k+1$. Thus, the total number of passengers to be delivered from all unexpected insertions on segment $(r, r+1)$, can be computed as follows

$$NPI_j(r, r+1) = E[IS]\,E[PI_j(r, r+1)] \tag{5.57}$$

where $E[PI_j(r, r+1)]$ is the total number of pick-up insertions occurring on segment $(r, r+1)$ and $E[IS]$ is the expected trips per pick-up stops introduced in Chapter 4. Note that in this case, $E[IS] = E[PS]$ since $E[ES] = 0$.

From either the performance of the system or by collecting historical data, it is possible to estimate the expected value and the variance of the length of internal trips for such a system. Let us call them $\mu = E[tIT]$ and $\sigma^2 = Var(tIT)$ respectively. Moreover, let us assume that the length of internal trips is distributed $N(\mu, \sigma^2)$, according to the parameters obtained from the system itself. On the other hand, let us assume that the start time of the trip associated to those passengers to be delivered (or individual trips) as in (5.57), is distributed uniformly over stretch $(r, r+1)$. By using a similar approach as that applied in the pervious sub-section, a discrete number of trips coming from the unknown pick-up insertions of equation (5.54) can be found, entering the system with the following inter-event time:

$$\Delta t_j(r, r+1) = \frac{E[tCL_j(r+1)] - E[tCL_j(r)]}{CNPI_j(r, r+1)} \tag{5.58}$$

where $E[tCL_j(r)]$ and $E[tCL_j(r+1)]$ are the expected clock time at stops r and r+1 respectively. In addition, to make the denominator of (5.58) discrete, define a corrected number of pick-up insertions, *CNPI*

$$CNPI_j(r, r+1) = \lceil (NPI_j(r, r+1)) \rceil \tag{5.59}$$

A correction accounting for such a simplification, as in the previous section, will be introduced later in the formulation. Given that, and arbitrary trip starting between stops r

and r+1, has certain chance of falling into segment $(k, k+1)$, according to the following graphical representation:

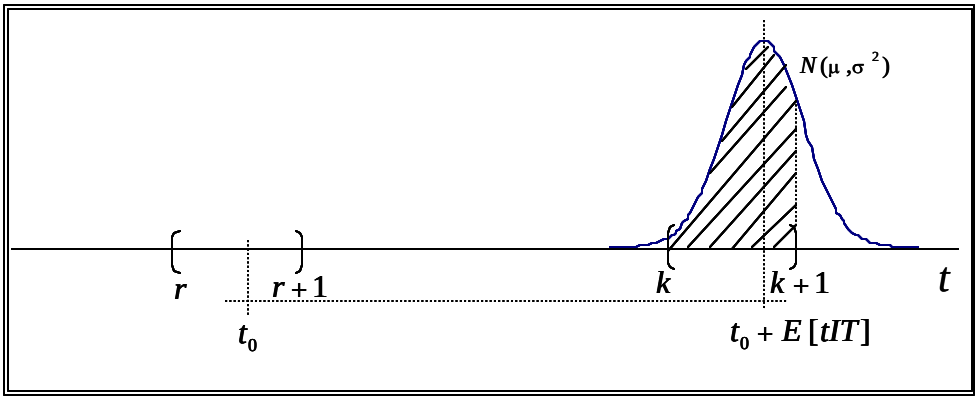

Figure 5.12 Probability of arrival of an arbitrary customer on segment $(k, k+1)$

The shaded area in Figure 5.12 can be found by knowing the values of the cumulative distribution function for the standard normal $\Phi(x)$, according to the following analytical expression:

$$\lambda_j(t_0; k, k+1) = \Phi\left(\frac{E[tCL_j(k+1) - t_0 - E[tIT]}{\sqrt{Var(tIT)}}\right) - \Phi\left(\frac{E[tCL_j(k) - t_0 - E[tIT]}{\sqrt{Var(tIT)}}\right) \quad (5.60)$$

Expression (5.60) can be seen as the contribution to the expected number of delivery insertions due to the individual trip starting at t_0. Therefore, by applying expression (5.60) for all the $CNPI_j(r, r+1)$ new individual trips starting on stretch $(r, r+1)$, and summing over all segments $(r, r+1)$ preceding stretch $(k, k+1)$, that is to say for all

segments containing the stops from position 0 to k, the expected number of delivery insertions for the advance scenario between stops k and k+1 can be computed as follows

$$E[DI_j(k,k+1)]_A = \sum_{r=0}^{k} \sum_{i=1}^{CNPI_j(r,r+1)} \lambda_j(E[tCL_j(r)] + \Delta t_j(r,r+1); k,k+1)\ \Gamma_j(i;r,r+1) \quad (5.61)$$

where

$$\Gamma_j(i;r,r+1) = \begin{cases} 1 & \text{if } i < CNPI_j(r,r+1) \\ PLASTD_j(r,r+1) & \text{otherwise} \end{cases} \quad (5.62)$$

The correction factor $PLASTD_j(r,r+1)$ is computed like $PLAST$ (equation 5.43), accounting for the chance the last trip on the segment has of entering the system. That is

$$PLASTD_j(r,r+1) = 1 - CNPI_j(r,r+1) + NPI_j(r,r+1) \quad (5.63)$$

Graphically, each interior summation of expression (5.61) can be seen as a superposition of repeating Figure 5.12, but associated to different origins, all spread out over segment $(r,r+1)$:

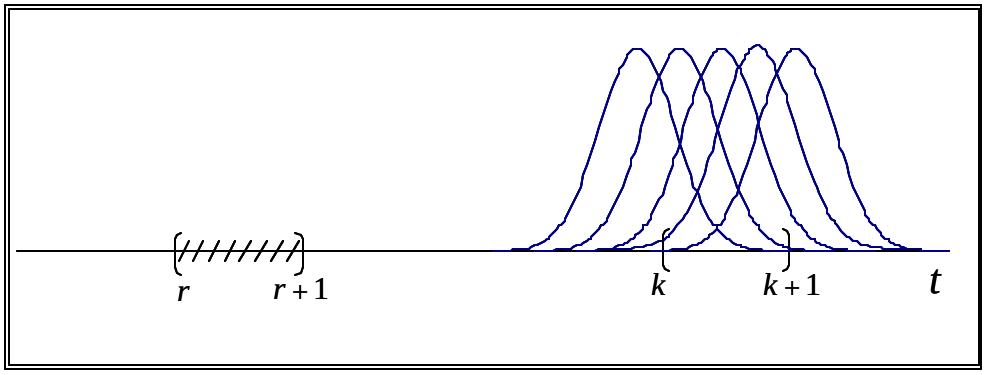

Figure 5.13 Expected number of deliveries on segment $(k, k+1)$ from a specific preceding segment

As mentioned above, in some very special cases it will be necessary to iterate with respect to $E[PI_j(k,k+1)]_A$ in the following way:

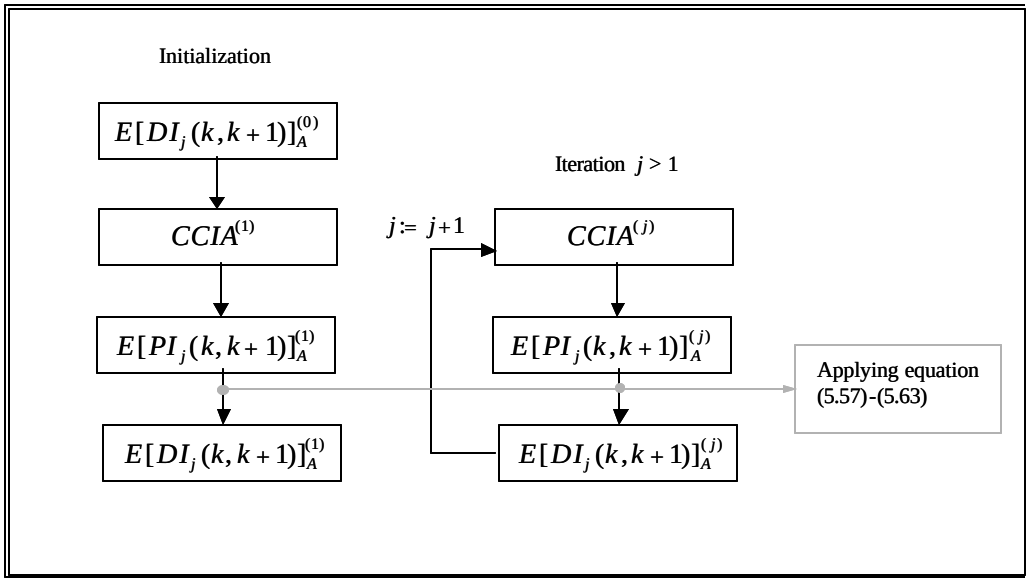

Figure 5.14 Iterative process to compute $E[DI_j(k,k+1)]_A$

The solution should converge to a value very close to zero for the last segment contribution, which is the one that generates the iterative algorithm. Therefore, a good

starting point $E[DI_j(k,k+1)]_A^{(0)}$ could be the estimated expected number of delivery insertions from all segments preceding stop k.

In the next sub-section, a similar methodology is used to treat the sudden reassignments scenario (*S*), for both, the expected number of pick-up and delivery insertions.

5.5.3.2 Sudden reassignments: Scenario *(S)*

For this scenario graphically described in Figure 5.7, the BPA is applied in the same way as done before for the in-advance reassignments case. However, there are some important aspects characterizing this new phenomenon that make the BPA treatment somewhat different when the assignment decisions are taken while vehicle is traveling along the candidate insertion segment:

- There is an extra source of uncertainty when computing the candidate insertions, since they depend on the time the vehicle crosses the segment. The reason is that, in this case the segment itself matches the insertion segment, therefore the segment time will also depend on the insertions on that specific segment, which are obtained from the potential candidate insertions. This linked process will add an iterative adjustment in the computation to be discussed further in this section.
- The scheme depicted in the right side of Figure 5.7 reflects a difference in the computations with respect to scenario (*A*), as a result of the time at which the new request enters the system. Mostly, the new request will find the vehicle moving along the possible insertion segment, therefore, the route-change point will be in general different from the origin stop of the segment, which slightly

modifies the expressions for computing the explanatory variables in the probability insertion formulation (as in expression (5.34)). In this section, the new formulation will be developed in detail.

Given that both processes occur sequentially, for obtaining the formulations associated to scenario (S) it is assumed the expected travel time on segment $(k, k+1)$ resulting from the development of scenario (A) is known with certainty. The notation will be simplified noting that all in-advance variables used in the mathematical development depend and are defined exclusively for segment $(k, k+1)$. That is,

$$E[tS_j(k,k+1)]_A = TA = tS_j(k,k+1) + \overline{\Psi}\, IA \tag{5.64}$$

where $IA = E[I_j(k,k+1)]_A = PIA + DIA = E[PI_j(k,k+1)]_A + E[DI_j(k,k+1)]_A$. At this point, a new variable called "the pick-up ratio in-advance" is defined as follows

$$PRA = \frac{PIA}{IA} \tag{5.65}$$

The pick-up ratio in-advance PRA measures the actual ratio between expected pick-ups and total insertions according to the expected values and probabilities computed in 5.5.3.1. As a result, it is straightforward to define the "delivery ratio in-advance", $PDA = 1 - PRA$.

All calculations performed here will assume that the base travel time for moving from stop k to stop k+1 equals to TA. Following the same arguments provided in the last

section, the base approximate number of candidate insertions on section $(k, k+1)$ for the sudden scenario CIS_b, is computed in the following way

$$CIS_b = \mu_{CD}\ TA \cdot \frac{AR_j(k, k+1)}{A} + E[DI_j(k, k+1)]_S \tag{5.66}$$

Its has been labeled as base, since this expression just considers the travel time resulting from the in-advance scenario, however, the actual travel time is supposed to be greater than *TA* due to the insertion of additional assignments into the vehicle's route in the context of the sudden scenario. That is why the final result has to be corrected iteratively, in the way it will be shown further in this section. Finally, it is worth to mention the inclusion of the expected number of delivery insertions obtained from the extra pick-up insertions occurring in the same segment. Normally, and for cases of segments with a reasonably length, it can be assumed this contribution to be negligible. However, in some extreme cases, where for example a vehicle has currently been assigned only to pick-up one passenger very far from its current location, it could be necessary to take this portion into account, considering the chance of having pick-ups and deliveries happening between the current position and the distant scheduled stop. A simplified treatment for this term will be also discussed further in this section.

In a first approximation, it could be assumed that no delivery insertions are generated from a pick-up insertion occurring within the same segment, in which case (and just as an initial solution), $E[DI_j(k, k+1)]_S = 0$. However, in order to make the analysis more general it should be assumed that $E[DI_j(k, k+1)]_S$ has a value different

from zero. The problem with this generalization is that, unlike scenario (*A*), the computation of this term will depend on the calculation of the expected number of pick-up insertions over the same segment, generating an undefined problem, which requires an iterative solution. This fact will be discussed further in this section. So far, assume $E[DI_j(k,k+1)]_S$ known.

As before, the ceiling of CIS_b is defined to correct the inclusion of non-integer values into the *BPA*, therefore, $CCIS_b = \lceil CIS_b \rceil$. From expression, (5.66), and likewise expression (5.27), it is possible to define the probability of a new insertion to be either a pick-up or a delivery under the new scenario as follows:

$$P_{PS} = \frac{\mu_{CD}\, TA \cdot AR_j(k,k+1)}{CIS_b \cdot A} \qquad P_{DS} = 1 - P_{PS} \tag{5.67}$$

In what follows, the subscript b will be omitted in all variables for the sake of simplicity, however, it is important to keep in mind that the proposed formulation assumes TA to be the expected travel time on segment $(k,k+1)$, which could eventually change according to what it was explained above. Note that conceptual difference between expression (5.67) and the ratio computed in equation (5.65), since they come from different scenarios and the pick-up ratio in-advance is know deterministic and know at the time of start experiencing the sudden scenario.

At this point, it is again necessary to introduce the concept of augmented sequence *CSAS*, constructed from the original sequence CS_j but incorporating all candidate insertions between stops *k* and *k*+1. Thus,

$$CSAS = \{0,1,\ldots,k,k+1,\ldots k+L,\ldots,N_j + LS\} \tag{5.68}$$

where $LS = CCIS_b + 1$ as before. The notation and considerations are analogous to Section 5.5.3.1. The most important difference with the previous scenario is in the treatment of the explicative variables in the definition of the pick-up assignment probability. The calibration procedure does not significantly change under this option, however, in the analytical formulation, the variables have to be defined according to the new scenario. Thus, the assignment probability expressions in equations (5.30) and (5.34), now become

$$P_{S_j}(r,q,s;NI,PINS) = P_{PS} \cdot P_{AS_j} + P_{DS} \tag{5.69}$$

NI and *PINS* are defined as before but only accounting for those insertions happening under the sudden scenario conditions. All in-advance insertions are incorporated separately. The feasibility conditions in this case are defined as follows

$$\begin{aligned} & E[L_j^{AU}(r)] + P_{PS}\left(E[PS]-1\right) \le SPS_j(r,s) + 1 \\ & NITS(k+LS) < MRA \\ & gT(r,q,s) \le 1 \end{aligned} \tag{5.70}$$

The second feasibility condition in (5.70) must incorporate the already decided in-advance insertions, therefore, a new variable has been defined including the previously

reassignments of stop $k+L$ plus IA extra insertions. Then, $NITS(k+LS) = NIT(k+L) + IA$.

$P_{AS_j}(ATTR_j(r,s),0;NI,PINS)$ is the effective pick-up assignment probability under the (S) scenario, under similar conditions as those detailed in the previous section. In this case, $ATTR_j(r,s) = \{SPS_j(r,s), TR_j^{AU}(r), WSP_j^{AU}(r,s)\}$, and $DCT(q) = 0$ for all q considered under this scenario.

A new variable $gT(r,q,s)$ has been introduced, and also a new condition has been added in expression (5.70). Analytically,

$$gT(r,q,s) = \frac{EZt(q) - E[tCL_j^{AU}(r)]}{E[tCL_j^{AU}(s)] - E[tCL_j^{AU}(r)]} \tag{5.71}$$

This variable represents the proportional difference in time of occurrence of new request q with respect to the expected time of arrival at locations defined by the stops r and s. The mentioned extra condition is an additional feasibility condition, stating that no insertions can happen on a segment if the vehicle already passed that segment.

The major difference in this scenario with respect to the previous one, is in the definition of three fundamental variables according to the new prevailing conditions, which are the approximated available space for future pick-ups in the context of the sudden scenario, the cumulative travel time experienced by all customers on board and the difference in time between the expected arrival time of the vehicle at stop k: $SPS_j(r,s)$, $TR_j^{AU}(r)$. The computation of $WSP_j^{AU}(s,r) \approx WSP_j^{AU}(k,k+LS)$ is realized under the same assumptions as those established in the previous section.

The modeling framework is similar to that introduced in (5.34), that is

$$\beta' x_i = \beta_0 + \beta_1 \frac{SPS_j(r,s)}{E[PS]} + \beta_2 TR_j^{AU}(r) + \beta_3 WSP_j^{AU}(k, k+LS) + \beta_4 DCT \qquad (5.72)$$

therefore, the data collection for calibrating the corresponding coefficients will follow the same procedure, independent of the insertion scenario.

In the context of this scenario, we could have a sequence of reassignments as in Figure 5.15. Note that there are two different types of insertions. One occurring in the same way as in scenario (*A*), and a new type of insertion, where the decision point is not a physical stop, but a point along the pre-defined route where the vehicle takes a new path towards the new insertion point dynamically assigned at that time. In Figure 5.15, these decision points (which are not real stops) have been highlighted by green squares.

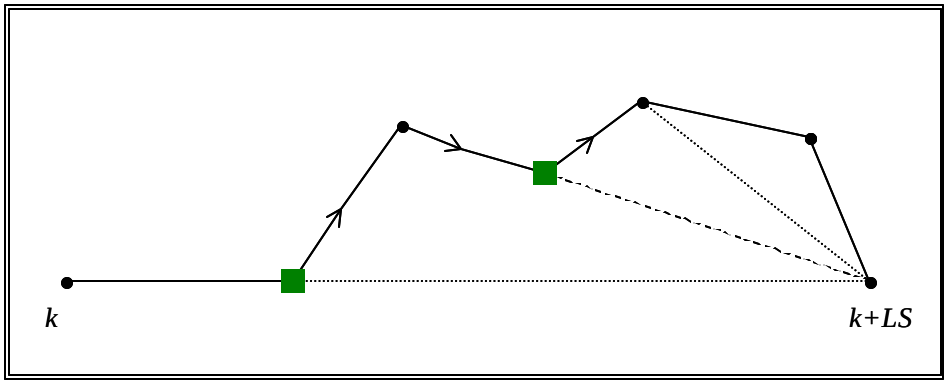

Figure 5.15 Possible sudden scenario insertion

Therefore, two cases can be distinguished in this analysis. Following the usual notation, suppose a new candidate pick-up insertion q (which is really $k+p$ as a member of the sub-sequence *CSAT*) has to be considered for insertion on the already decided stretch (r, s). Then,

Case 1: The new request occurs in time when vehicle is already moving along candidate insertion segment (r, s). Analytically, it happens if $0 \leq gT(r, q, s) \leq 1$. Graphically:

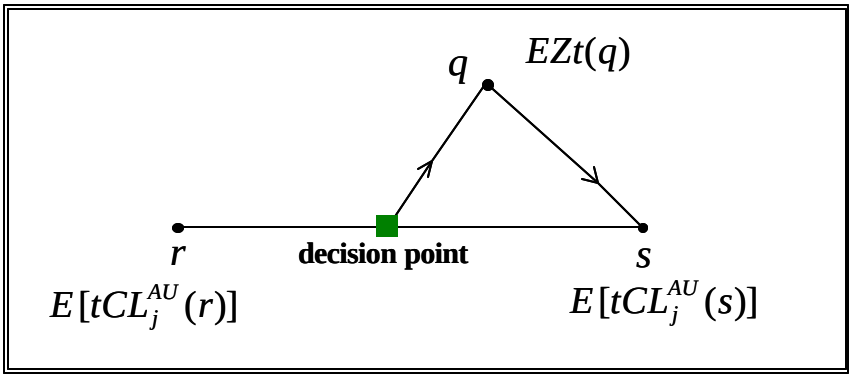

Figure 5.16 Case 1: insertion request

Case 2: The new request occurs in time when vehicle has not yet reached scheduled stop *r*. Analytically, it happens if $gT(r, q, s) < 0$. Graphically:

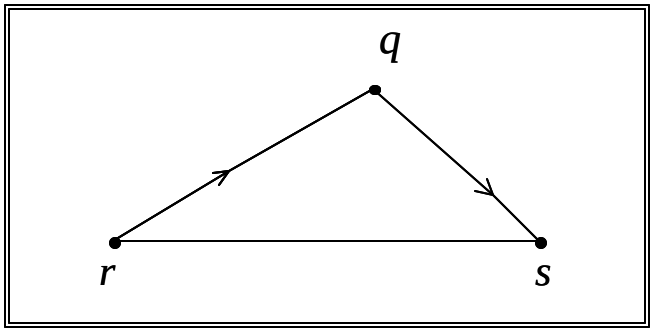

Figure 5.17 Case 2: insertion request

For notation effects, the decision point will be defined as $\bar{r}$ and it will only be considered in the definition of the variables for *Case* 1. Cases where $gT(r,q,s)>1$ are not feasible insertions, therefore, they are excluded from the analysis (see the third condition of expression (5.70)).

Let us first consider the case of $SPS_j(r,s)$. Analytically, the variable will take two different forms depending on the case.

If <u>*Case 1*</u>,

$$\begin{aligned} SPS_j(\bar{r},s) = & \left[L_{MAX} + P_{DS}\left(NI - PINS\right) + \right. \\ & + (1-PRA)\left[(1-gT(r,q,s))\cdot IA(r,s) + IA(s,k+LS)\right] \\ & - E[L_j^{AU}(k)] - PINS\,P_{PS}\,E[PS] - PRA\left(IA(k,r) + gT(r,q,s)\cdot IA(r,s)\right)E[PS] \\ & \left. - \left[LRES_j^{AU}(k+LS)\right]^+\right]^+ \end{aligned} \tag{5.73}$$

Simplifying terms, (5.73) becomes

$$\begin{aligned} SPS_j(\bar{r},s) = & \left[L_{MAX} - IA(k,r)\,PRA\;E[PS] + \right. \\ & + IA(r,s)\left[1 - gT(r,p,s) - PRA + PRA\cdot gT(r,q,s) - PRA\cdot gT(r,q,s)\,E[PS]\right] + \\ & + IA(s,k+LS)[1-PRA] + P_{DS}\cdot NI - PINS\left(P_{DS} + P_{PS}\,E[PS]\right) \\ & \left. - E[L_j^{AU}(k)] - \left[LRES_j^{AU}(k+LS)\right]^+\right]^+ \end{aligned} \tag{5.74}$$

In (5.73) and (5.74), $IA(m,n)$ represents the expected in-advance insertions occurring between stops *m* and *n*, and is computed depending on the pair of stops involved at any case. That is,

$$IA(r,s) = \frac{IA}{NI+1} \tag{5.75}$$

$$IA(k,r) = \frac{IA}{NI+1} PINS \tag{5.76}$$

$$IA(s,k+LS) = \frac{IA}{NI+1}(NI - PINS) \tag{5.77}$$

Note that $IA(k,r) + IA(r,s) + IA(s,k+LS) = \frac{IA}{NI+1}(PINS + 1 + NI - PINS) = IA$.

In the particular case when $E[PS] = 1$, (5.74) becomes

$$\begin{aligned} SPS_j(\bar{r},s) = &\left[L_{MAX} - IA(k,r)\,PRA + IA(r,s)\left[1 - gT(r,q,s) - PRA\right] + \right. \\ &+ IA(s,k+LI)[1-PRA] + P_{DS} \cdot NI \\ &\left. - PINS - E[L_j^{AU}(k)] - \left[LRES_j^{AU}(k+LS)\right]^+ \right]^+ \end{aligned} \tag{5.78}$$

since $P_{PS} + P_{DS} = 1$.

On the other hand, if <u>*Case 2*</u> the final expression for this variable is much simpler than that obtained for <u>*Case 1*</u>. In fact,

$$\begin{aligned} SPS_j(r,s) = &\left[L_{MAX} + P_{DS}\left(NI - PINS\right) + \left(1 - PRA\right) \cdot IA(s,k+LI) \right. \\ &- E[L_j^{AU}(k) - PINS \cdot P_{PS}\,E[PS] \\ &\left. - PRA\left(IA(r,s) + IA(k,r)\right)E[PS] - \left[LRES_j^{AU}(k+LS)\right]^+ \right]^+ \end{aligned} \tag{5.79}$$

or, by rearranging terms,

$$SPS_j(r,s) = \Big[L_{MAX} - IA(k,r)\,PRA \cdot E[PS] - IA(r,s)\,PRA \cdot E[PS] + \\ + IA(s,k+LS)(1-PRA) + P_{DS} \cdot NI - PINS\,(P_{DS} + P_{PS}\,E[PS]) \\ - E[L_j^{AU}(k)] - \left[LRES_j^{AU}(k+LS)\right]^+\Big]^+ \tag{5.80}$$

and, when $E[PS]=1$, (5.80) turns to

$$SPS_j(r,s) = \Big[L_{MAX} - IA \cdot PRA + IA(s,k+LS) + P_{DS} \cdot NI \\ - PINS - E[L_j^{AU}(k)] - \left[LRES_j^{AU}(k+LS)\right]^+\Big]^+ \tag{5.81}$$

In general, final expressions (5.74) and (5.80) establish the value of $SPS_j(r,s)$ in both possible insertions cases. Note that the estimation of the available space for new candidate pick-ups considers the joint effect of both scenarios, weighted by the corresponding factors at any case. On the one hand, in case of the in-advance scenario, the expected number of insertions and the pick-up ratio weight are assumed to be known. On the other hand, the sudden scenario effect is added through the new candidate insertions and the pre-defined probabilities, generating the above described analytical expressions.

The corresponding *BPA* formulation for the sudden scenario, to be developed further in this section, will strictly determine which case corresponds to each candidate insertion under analysis.

The second variable $TR_j^{AU}(r)$ has also a different treatment depending on the insertion case.

Thus, if *Case 1*, then

$$TR_j^{AU}(\bar{r}) = TR_j^{AU}(k) + \frac{E[tS_j^{AU}(k,r)]}{2} E[PS] \cdot \{P_{PS}\, PINS + PRA\, IA(k,r)\} + \\ + \frac{gT(r,q,s)^2}{2} E[tS_j^{AU}(r,s)]\, PRA \cdot IA(r,s) \cdot E[PS] \tag{5.82}$$

where the expected travel time between stops k, r and r,s is computed accordingly, that is

$$E[tS_j^{AU}(k,r)] = \left(\frac{TA + \overline{\Psi} \cdot NI}{NI+1} \right) PINS \tag{5.83}$$

$$E[tS_j^{AU}(r,s)] = \alpha_S = \left(\frac{TA + \overline{\Psi} \cdot NI}{NI+1} \right) \tag{5.84}$$

If *Case 2*,

$$TR_j^{AU}(r) = TR_j^{AU}(k) + \frac{E[tS_j^{AU}(k,r)]}{2} E[PS] \cdot \{P_{PS} \cdot PINS + PRA \cdot IA(k,r)\} \tag{5.85}$$

Before computing the third variable $DCT(q)$, it is necessary to compute an analytical expression for the factor $gT(r,q,s)$ in the context of this formulation. Recalling the definition of the sub-sequence $TCAT = \{k+1, k+2, \ldots, k+LS-1\}$ and the definition of α_S in (5.84), it is easy to find an expression for $gT(r,q,s)$ from (5.71). α_S represents an approximated measure of the vehicle travel time between two consecutive stops on segment $(k, k+LS)$, in the same way that α_A was defined for the in-advance scenario in the previous section.

It is straightforward to show that the denominator in expression (5.71) is equal to α_S. In fact,

$$E[tCL_j^{AU}(r)] = E[tCL_j^{AU}(k)] + \left(\frac{TA + \overline{\Psi} \cdot NI}{NI + 1}\right) PINS$$

and

$$E[tCL_j^{AU}(s)] = E[tCL_j^{AU}(k)] + \left(\frac{TA + \overline{\Psi} \cdot NI}{NI + 1}\right) (PINS + 1)$$

therefore

$$E[tCL_j^{AU}(s)] - E[LCt_j^{AU}(r)] = \left(\frac{TA + \overline{\Psi} \cdot NI}{NI + 1}\right) = \alpha_S \qquad qed \qquad (5.86)$$

In addition $q \in TCAT$, therefore it can be written as $q = k + p$, with $1 \le p \le LS - 1$. Then, by assuming that the candidate calls are drawn from a uniform distribution in time, it is a direct result that

$$EZt(k + p) = \left(\frac{p}{LS}\right) TA \qquad (5.87)$$

therefore, by replacing back (5.86) and (5.87) into (5.71), the following expression for $gT(r, k + p, s)$ is obtained:

$$gT(r, k + p, s) = \left(\frac{p}{LS}\right) \frac{TA}{\alpha_S} - PINS \qquad (5.88)$$

Expression (5.88) will condition the case, whether it is *Case 1* or *Case 2*, depending on the assumed time at which a random call enters the system, and also, depending upon the

characteristics of sub-segment (r,s) where the customer is considered for insertion. Recalling the definition of each case, it is possible to rewrite the original feasibility conditions as follows:

<u>*Case 1*</u>: $0 \le \frac{p\,TA}{LS\,\alpha_S} - PINS \le 1$

or equivalently $\frac{p}{LS} TA \ge \alpha_S\, PINS \quad \wedge \quad \frac{p}{LS} TA \le \alpha_S\,(PINS+1)$. The interpretation of this condition is very simple. Basically, the time of generation of the new request $k+p$ (left-hand side term in both inequalities) has to occur after vehicle *j* arrives at stop *r* and before it leaves stop *s* (right-hand side term of the first inequality for *r* and of the second one for *s*).

<u>*Case 2*</u>: $\frac{p\,TA}{LS\,\alpha_S} - PINS < 0$

or equivalently $\frac{p}{LS} TA < \alpha_S\, PINS$. This condition states that in <u>*Case 2*</u> the new request has to be generated before than vehicle *j* arrives to stop *r*.

<u>*Infeasible Case*</u>: $\frac{p\,TA}{LS\,\alpha_S} - PINS > 1$

or equivalently $\frac{p}{LS} TA > \alpha_S\,(PINS+1)$, meaning that the new request enters the system after vehicle *j* already leaves stop *s*.

At this point, it is possible to formulate the *BPA* structure in the same way it was done for the (*A*) scenario.

Scenario (*S*): Expected number of pick-up insertions calculation using a branched process approach (BPA)

The expression for computing the expected pick-up insertions on a segment $(k, k+LS) \in TCS$ under this scenario, assuming that the in-advance scenario has already been decided, follows the same structure as before. Analytically,

$$E[PI_j(k,k+1)]_S = E[PI_j^{AU}(k,k+LS)]_S = \Lambda S_j(k,k+LS;k+1) \tag{5.89}$$

where

$$\Lambda S_j(k,k+LS;k+p) = \begin{cases} \Lambda SB_j(k,k+LS;k+p) & \text{if } p < LS \\ 0 & \text{otherwise} \end{cases} \tag{5.90}$$

and

$$\begin{aligned} \Lambda SB_j(k,k+L;k+p) = {} & P_{S_j}(k,k+p,k+LS;0,0)\,\Lambda S_j(k,k+p,k+L;k+p+1) + \\ & + \{1 - P_{S_j}(k,k+p,k+L;0,0)\}\Lambda S_j(k,k+L;k+p+1) \end{aligned} \tag{5.91}$$

Notice that, as before, the initial value for *NI* and *PINS* is equal to 0, since they represent the previous insertions under the sudden scenario only. Those insertions decided for the in-advance scenario were incorporated in all the expressions described above in this section.

Define $S(u,v) = \{ s(u), s(u+1), s(u+2)\ldots, s(v)\}$ as the sequence of already decided insertions occurring between k and $k+L$ in some order determined by the definition of $s(i)$ and associated to some branch. Then, in the general case, assuming that a sequence $S(1,m)$ has been already accepted:

$$\Lambda S_j(k,S(1,m),k+LS;k+p) = \begin{cases} \Lambda SB_j(k,S(1,m),k+LS;k+p) & \text{if } p < LS \\ m P_{PS} & \text{if } (p=LS)\wedge(k+LS-1\notin S(1,m)) \\ \{m-1+P_{LASTS}\}P_{PS} & \text{if } (p=LS)\wedge(k+LS-1\in S(1,m)) \end{cases} \quad (5.92)$$

and

$$\begin{aligned} &\Lambda SB_j(k,S(1,m),k+LS;k+p) = \\ &= \frac{1}{m+1}\Big\{ P_{S_j}(k,k+p,s(1);m,0)\,\Lambda S_j(k,k+p,S(1,m),k+LS;k+p+1) + \\ &+ \left[1-P_{S_j}(k,k+p,s(1);m,0)\right]\Lambda S_j(k,S(1,m),k+LS;k+p+1) + \\ &+ \sum_{r=1}^{m-1}\Big(P_{S_j}(s(r),k+p,s(r+1);m,r)\,\Lambda S_j(k,S(1,r),k+p,S(r+1,m),k+LS;k+p+1) + \\ &+ \left[1-P_{S_j}(s(r),k+p,s(r+1);m,r)\right]\Lambda S_j(k,S(1,m),k+LS;k+p+1)\Big) + \\ &+ P_{S_j}(s(m),k+p,k+LS;m,m)\,\Lambda S_j(k,S(1,m),k+l,k+LS;k+p+1) + \\ &+ \left[1-P_{S_j}(s(m),k+p,k+LS;m,m)\right]\Lambda S_j(k,S(1,m),k+LS;k+p+1)\Big\} \end{aligned} \quad (5.93)$$

As before, expression (5.93) has now two different options to close the recursion. The problem has been split into two cases in order to correct the fact that CIS_b is not necessarily an integer, motivating the definition of a new variable $CCIS_b$ in order to generate a countable number of events in the BPA scheme. The way to correct this difference is by assuming that the first *LS-2* candidate calls enter the system with probability equal to 1, however the last candidate request *k+LS-1* could eventually not

appear while vehicle is moving along segment $(k, k+LS)$. It is assumed it has a probability of occurrence equals to $P_{LASTS} = 1 - CCIS_b + CIS_b$.

Notice that if m=1, equation (5.93) still works. However, in such a case $S(1,m) = S(1,1) = s(1)$. In this case, the summation does not make any sense, and therefore, it has to be discarded from the calculations.

It is clear that in this case there is no significant difference with the in-advance scenario. The major difference is in the extra complexity in the computation of the variables on the right-hand side of expression (5.72), and also that the in-advance insertions already decided are now part of the base system for this scenario, mostly influencing vehicle travel times.

For example, suppose that in expression (5.93) it is necessary to compute $P_{S_j}(k+1, k+2, k+3; 1, 1)$. Assume that there are one in-advance insertion scheduled and $CIS_b = 2$, which is equivalent to $LS = 3$. In addition, the base travel time is equal to 15 minutes and $\overline{\Psi} = 5$ min. In this case, $TA = 15 + 5*1 = 20$ min. considering the existent in-advance insertion.

Thus,

$$gT(k+1, k+2, k+3) = \frac{2*20}{3*12.5} - 1 = 1.0666666 - 1 = 0.0666666 \qquad (5.94)$$

since $\alpha_S = \frac{20+5*1}{2} = 12.5$. Therefore, this insertion corresponds to a *Case 1* insertion, and then, the explicative variables have to be computed using formulas (5.74) and (5.82) accordingly.

Scenario (*S*): Expected number of delivery insertions calculation

In case of computing the expected number of delivery insertions occurring within the same segment it is necessary first to estimate the expected number of pick-up insertions under the sudden scenario conditions. After that, the same methodology applied for the in-advance scenario has to be used, summarized in equations (5.57)-(5.63), but considering only the pick-up insertions generated within the same segment.

Analytically, defining

$$NPIS_j(k,k+1) = E[IS]E[PI_j(k,k+1)]_S \tag{5.95}$$

then

$$\Delta ts_j(k,k+1) = \frac{E[tCL_j(k+1)] - E[tCL_j(k)]}{CNPIS_j(k,k+1)} \tag{5.96}$$

where

$$CNPIS_j(k,k+1) = \lceil (NPIS_j(k,k+1)) \rceil \tag{5.97}$$

using (5.60), the following expression is obtained for the expected number of pick-up insertions on segment $(k,k+1)$ under the (*S*) scenario:

$$E[DI_j(k,k+1)]_S = \sum_{i=1}^{CNPIS_j(k,k+1)} \lambda_j(E[tCL_j(k)] + \Delta ts_j(k,k+1); k,k+1)\ \Gamma_j(i;k,k+1) \tag{5.98}$$

where

$$\Gamma_j(i;k,k+1) = \begin{cases} 1 & \text{if } i < CNPIS_j(k,k+1) \\ PLASTDS_j(k,k+1) & \text{otherwise} \end{cases} \tag{5.99}$$

and

$$PLASTDS_j(k,k+1) = 1 - CNPI_j(k,k+1) + NPI_j(k,k+1) \tag{5.100}$$

As mentioned above, in most cases this contribution can be considered equal to zero without any discussion ($\lambda_j \to 0$), however in some special cases, this effect cannot be discarded *a priori*, specially when the segment under analysis is long enough, or simply when there is no other alternative of insertion left after the corresponding pick-up. However, the objective of this methodology is not to use the expected number of unknown insertions directly, but to estimate a realistic travel time due to possible future insertions, in which case, the contribution of this part is not considered significant.

In any case, solving this problem requires an iterative approach summarized in the next chart:

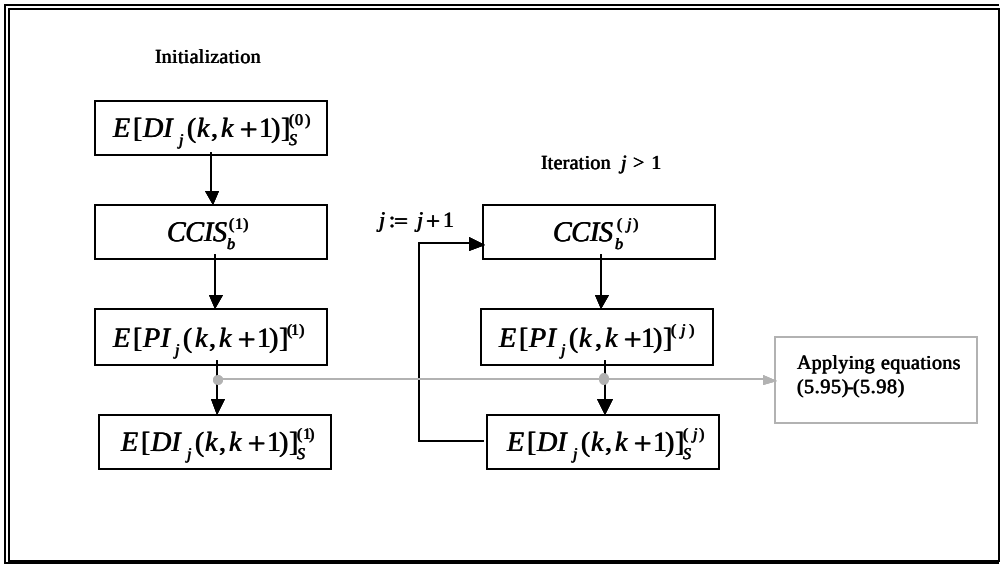

Figure 5.18 Iterative process to compute $E[DI_j(k,k+1)]_S$

In this case, the solution should converge to a value very close to zero in most cases for $E[DI_j(k,k+1)]_S$, therefore, a value $E[DI_j(k,k+1)]_S^{(0)} = 0$ is probably the most reasonable starting point for running this algorithm.

In the next sub-section a second iterative correction is addressed in order to find the total expected number of pick-up insertions under scenario (*S*).

Scenario (*S*): Correction factor in the estimation of the expected number of pick-up insertions

There is one last correction left to address in this chapter. Notice that the expected number of pick-up insertions on segment $(k,k+1)$ obtained in expression (5.89) depends on the total number of candidate requests occurring whereas vehicle j is moving through the segment, which was defined as CIS_b. By observing expression (5.66), it can be seen that CIS_b depends upon the base travel time TA and on the expected number of delivery insertions $E[DI_j(k,k+1)]_S$. The issue is that the more candidate insertions are considered, the more actual insertions should happen on the segment. That increase in number of insertions should eventually raise the travel time of the vehicle on that segment, generating, as a result, more candidate requests. These extra candidate requests could eventually increase the actual number of insertions under scenario (*S*). The feasibility constraints should stabilize the system converging to certain value.

The objective of this correction is to check if the expected number of insertions on a segment (mainly pick-ups) remains invariant due to sudden reassignments happening dynamically when a transit vehicle is moving between two scheduled stops of

its route. The aforementioned iterative process, associated to this correction, can be represented by the following two charts:

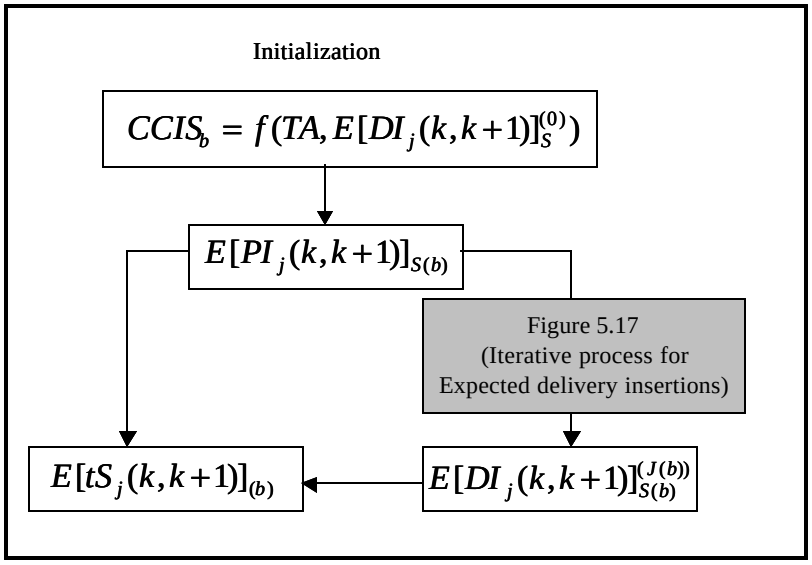

Figure 5.19 Iterative process to compute $E[I_j(k,k+1)]_S$: initialization

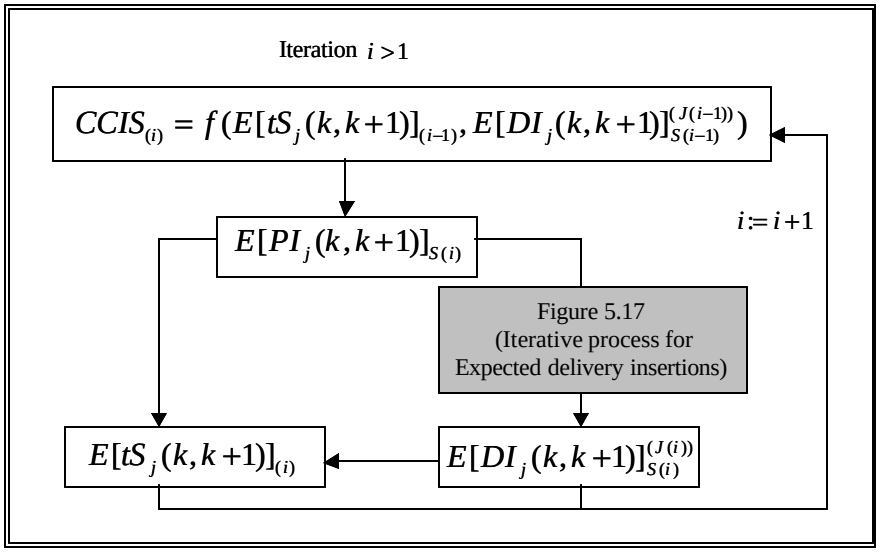

Figure 5.20 Iterative process to compute $E[I_j(k,k+1)]_S$: iteration *i*

Figure 5.19 basically represents the process summarized throughout the whole Section 5.5.3.2. From an initial rate of demand, a known base travel time obtained after considering the in-advance scenario, and an initial guess about the expected number of pick-up insertions from the sudden scenario, it is possible to compute the expected number of pick-up insertions on segment $(k, k+1)$ belonging to vehicle j's route, using a BPA procedure from an adaptive predictive control basis. With the expected number of pick-up insertions, the iterative process in Figure 5.18 can be run for determining the expected number of delivery insertions from the sudden scenario. Finally, from the total number of expected insertions, the expected vehicle travel time on that segments can be estimated. Figure 5.20 shows haw to adjust this calculation by iterating on the candidate insertions till the expected vehicle travel time becomes stable (almost invariant). In Chapter 7 some numerical experiments for analyzing the pattern of convergence of these algorithms will be carried out under different supply and demand conditions.

5.6 Stochastic rerouting delay for dynamic vehicle routing: data collection and model calibration

In Appendix to Chapter 5, a brief discussion about discrete choice theory and estimation methodology is presented, taken from Maddala (1985). As mentioned in Section 5.5, it has been assumed that the process of vehicle-customer assignment can be described as a discrete regression model. By discrete regression model, we mean those models in which the dependent variable assumes discrete values. In this case, it is also the simplest of

these models, in which the dependent variable y is binary (it can assume only two values, which for convenience and without any loss of generality, are denoted 0 and 1).

In this particular modeling framework, y can be defined as 1 if the dispatching module decides to assign a combination vehicle-segment-customer, 0 otherwise. The right-hand side variables, or independent variables, were chosen as the ones that should better explain the minimizing cost behavior of the dispatching module when taking assignment decisions, given the feasibility constraints of the problem discussed in Chapter 4.

In addition, it was decided to try two different functional forms for the cumulative distribution function, determined by the assumptions made about the error terms of the linear regression relationship (see Appendix to Chapter 5 for more details): the binary logit and binary probit models.

Let us illustrate the meaning of the probability of accepting a new pick-up assignment q, on stretch $(k, k+1)$, defined as $P_{AS_j}(ATTR_j(k,k+1), DCT(q))$, assuming that the cumulative distribution of the error (μ_i) is the logistic, obtaining the logit model. In general

$$P_{AS_j}(ATTR_j(k,k+1), DCT(q)) = \text{Prob}(y_i = 1) = \text{Prob}(\mu_i > -\beta' x_i) = 1 - F(-\beta' x_i) \quad (5.101)$$

where

$$\beta' x_i = \beta_0 + \beta_1 \frac{SP_j(k,k+1)}{E[PS]} + \beta_2 TR_j(k) + \beta_3 WSP_j(k,k+1) + \beta_4 DCT(q) \quad (5.102)$$

In this case, under the logit model assumptions

$$F(-\beta' x_i) = \frac{exp(-\beta' x_i)}{1+\exp(-\beta' x_i)} = \frac{1}{1+\exp(\beta' x_i)} \tag{5.103}$$

hence,

$$P_{AS_j}(ATTR_j(k,k+1), DCT(q)) = 1 - F(-\beta' x_i) = \frac{\exp(\beta' x_i)}{1+\exp(\beta' x_i)} \tag{5.104}$$

In this case, a closed form for (5.104) is available. The estimation of the coefficients and the corresponding tests of significance (to be evaluated numerically in Chapter 7) are detailed in appendix 5.

An important issue is how to collect the necessary data from the real operation of transit vehicles, or in this case, through simulation, in order to estimate the β coefficients properly.

Notice that from the system operation, all independent variables in (5.102) are measurable at any decision time. Therefore, every time a new pick-up request enters the system (and therefore, every time the dispatching module has to take an assignment decision), it is possible to observe the insertion decision and collect the corresponding data necessary to estimate model (5.101)-(5.104). In addition, some extra data can be measured for updating system indicators, such as, the expected pool size $E[PS]$ per pick-up stop, the expected waiting time at stop locations, the expected value and the variance of the length of internal trips, etc.

For instance, suppose there are four vehicles moving around as in Figure 5.20. Suddenly, when all vehicles are in the position indicated by the green squares already assigned to serve the scheduled routes showed in the figure, a new call q is received

asking for immediate service (with destination at spot q_d). Then, the dispatching module decides that the best insertion is the indicated in the figure, along the route of vehicle 1.

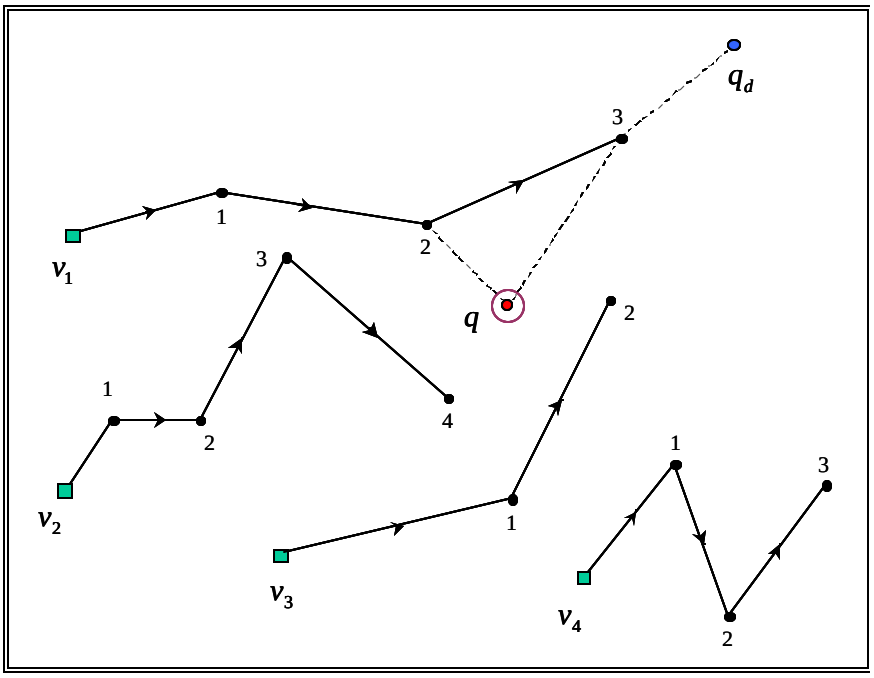

Figure 5.21 Example of vehicle-segment-customer assignment decision

For the dispatching module to take this kind of assignment decision, he(she) is required to have information about the state of the system at that time in order to evaluate the incremental cost of each insertion according to the modeling framework described in Chapter 4.

At this point, the system will record the data corresponding to this assignment in the following format:

Table 5.1 Example of vehicle-segment-customer assignment data for model calibration

Vehicle (#)	$SP_j(k,k+1)$	$TR_j(k)$	$WSP_j(k,k+1)$	$DCT(q)$	*y*
1	$SP_1(0,1)$	$TR_1(0)$	$WSP_1(0,1)$	$Dt_1(0,q)$	*0*
1	$SP_1(1,2)$	$TR_1(1)$	$WSP_1(1,2)$	$Dt_1(1,q)$	*0*
1	$SP_1(2,3)$	$TR_1(2)$	$WSP_1(2,3)$	$Dt_1(2,q)$	*1*
2	$SP_2(0,1)$	$TR_2(0)$	$WSP_2(0,1)$	$Dt_2(0,q)$	*0*
2	$SP_2(1,2)$	$TR_2(1)$	$WSP_2(1,2)$	$Dt_2(1,q)$	*0*
2	$SP_2(2,3)$	$TR_2(2)$	$WSP_2(2,3)$	$Dt_2(2,q)$	*0*
2	$SP_2(3,4)$	$TR_2(3)$	$WSP_2(3,4)$	$Dt_2(3,q)$	*0*
3	$SP_3(0,1)$	$TR_3(0)$	$WSP_3(0,1)$	$Dt_3(0,q)$	*0*
3	$SP_3(1,2)$	$TR_3(1)$	$WSP_3(1,2)$	$Dt_3(1,q)$	*0*
4	$SP_4(0,1)$	$TR_4(0)$	$WSP_4(0,1)$	$Dt_4(0,q)$	*0*
4	$SP_4(1,2)$	$TR_4(1)$	$WSP_4(1,2)$	$Dt_4(1,q)$	*0*
4	$SP_4(2,3)$	$TR_4(2)$	$WSP_4(2,3)$	$Dt_4(2,q)$	*0*

$Dt_j(k,q) = E[tCL_j(k)] - EZt(q)$ is obtained from the system data, depending on the time of the new call and on the expected arrival time to each vehicle-stop.

The number of valid observations will depend on the limits assigned to the segment catchment area associated to each route segment. In the example above, the shaded rows seem to be the reasonable observations to be added to the sample for calibrating (5.104).

With enough data, the modeler could decide to update the model of equation (5.104) if the real measured travel times do not match the expected segment travel times according to the methodology developed in this chapter.

5.7 Final remarks

The objective of this chapter is to formulate in detail a novel methodology proposed in the context of this dissertation, in order to correctly compute the expected vehicle travel time when solving a dynamic and stochastic pick-up and delivery problem. The problem is dynamic since the customers entering the system are not known in advance, and therefore the system is allowed to dynamically adapt its components due to changes in demand patterns, and it is also stochastic because of the random nature (in time and space) of the demand generation process.

After developing the methodology summarized in Chapters 4 and 5 to treat this kind of assignment problems, there are some final comments with regard to the limitations and features of the previous formulations to be addressed in this section:

- The expected vehicle travel time will mainly depend upon the expected number of insertions about to happen on a predefined vehicle route. After analyzing the mathematical treatment given to this approach, it is apparent that the expected number of insertions on a predefined segment of a vehicle route depends not only on the candidate insertion segment length, but also on how far is the segment from the current position of the vehicle. This is because of the nature of the in-advance scenario where the insertions can occur on a portion of the vehicle route not reached at the time the dispatching module take the decision of reassignment. This dependence incorporated into the cost models in Chapter 4 is not very evident without understanding the role of both insertion scenarios as part of the reassignment process – already described throughout this chapter.

- The constraints applied to the system operations could strongly influence the results obtained after applying this approach, not only through model calibration-estimation but also due to the constraints on the maximum number of reassignment allowed. These numerical limits ensure stability of vehicle reassignments, avoid infinite deferment and allow the modeler to control the system based on the results of the operation and previous experience (historical data, previous knowledge, etc.)
- Adapting these formulations to the *HCPPT* system proposed in Chapters 3 and 4 is straightforward and it does not require modification of the formulation and algorithms proposed here. The only difference is the addition of the last segment of the reroutable portion of vehicle routes (connecting this portion with the non-reroutable or express corridor towards the adjacent hub), which can be treated as a normal segment considering the assumptions and modeling details addressed in Chapter 4.

6 SIMULATION OF *HCPPT* IN URBAN TRANSPORTATION NETWORKS

6.1 Introduction

The major topic of this chapter is the development of a multiple-class simulation platform as a way to evaluate the *HCPPT* system serving real transit demand under realistic network traffic conditions. Simulating the real-time routing schemes and the dispatching rules based on real-time stochastic control described in Chapters 4 and 5 requires modeling the network congestion dynamics and thus modeling of realistic traffic details is necessary. Needless to say, none of the existing simulation software allows simulation of such a system. In fact, even the simulation of something very basic as a streetcar system or a paratransit service is not an option in most existing simulation software packages. A cursory study itself would reveal that simulation of any vehicle class other than personal autos is always developed as an afterthought. In most cases the simulation developers have done only a superficial addition of transit simulation on top of detailed simulation of autos and control mechanisms on freeways and arterials.

However, there are considerable new developments taking place in urban transportation networks, where real-time routing is being increasingly used for several fleets - both transit and commercial - due to the advent of wireless and GPS technology. On the other hand, there are no modeling tools available for studying dispatching rules or routing policies/alternatives for real-time routing for new transit schemes such as ADART (Dial, 1995) and *HCPPT*, or for the case of emergency services. Newer but

more conventional ideas such as Bus Rapid Transit (BRT) are also receiving attention thanks to some large-scale initiatives from FTA. Again, methods to research their performance characteristics are scant, though some researchers have attempted to use simulation for these purposes.

There are reasons why simulation could become a useful (and perhaps the only) option to study many of such systems. First, the systems do interact with auto traffic and influence the network performance, or are at least influenced by it. On shared right of way with auto traffic, buses, streetcars or LRT can influence the supply characteristics at least in the heavier transit corridors. Control schemes such as transit preemption or signal preemption by emergency vehicles affect the network conditions. In all cases of shared right of way, the auto traffic (with the well known non-linear congestion characteristics they collectively produce) influences the movement of these other classes of vehicles. Such interactions are difficult to model with abstract mathematical models other than in simplistic academic contexts, and simulation may be the only option. On the performance side, even the traditional capacity analysis schemes may use simulation schemes in the future.

There are also expectations that microsimulations would be the way of the future, which is perhaps more due to their flexibility and detail-oriented design. This makes it easier for developers to at least claim to have incorporated capabilities (albeit often with insufficient insights or care and without well-tested fundamental models) to simulate various aspects of urban transportation networks that practitioners and researchers would like to study.

The above discussion points to the need for a careful look at modeling of urban networks in completeness, properly accounting for various vehicle classes and modal details. It is true that there cannot be any model that can simulate everything in an urban network; however the state-of-the art needs to improve significantly. This chapter summarizes the contribution made in this direction in the context of this dissertation.

In effect, the work in this chapter develops a general purpose simulation framework for microscopic simulation of multiple classes of vehicles, with particular attention to fleets. The model developed can be used for purposes other than for *HCPPT* simulation, including police fleets, ambulance fleets etc. Such systems include vehicles that travel from origins and destinations determined "on the fly" during the simulation and thus the traditional simulators have serious deficiencies in their modeling. The primary problem is in the prevalent method of viewing the vehicles to be traveling from origin zones to destination zones, a historical legacy from the transportation planning paradigms, with serious limitations in real-time dynamic contexts for non-auto fleets. The framework and approaches developed here are expected to lead to considerable other developments in microscopic fleet simulations in the future.

The chapter begins with a background section that focuses on broad classifications and some of the existing simulation approaches, and proceeds to suggest possible modeling schemes. Next, in Sections 6.3 to 6.6, the discussion is focused in showing the needs and requirements of comprehensive multiple-class simulations and in pointing out the complexities in the modeling. In Section 6.7, the details of such a simulation scheme applied to the *HCPPT* system are addressed and explained at length.

6.2 Classification of simulation types

Simulation models can be categorized according to the nature of the system that they are trying to represent, and some of these concepts have been well-known in the general systems simulation area for decades. A discrete system is one in which the state of the system changes at discrete points in time. One can think of car arrivals at a certain point, occurring at distinct points of time, as events. Between two consecutive items nothing happens; that is, the state of the system remains unchanged. When the number of these events is finite, the simulation is known as discrete event simulation. A continuous simulation is one where the state of the system changes continuously over time. Water level in a dam may be thought of as a system where it changes continuously over time according to some known differential equations describing its state. Note that a discrete event simulation could be used as an approximation to continuous time simulation in many cases. Most systems in the real world are both discrete and continuous but usually one predominates over the other (Law and Kelton, 2000). Traditional traffic flow descriptions have been based on continuous speed and distance variables. As far as the personal auto traffic is concerned, continuous simulation is the only possibility, as the system performance (in terms of speeds, flow, and density) is the result of continuous interactions between vehicles. In other words, a continuous simulation for personal auto traffic becomes a requirement, as it is a "collective" system with continuous response. On the other hand, the simulation of control hardware operations (signals) as well as some of the fixed schedule transit systems has been done using event-based simulation

In recent years, object-oriented or agent based simulations have been proposed to be useful in depicting traffic movement. Traffic can be viewed as a complex system

composed of various entities interacting with each other. Through agent-based simulations, relatively complex global phenomena can be expressed as a sum of small, localized interactions among the agents in the system. The main entities in the traffic network are road segments, vehicles, traffic signals etc., which can be modeled as agents. These agents have the ability to perceive the changes in their environment. Based upon this perception, an agent can modify its behavior to achieve a certain goal. For instance, a vehicle on the road can sense its neighboring vehicles and can change its speed or acceleration, which is analogous to "behavior". Thus, this car-agent behaves as a real-life driver who wants to reach his destination while attaining certain goals. An example of one such goal might be to reach the destination in the shortest time possible without violating any speed limits. An advantage of such an environment is that it lends itself naturally to distributed computing. Since each agent is in full control of itself, the whole simulation can be divided into individual pieces in an intuitive fashion, which can then be simulated by different processors. It must be noted that some of the recent literature on "agent based simulations" in traffic is only referring to the object-oriented design of the software, and the fundamental interactions simulated are not exactly true to the definition of agent interactions. Admittedly, the definitions in the literature are not rigid, however.

Classification of simulation based on modeling the personal autos:

Simulation models can be categorized according to the level of detail at which automobile traffic is represented. This is essentially a classification based on the modeling of one of the classes of vehicles of interest to us, however it is significant

because it is the dominant class of vehicles in most urban contexts. Furthermore, this classification is one that will conceivably influence the development of more comprehensive simulation environments in the future. The categories are well known and the literature on auto-traffic simulation models is vast. A brief overview is provided for completeness.

Macroscopic simulation models deal with a group of vehicles rather than treating vehicles at the individual level. General flow relationships, applicable to fluids, can be applied in these models to arrive at the condition of traffic at any given time. This level of aggregation can be usually found in static planning models of typically large areas. Being static, these models do not respond to changes in traffic conditions over a short period of time and hence, are fairly limited in their application. Many of the well known macroscopic applications were developed in the late 1960's or the early 1970's. Examples of such models include TRANSYT, FREQ, FREFLO, META, SIMAUT and several special-purpose research simulation programs.

A slightly higher level of representation is provided in mesoscopic simulation models. These models are able to handle small changes in the traffic patterns over a short period of time, which can be of the order of a few seconds. The level of detail in these simulations may change over time depending on traffic conditions. DYNASMART, which uses a scheme based on macroscopic traffic flow relations but with individual vehicle tracking, is sometimes added in this category. Other examples include SIMNET, SATURN, PREDICT, CONTRAM and PACSIM.

The highest level of traffic detail is provided by microscopic simulation models which simulate the time-space trajectory of each individual vehicle by applying models

of car-following and lane-changing. They are more accurate than macroscopic models in estimating delays, queue lengths and other associated traffic characteristics, but often suffer from the deficiencies of the underlying microscopic models which may not have been well-calibrated. Representative examples in this category include AIMSUN2, PARAMICS, CORSIM, TRANSIMS, MITSIM and VISSIM.

Emergence of microscopic simulation as a viable practitioner option

The details in microscopic models yield the flexibility to add many more modeling contexts and options than macro and mesoscopic models, as well as show much more detailed graphical and animated displays. This makes it easier for them to be "sold" to the practitioners, despite the limitations of the fundamental equations therein, many of which may be rudimentary at this time (but could conceivably improve in the future if the models become popular). With some of the microscopic models having been developed commercially and marketed more aggressively than the models of the past, it is our opinion that microscopic simulation is indeed here to stay.

There is indeed a perceptible change in the practitioners' and researchers' view of microscopic simulation in recent years, brought about partly by the vastly improved computational capabilities and, along with it, the development of a few elaborate commercial microscopic simulation software. Microscopic simulation is now being considered as a potentially viable option for analyzing traffic networks in the near future. In addition to the analysis of real-time operational policies in urban transportation systems, microscopic simulation is considered a possibility even for planning purposes where static (assignment-based) models have been the primary method for decades.

6.3 Capabilities of current microscopic simulators in modeling non-auto traffic

The differences between microsimulation models existing in the market have been broadly studied in several research projects, such as Smartest (2000), Choa *et al.* (2002) and Bloomberg and Dale (2000) and their applications have been tested in many studies.

However, work involving these microsimulation packages has always been related to general traffic; very few studies in the literature deal with the simulation of transit and other special vehicle fleets. In addition, the objective of introducing the simulation of transit in a network has always been to evaluate the automobile performances taking into account the private user (car-owner) standpoint, thus focusing on the effect of transit on auto traffic. In very few cases has the simulation of transit been a tool to study and analyze the performance of the transit system itself from the point of view of the transit operator. Not many studies have been performed where the objective has been to simulate transit as an end in itself. In the following discussion, we use the word "traffic" to refer to the auto traffic.

Some of the traffic simulation software packages in the market nowadays are able to simulate fixed route transit systems quite accurately. In order to model such a fixed-route vehicle class, users have to follow the following steps, largely independent of the software they use. The routes for the system have to be predetermined, then locations of the stops need to be fixed and, finally, the frequencies of the service for all the routes have to be input. The packages such as PARAMICS and VISSIM have many vehicle types, allowing users to choose a different type for transit vehicles. The packages do not necessarily include sufficient details to model the operational

characteristics of any type of vehicles, such as say a people mover system or a streetcar system that shares the right of way with auto traffic.

Simulation packages vary in their ability to simulate transit. For instance, most available simulators have a fixed time for stoppage. However, in VISSIM, the duration of wait at a given stop for a transit vehicle can also be specified as a function of the demand at that place, namely the number of people waiting to take the bus in this location. Only VISSIM allows vehicles to stop/stage on the left-hand side of the road. Signal preemption schemes for transit vehicles are very difficult to model properly in the existing simulators primarily. Signal preemption for emergency vehicles is generally impossible to model in the packages because there is no simulation of a special class of non-auto vehicles without fixed routes. The available simulation packages are not flexible enough to simulate a real-time routed transit systems without any external subroutines like Application Programming Interfaces (API's).

6.4 Special Techniques to simulate Transit Systems

Some researchers have invented approximate techniques to overcome the deficiencies of commercial packages in simulating real transit systems, in effect "tricking" the package to do transit simulation.

For the simulation of LRT (Light Rail Transit) in Venglar (1995) using NETSIM, the network had to be modified in order to have fewer nodes and different configurations regarding the preferences in the intersections. For simulating a BRT (Bus Rapid Transit) in Inga (2001) using VISSIM, the corridor had to be divided into shorter sections in order to model the center-running guided busway on an arterial street. A different

vehicle type was coded for the buses as well. Another example of a BRT simulation using VISSIM is in Multisystems Inc. (2000). The network was coded similar to the preceding case, and additional signals were coded to hold vehicles at some pre-specified locations in the network and maintain a constant headway. The control held a bus in a location if its headway to the bus ahead of it was less than the minimum time desired. Another interesting example is a simulation of the streetcar system using PARAMICS for the city of Toronto (Abdulhai *et al.*, 2002). Tracks were coded on top of the existing network, and stops were coded as additional nodes with virtual traffic lights affecting only streetcars. The traffic was trapped behind the streetcars during the red phase of the traffic light. In this case, the duration of the stop of the streetcar was a function of the demand. Note that in all the above cases the location of the stops has to be pre-specified. It is not clear how to simulate a system with uncertain demand occurring at any random point in the network. The struggle by the researchers and practitioners to attempt to simulate anything other than an auto traffic system is sufficient cause for a careful look into the requirements for comprehensive microscopic simulation environments to be developed for the future.

6.5 Network hierarchy and network size issues

The primary difficulty with many microscopic simulation models is their inability to handle path dynamics in large networks. For example, PARAMICS allows vehicle routing according to routing tables and feedback capturing information supply, but does not allow storage of sufficient path trees and storage of individual vehicle's routes, which are essential requirements for the simulation of route choice. The difficulty arises

from the detailed network descriptions used in such microscopic simulation models. The node and link representations for microscopic simulations are often such that any point on a physical link with a change in geometry or other characteristics results in an extra node in the representation.

The need to properly model the street and lane infrastructure in microscopic models results in an order of magnitude more nodes and links than needed to model the path dynamics, which requires only the network comprising the true decision nodes. These are the nodes that are of significance in for example, the transit driver's route decision or the route decision taken by a central dispatcher in order to optimize the operation of a service fleet. Microscopic models such as PARAMICS have the scalability to permit vehicle simulation of large networks, but if detailed service response modeling and path processing are to be incorporated, such models can only be used to simulate small to medium-sized urban areas. This is because many network path processing algorithms show nonlinear increase in storage and computational requirements as network size increases, as opposed to the auto traffic simulation algorithms that can be intuitively seen to operate on local variables and thus show linearly increasing computational requirements.

Thus, if we consider large networks where a microscopic simulation data sets would includes several tens of thousand nodes, the storage of individual paths of each transit vehicle with such networks require prohibitive random access memory (RAM). This is true even with modern computers and thus microscopic models developed for such purposes will justifiably have limitations in path processing. It is logical to see that traffic flow modeling requires only local information and can be very scalable. That

is, larger networks can be modeled with carefully developed distributed processing schemes that allocate the modeling of portions of the network to different processors. On the other hand, modeling changes in path-related characteristics such as travel times, and path-related decisions by transit managers, requires information from possibly all parts, the paths going across sub-areas. Thus microscopic models which were developed with initial focus on auto traffic simulation in relatively smaller regions would need to be augmented with schemes to handle them at a different level of network abstraction, as discussed further below.

A previous work by Oh *et al.* (2000) developed a hybrid simulation approach, integrating the PARAMICS microscopic simulation with the routing and behavior response simulation schemes as in DYNASMART (Jayakrishnan *et al.*, 1994), so that the integrated simulator could evaluate information/routing schemes with route choice behavior models. This approach was based on integrating networks of two different levels of abstraction and communication of vehicle positions between the detailed network (as in PARAMICS) to the more abstract network (as in DYNASMART). The vehicle route decisions processed at abstract network are then transmitted to the detailed network simulation that controls vehicle movement at the microscopic level. The integrated simulation program allowed them realistic evaluation of a variety of technologies in advanced traffic management and information systems (ATMIS).

The scheme proposed in the context of this paper is based on some ideas developed in Oh *et al.* (2000), but oriented to a different kind of approach for communication, integration and routing. It is possible to study the detailed operation of a general transit or commercial fleet system, where all the path-based decisions,

treatment of passengers and routing of transit vehicles are made at an abstract level, while all traffic operations are controlled by the microscopic simulator. Thus, the vehicle route decisions processed at abstract network are then transmitted to the detailed network simulation that controls vehicle movement at the microscopic level.

Hence, one of the key aspects of our approach is the data communication from the micro-level to the abstract level and vice versa. The idea is to create a simplified network to be used at the abstract level, but consistent with the original network coded for running the microscopic model. In addition, this process may require the construction of a lookup table ("communication interface") for passing information between the two networks, such as level of service and detailed information of individual vehicle positions, speeds, route decisions, etc.

The equivalent abstract network (henceforth ABSNET) is made taking into account the following possible simplification. In terms of link characteristics, a relatively simple program can aggregate across those microscopic links, whose end nodes represent only a change in geometry or capacity. Note that this includes all nodes except for the real decision nodes such as intersections or interchanges. Calculation of link cost in the abstract network is consistent with the microscopic model link cost calculation, which includes link costs themselves, and turn movement costs. Look-up tables identify the original links that corresponds to the abstract network links and to aggregate travel times on them.

In Figures 6.1 to 6.3, a representation of a small network coded in PARAMICS, along with a representation of the corresponding ABSNET associated are shown.

In the case of simulating transit systems running on a subset of the whole area network (such as a fixed route system), the ABSNET will be a representation of just that sub-net (the rest is not needed at the abstract level). Moreover, for fixed route transit and fleet systems, where the route remains fixed, it is not necessary to transfer information related to link and turn movement costs, since there are no decisions taken according to the level of congestion of the network at any time. This is in contrast to simulating taxi, "dial-a-ride" or combined systems where the driver or the central dispatcher could choose the shortest path to get to his next stop, depending upon network traffic conditions. In this case, however, it is important to communicate vehicle-related information between both levels of abstraction, say the vehicle positions and the stop-time required by every vehicle when it reaches its predefined bus stops.

In the next section, a detailed description of a hybrid scheme to simulate a general transit or service system using detailed traffic simulation from a last generation microscopic simulation model is provided.

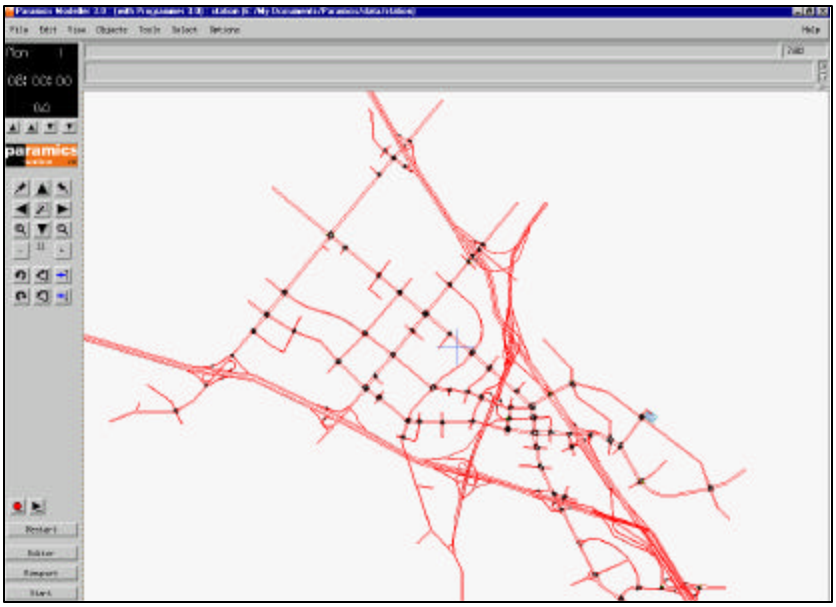

Figure 6.1 PARAMICS sample network

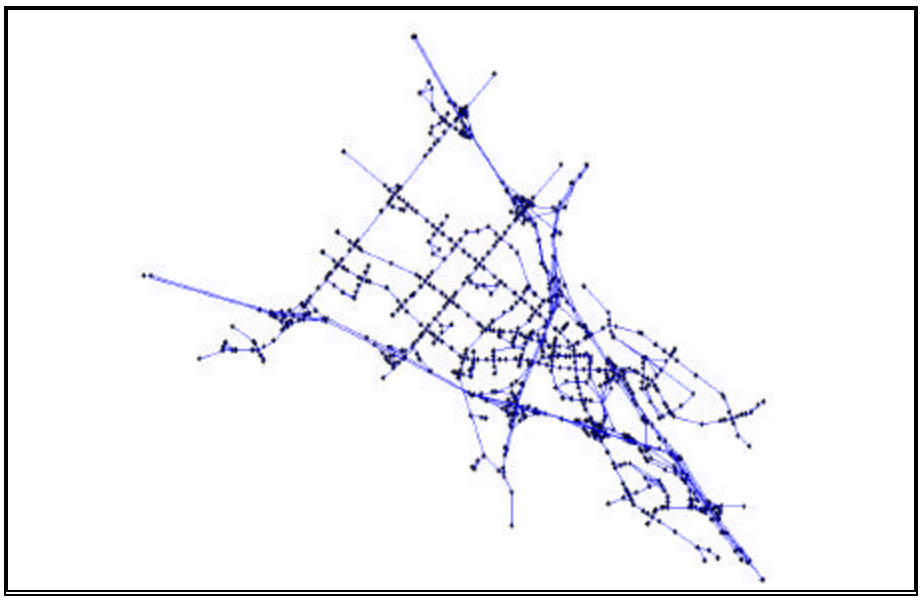

Figure 6.2 Detailed representation of the network

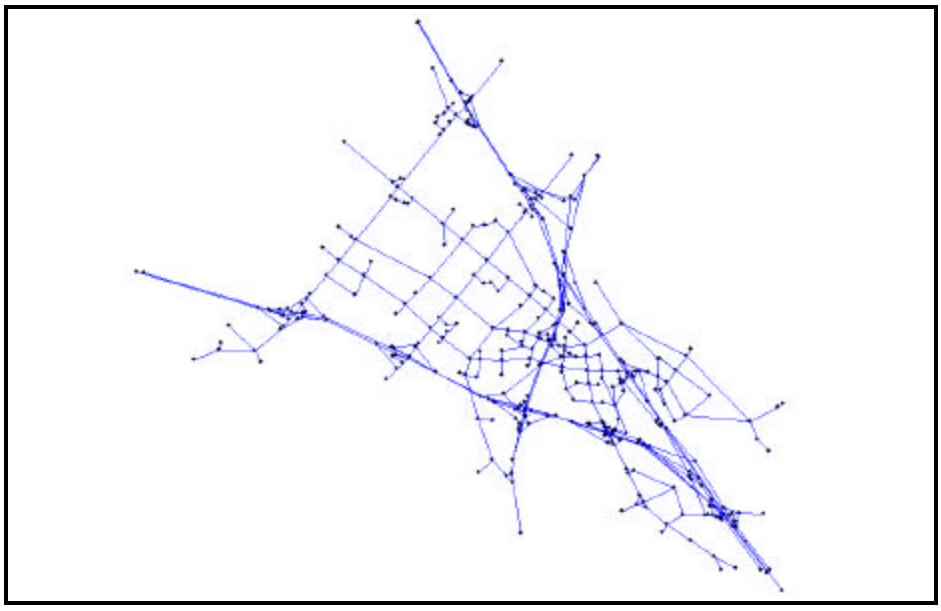

Figure 6.3 Corresponding ABSNET network

The general approach requires the microscopic model to have some modern software capabilities in terms of the development of functional interface or application programming interface (API), allowing additional functionality by adding more external

modeling routines. Many of the existing simulation software do allow such APIs, which may also be called plugins, as the developers are aware of the need for it for anything but the simplistic auto traffic simulation in realistic urban networks.

6.6 A general purpose scheme for simulation of multiple classes of vehicles and services

The modeling scheme suggested is for simulating any kind of flexible real-time routed service. In this sense, it will be possible to model optimal routing algorithms which may be based on the individual vehicle's position, passenger calls, and real-time traffic conditions, in case of studying "reroutable systems", such as typical paratransit, "dial-a-ride" or more complex designs such as *HCPPT*. In addition, the most general "pick-up and delivery" commercial fleet contexts could be modeled under real time traffic conditions, incorporating any kind of decision rule to assign vehicles to serve customers optimally. Note that fixed route systems are essentially a restricted subset of flexible-route systems or real-time routed systems.

The premise is that once the routing modules are separately coded in the simulation environment through the API, they can be easily modified to simulate various designs of fixed route systems, feeder short-haul services, etc. Customer demand generation and performance measures are also embedded into the simulation framework.

The new capabilities need to include the detailed modeling of vehicle operations at stop or passenger pick-up and delivery locations, real-time traffic network conditions impacting vehicle travel times and user waiting times at pick-up locations.

In Figure 6.4 a scheme is shown to represent the proposed framework for simulating/evaluating a general commercial fleet service.

General, meaning, all modules composing the integrated system can be adapted to simulate most transit and commercial services available nowadays. The next section provides examples of application of such a scheme to the modeling of various multiple-class vehicle/service systems.

First of all, in the figure it is possible to visualize the two levels of aggregation. On the right side of the chart, we have all the procedures (modules) corresponding to the aggregated level, including the implementation of any sort of routing/scheduling rule and the customer behavioral models. Note that we use the term "aggregated level" only under an assumption that the network on which the transit or fleet system's routing and other details are simulated would be a smaller version of the detailed microscopic auto traffic simulation network, generally with only "decision nodes", as mentioned above. It would be entirely appropriate to call this network "transit/fleet routing network".

Next three important issues regarding the proposed scheme are described: the objects (fundamental data structures), the important events that trigger some action at the aggregated level, and the routing/scheduling rules and interrelation modules within the API.

Fundamental data structures

The basic API frame is composed by the fundamental data structures that can be either objects or agents, depending upon the kind of system to be simulated. These are the “fleet”, the “customer (user)” and the “network” data structures.

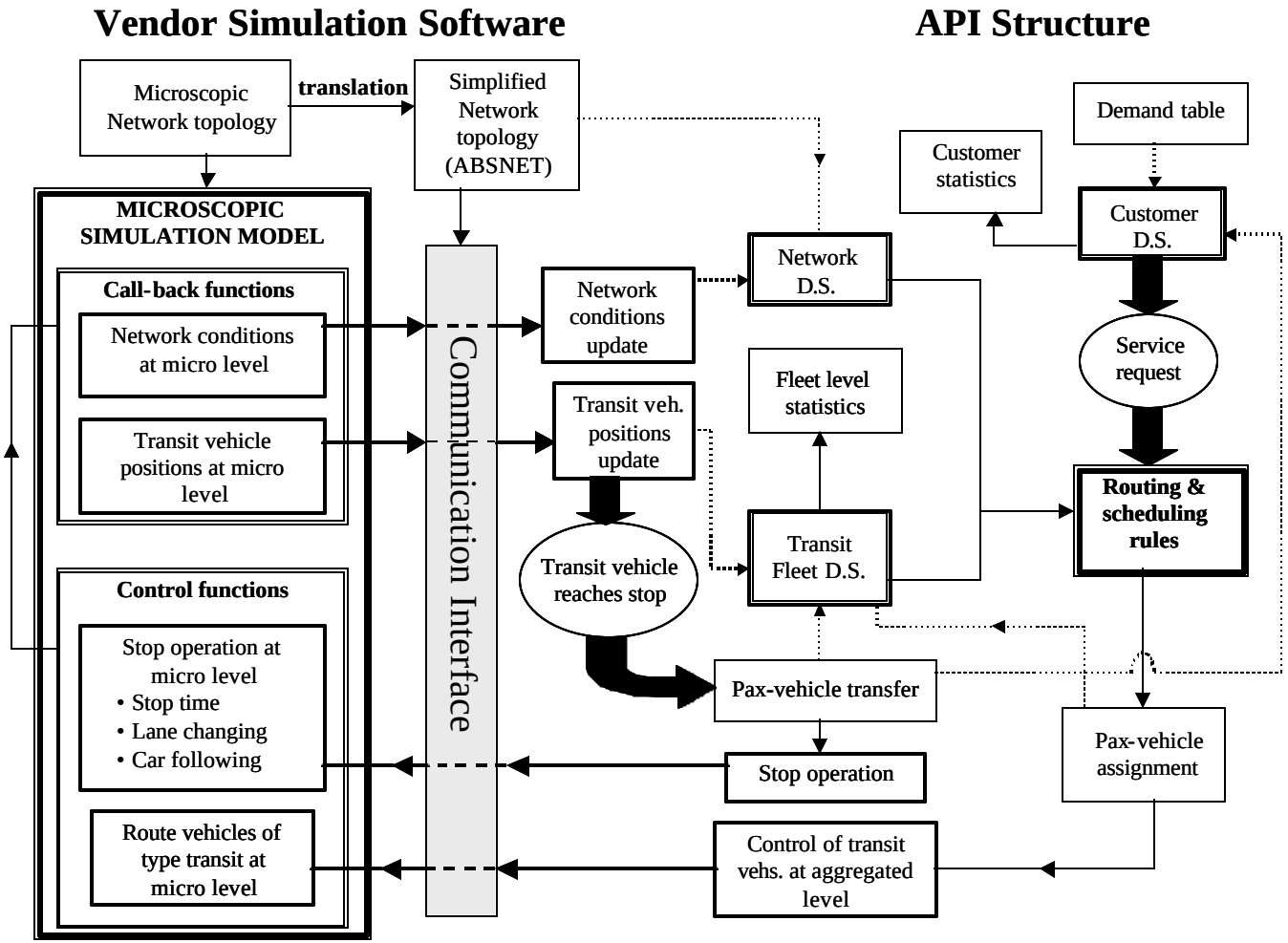

Figure 6.4 Simulation framework for a general transit system

The “network” data structure keeps the information contained into the ABSNET (simplified network) and the way in which the information is kept within it, will depend on the kind of algorithm to be run in the “Routing and Scheduling” module, requiring any attribute of the network at any time (i.e. link and turn movement cost). For example, one could need a different data structure for running efficiently a shortest path algorithm, or a TSP-based routine for vehicle dispatching.

The “network” data structure reads information from the “network conditions update” routine, which is fed directly off the detailed microscopic network conditions via the “communication interface” module that translates every network attribute from one level of aggregation to the other. The interface includes thee necessary Callback functions as defined in the microscopic simulation software's features). Normally, the network conditions used in the context of our scheme are the link travel time and turn movement cost (intersection delay).

The “customer” data structure can keep all the information obtained from a demand table, which can be either known in advance or generated in simulation time. It is important to store the attributes, special requirements and general features of the customers of the system. All information obtained from the simulation concerning the performance measures associated with the customer (say average ride and waiting time, which can be at the pick-up spot or in a bus stop, etc.) need to be stored, as these would be needed in any customer decision process modeling.

Finally, the “fleet” data structure becomes the central object of the system, keeping the details and behavior of all transit vehicles at the aggregated level. For each vehicle in the transit or fleet system that we keep track of in the ABSNET we have a

corresponding actual vehicle moving around in the microscopic simulation model. The correspondence between vehicles needs to be handled through a look up table (or a hash-table, depending on the microscopic simulation model's vehicle naming/numbering method). In this data structure, we will store the vehicle paths and stops at the aggregated level in order to control the movement of such vehicles at the microscopic level. These paths can be either fixed or variable, depending on the system, routing rules, etc. In addition, this data structure keeps all transit vehicles' features and they are connected to a module that stores the statistics of the performance of the system from a vehicle (manager) standpoint. In all figures hereafter, dotted lines pointing to data structure boxes represent all those procedures that update the information contained in the respective data structure.

Simulation events in the API

In the background section, the difference between continuous and discrete simulation was pointed out. The way in which microscopic traffic movement is simulated is by discretizing the continuous operation of vehicles on the network over time, using a fixed time step $\Delta\tau$. On the other hand, the nature of commercial fleet operations makes the use of a discrete event simulation approach more attractive. The key factors that trigger a change on the evolution of the system at certain moment are basically discrete events (see for example, the feasibility study approach used in Chapter 3, for simulating spatially a high-coverage point to point transit system under simplified conditions). In this case, the simulation tools utilized are those provided by the microscopic model itself (defining both, the simulation time-step $\Delta\tau$ and the update or feedback time-step Δt)

along with the embedded nature of a discrete event simulation for calling the API's procedures.

In the scheme of Figure 6.4 two discrete events are identified, which generates an action performed by some of the external routines: "Service request" and when a "Transit vehicle reaches a stop".

- Service request: Every time a customer asks for service, the central dispatcher has to take a decision of routing and scheduling, changing the conditions of the system. Basically, the general "Routing and Scheduling Rules" are called in order to decide which transit vehicle has to serve the new customer, and in which position of the specific vehicle' route, among all previously scheduled stops. Once this decision is made, the vehicle's path is modified in order to insert the new request into the original vehicle's route, changing the predefined vehicle' path at the aggregated level. This event happens at absolute time *s*, measured from the beginning of the simulation time t_0. The new vehicle's route is communicated to the microscopic model via the "communication interface" after the execution of the time-step at which *s* belongs to. Then, the new route is introduced into the microscopic level.

- Transit vehicle reaches a stop: Every time a vehicle reaches a stop location a transfer operation happens (whether passengers boarding to or alighting from buses, or a certain load is picked up or dispatched at a certain location, etc.). At this moment, say time *t*, the physical interchange takes place ("pax-vehicle transfer" box), modifying both the "customer" and "fleet" databases. The former is need as there are changes in the status of the customer at *t*. The latter is needed because when a

vehicle reaches a stop location, its load and status change. The details of the stop event operation depend on the type of operation, number of entities transferred, conditions of the transfer, physical place of the stop, etc. After considering all these stop features, the stop time and exact location of it are then transferred to the detailed microscopic network in the same way as before. At the microscopic level, some Control function written in the API software have to be modified in order to make the stop as realistic as possible, most of them associated to lane changing and car following procedures.

In the next sub-section, the role played by the "Scheduling and Routing Rules" is described in the context of the design of such an API, the importance of this module under different schemes and its interrelation with other modules and data structures.

Routines for routing and scheduling

The box containing the Scheduling and Routing Rules cannot be explicitly defined for the general case. Every particular system is commanded by a different set of rules, however, there are some common characteristics regarding this issue that can be broadly discussed here. A more detailed definition of this topic is presented in the next section for the *HCPPT* case. The discussion is largely about transit systems, but replacing the word "passengers" with "packages" would yield insights into a package pick-up-and-delivery system as well.

As we show in Figure 6.4, this procedure is called every time a new service request enters the system. The main objective of this procedure is to decide the

customer-vehicle assignment following any general objective function to be optimized (see Chapter 4 for details).

In some cases, as in service of fixed route, the rules are oriented only towards the assignment of passengers to the right vehicle, once the vehicle arrives to a terminal or bus stop, depending on the distribution of the demand, and customer features. In case of real-time routed services, on the other hand, where vehicles can change routes dynamically, the objective concerns both passengers and vehicles. And, for mixed services, combining long-haul corridors fed off "reroutable" services operating on surface street areas, routing rules will apply to some vehicles but not for others under fixed route operation. The interchange of passengers occurring at terminal or hub locations will define the limit between one kind of operation and the other (see Section 6.7.8 for implementation and operational details).

The other important issue to be considered in these procedures is more related to the nature of the demand: whether it is a system with uncertain demand generated in real-time or it is a system where the demand is known in advance. In the first case, depending on the scheduling rule used, the vehicle-passenger assignment does not necessarily have to end up as a transfer. The reason is that the optimization is made in real time and the assignment decisions could change over time before the transfer itself. That is why the "passenger-vehicle assignment" routine does not mean an update in the "customer" data structure attributes. In case of demand known in advance, the optimization would most probably take place at the beginning, and the resulting routes will remain invariant over the whole simulation period.

With regard to the initialization of the system, one needs to set the initial vehicle routes (which will not change in case of fixed-route services), read the vehicle lookup table, the ABSNET and the demand table as well. All this information has to be input as a part of the API-setup function.

In the next section, the details of the simulation plataform for the *HCPPT* case are described. The corresponding routing rules and procedures in coding/implementation of such a system are highlighted and discussed in detail

6.7 Application of the proposed simulation scheme to incorporate *HCPPT*

6.7.1 General considerations

The objective of this section is to show the details of the implementation of the discussed simulation scheme for the *HCPPT* system case. The data structures introduced in 6.5.1 remains the same; however the events and the interactions among the platform pieces are somewhat different.

The interactions shown in Figure 6.4 are more complex for the combined as illustrated in Figure 6.5. Figure 6.5, though seemingly more complicated, is still a representation of the general scheme illustrated in Figure 6.4.

The same data structures are kept for all objects, however the interrelation among entities is more complex due to the addition of two vehicle states corresponding to its presence on either the fixed portion or the reroutable portion of the trip.

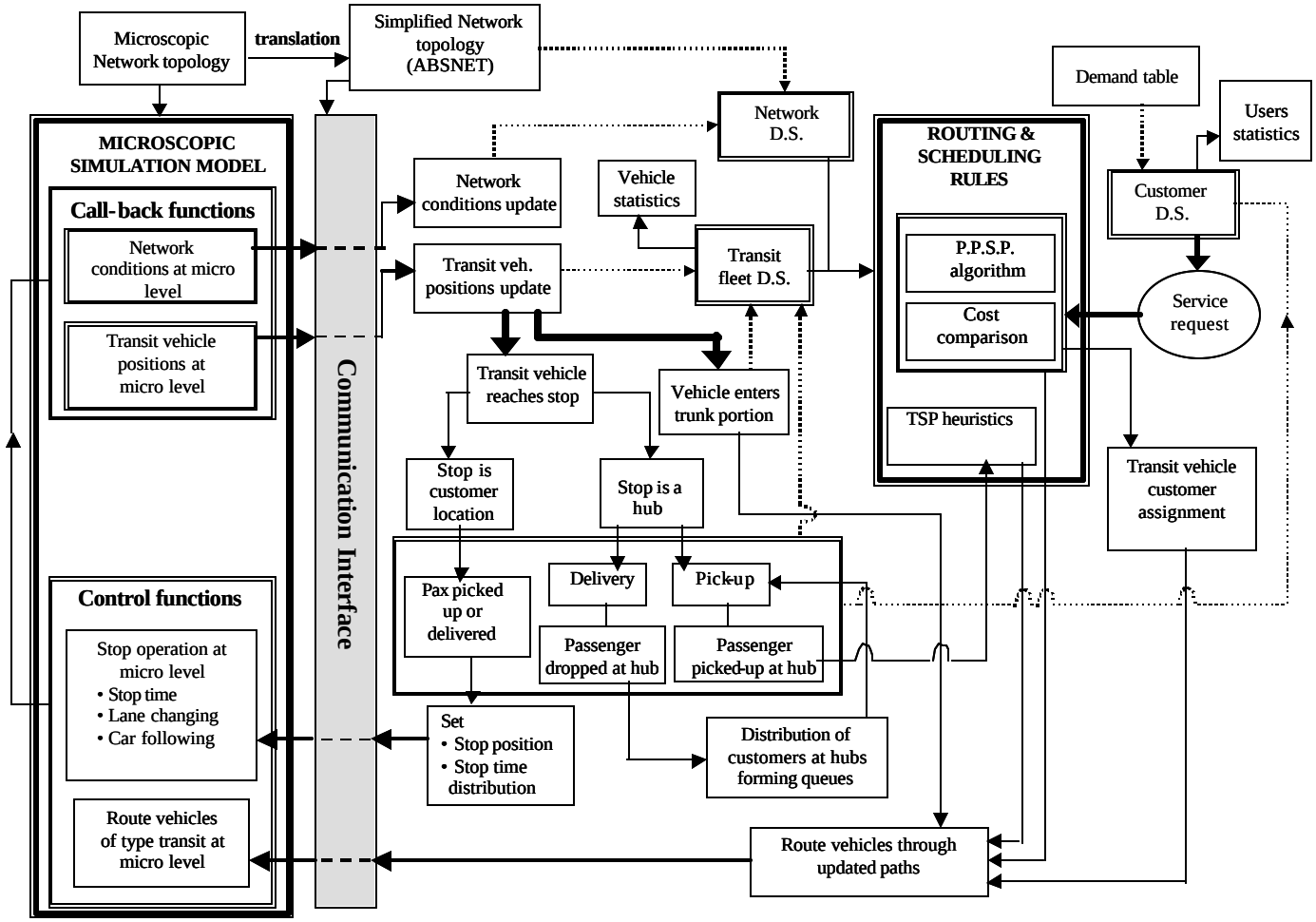

Figure 6.5 Simulation framework for *HCPPT*

In terms of events, one additional event is added, forcing an action at the abstract level, which is when vehicle enters the trunk portion of its route. At this time, the modeler sets the path as fixed, and sends that information to the microscopic simulator. The procedure "vehicle reaches stop" is further subdivided into two different cases: whether the stop is a customer location or a hub (terminal).

The interesting case is when the stop is a hub. At that point, the vehicle performs deliveries and pick-ups. However, in this case the passengers who are delivered there may not necessarily leave the system because the current hub may not be their final destination. In general, they are transferred to a different vehicle depending upon factors such as the size of the queue, the frequency of service, etc. That is why the "distribution of customers at hubs forming queues" procedure is depicted by a closed-feedback loop linking deliveries to pick-ups (the details of the operation schemes were presented in Chapter 4). The detail hub operation coding is presented later in Section 6.7.8.

In the figure, and according to the rules described in previous chapters, it was added the function within the "routing and scheduling rules" routine that computes the initial TSP route of a vehicle entering the reroutable portion of its journey for distributing all passengers picked up at the hub stop. The resulting route is conveyed to the microscopic simulation model through the "communication interface" module. All other procedures remain unchanged.

In the next subsection, the implementation of the most important routines of Figure 6.5, using the API functionalities of PARAMICS is described.

6.7.2 Communication interface

As mentioned in Section 6.5, the main idea of this interface is to convert the complex network details in PARAMICS into a simplified and aggregated abstract network called ABSNET, which only represents links needed to handle path-related decisions. In a system like *HCPPT* vehicle decisions are heavily based on the current information about link costs, position of vehicles, location of demand, to name just a few. Therefore, it becomes essential to develop an efficient data communication library composed of several subroutines that relay this information to-and-fro between the operator (decision-maker) and vehicles (PARAMICS objects). This data communication library can be categorized into three fundamental modules.

Lookup module

This module represents the backbone of the data communication library. This is where information gathered from the aggregation of PARAMICS network is first stored into data structures corresponding to the network, links, nodes and zones. Also, this module contains all the correspondence between ABSNET and PARAMICS nodes and links, which would become useful in cost-finding and position-finding modules. Following are the major functions in this module:

- Reading nodes and zones data: PARAMICS and ABSNET node names are converted into integer codes and stored in the network data structure. Zone information such as zone number and the corresponding origin node is also read and filled into the network data structure.

- Reading link data: ABSNET link information is read into a forward star data structure. This data structure stores the origin and destination nodes of the multiple PARAMICS links that comprise an aggregated ABSNET link.
- Node lookup: Given a node ID in PARAMICS, this function returns the corresponding node ID in ABSNET, and vice versa.
- ABSNET link search: Given an origin and destination node, this function returns the ABSNET link ID corresponding to this pair.
- PARAMICS link search: Given an origin and destination node, this function returns a pointer to the corresponding PARAMICS link.
- Finding the first and last PARAMICS links: The operator needs to know when a vehicle transfers from one ABSNET link to other. This can be simply thought of as transfer from the *last* PARAMICS link of one ABSNET link towards the *first* PARAMICS link of another ABSNET link. Given an ABSNET link, these functions return the first and last PARAMICS links associated with it.
- Link correspondence: This function returns the ABSNET link associated with a given PARAMICS link.
- Link length: Given an ABSNET link, this function sums the lengths of all PARAMICS links it contains and returns the true aggregated link length.

Cost module

All decisions regarding the routing and scheduling of *HCPPT* vehicles depend on knowing the travel costs. In the network context, it is much more logical to consider travel time as a suitable representation of this cost, rather than distance. After a certain

time period specified by the user (known as feedback period), PARAMICS recalculates these costs for all links. *HCPPT* vehicles are routed according to these costs, which change over time based on traffic conditions. This module performs two functions that are described below.

- Computing link costs: This function takes in an ABSNET link ID and returns the cost of traversing that link, by summing the link costs of all PARAMICS links associated to it.
- Computing turn costs: This function takes in two ABSNET link IDs and computes the cost of exiting from one link into the other. Note that is one example where it is necessary to know the first and last PARAMICS links associated with one aggregated ABSNET link.

The following PARAMICS Callback functions are needed for running the two aforementioned modules: "link_cost" returns the cost in seconds associated with a specific PARAMICS link in its cost table. This is usually the free flow travel time from the link start line to the link stop line. This does not include any turning penalty or feedback cost, and "link_exit_cost" returns the cost in seconds of traveling from the start of link1 to the start of link2 in the cost table, thus this cost includes the cost of the turn at the end of link1. If feedback routing is used, this value represents actual travel times.

Position finding module

The knowledge of the correspondence between ABSNET and PARAMICS network becomes indispensable here. The passenger demand that is generated in the abstract network needs to be represented in the PARAMICS network so that vehicles can

perform their pick-up and delivery actions. To facilitate these actions, the following functions were coded in this module.

- ABSNET link position: Given the location (that is, link ID and distance from downstream end) of a point in the detailed PARAMICS network, this function returns the corresponding location in the abstract network.
- PARAMICS link position: Given the location of a point in the abstract, this function returns the corresponding location in the PARAMICS network.
- Stopping distance: Vehicles have to decelerate to stop at any point. The numerical value of this deceleration is calculated by knowing how far the vehicle is from a stop (i.e., passenger). This function takes in a vehicle pointer and computes the current distance between the vehicle and the stop.

6.7.3 Generation of parametric demand

At this point, the demand will be generated based on the demand matrix of the microsimulation, assuming a parametric modal split (say, 2, 4, 6, 8 % of the PARAMICS demand) assigned to the *HCPPT* system. A more elaborated methodology in order to get a modal split modeling is mentioned in the conclusions chapter as part of the proposed further research. A simple demand model is numerically tested in Chapter 7 using this simulation scheme in order to have an idea of the potential of the proposed system as a viable travel alternative for automobile users.

The demand generation is as follows: once the passenger demand is generated into the PARAMICS zones, such passengers are distributed over the physical aggregated network ABSNET, identifying the specific places where the passenger (or a set of

passengers in case of adding pooling to the modeling) is (are) generated. The passenger demand is distributed uniformly across all link stretches associated with each PARAMICS zone, excluding stretches close to signalized intersections and a priori restricted links (such as freeway links or unconnected links). The modeler arbitrarily defines the link stretches that are associated to each PARAMICS zone (based on distance and gridness considerations). Thus, a one-to-one mapping, translating links into coordinate stretches is written. Then, uniform random numbers falling into the stretches are generated, defining the exact passenger call location. The time between calls is also assumed to follow certain distribution (negative exponential, uniform, etc.).

The group or pool size at each stop is computed assuming certain discrete distribution as explained in the next sub-section. Once the total number of passengers is finally generated, based on the original vehicle demand along with the assumed pooling distribution, the process is terminated and the calls are set in time and space.

Candidate passenger demand generation regions associated to specific PARAMICS zones are illustrated in Figure 6.6.

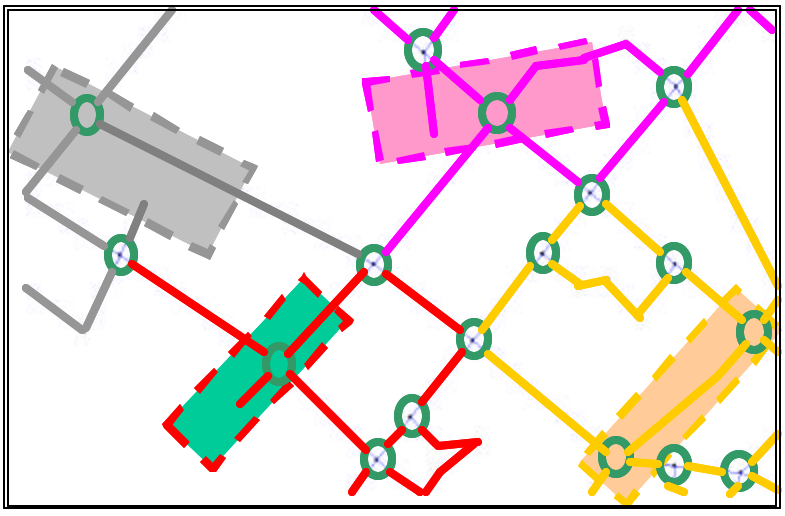

Figure 6.6 Passenger demand generation example

6.7.4 Passenger pooling modeling

As mentioned in Chapter 3, vehicle occupancy can be improved encouraging passengers to wait for the transit vehicle at a common origin spot. That is what has been called "passenger pooling strategy", in which passengers can join at the same stop, for a fare incentive. This strategy is superior to "car pooling" due to lack of restrictions in destinations. This is a relatively more controllable demand pattern, as fare incentives can be used to encourage passengers to "pool" together at stops.

The question here is how to simulate this kind of operation. In Chapter 3, for the feasibility study, a simple discrete probabilistic approach was assumed. In fact, it was introduced certain probability of having more than one person waiting to be picked up at any origin (say, $P = 1/3$). Additionally, among all the pooling origins, another probability of having one or two additional people there (say 0.5 for each case) was assumed.

In this more detailed simulation scheme, two discrete distributions were investigated to model the phenomena of passengers pooling together at a specific stop: a binomial and a Poisson distribution.

For the binomial random variable methodology, let us suppose that n independent passengers are willing to pool at certain stop. Each one has a probability of showing up at the stop (say p), and consequently $1-p$ of not showing up. If X represents the number of success (passengers joining at the stop) that occurs in the n trials, then X can be modeled as a binomial random variable with parameters (n, p).

The probability mass function of a binomial random variable having parameters (n, p) is given by

$$p(i) = \binom{n}{i} p^i (1-p)^{n-1}, \quad i = 0,1,\ldots,n \tag{6.1}$$

In Figure 6.7, the cumulative distribution function F for such a distribution is plotted assuming $n = 3$ and $p = 0.45$.

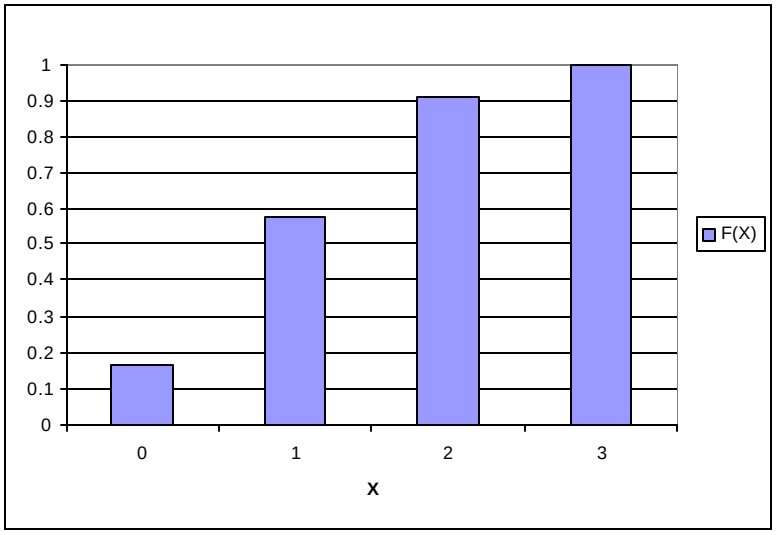

Figure 6.7 Graph of F(X): binomial distribution

Notice from the figure that the event $X = 0$ has also a positive probability of occurrence (the case when nobody meets at the stop). In the case of Figure 6.7, the mean is 1.35 with standard deviation of 0.86.

A binomial (n, p) random variable can be easily simulated by recalling that it can be expressed as the sum of n independent uniform (0,1) variables, then letting

$$X_i = \begin{cases} 1 & \text{if } U_i < p \\ 0 & \text{otherwise} \end{cases}$$

it follows that $X \equiv \sum_{i=1}^{n} X_i$ is a binomial random variable with parameters n and p.

An alternative approach is to assume that X is a Poisson random variable with parameter λ. In such a case, for $\lambda = 1.5$, the cumulative distribution function is shown in Figure 6.8. In this case, the mean is 1.5 and the standard deviation is 1.225.

To simulate a Poisson random variable with mean λ, one has to generate independent uniform (0,1) random variables $U_1, U_2, \ldots$ stopping at

$$N + 1 = \min \left\{ n : \prod_{i=1}^{n} U_i < e^{-\lambda} \right\} \tag{6.2}$$

The random variable N is then Poisson, which can be seen by noting that

$$N = \max \left\{ n : \prod_{i=1}^{n} -\log U_i < \lambda \right\} \tag{6.3}$$

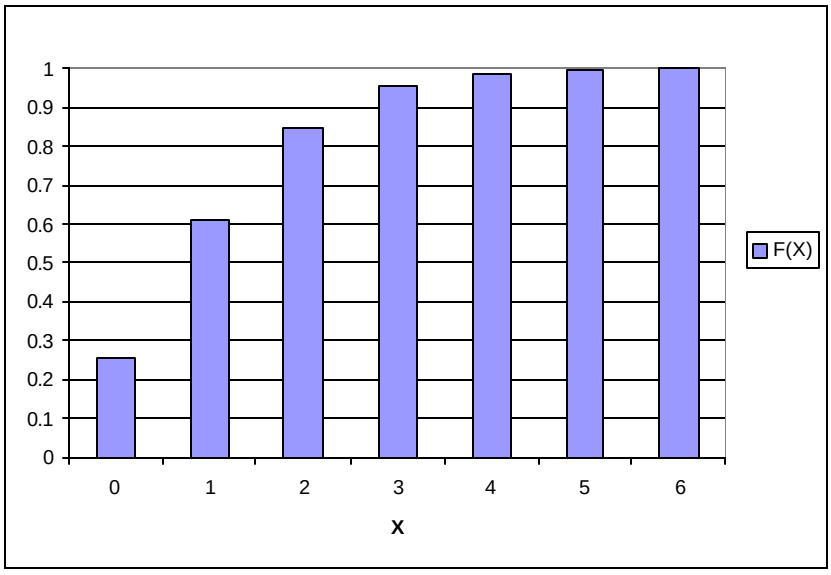

Figure 6.8 Graph of F(X): Poisson distribution

But $-\log U_i$ is exponential with rate 1, and so $-\log U_i, i \geq 1$ can be interpreted as the interarrival times of a Poisson process having rate 1, therefore $N = N(\lambda)$ would equal the number of events by time λ. Hence N is Poisson with mean λ.

A more detailed behavior-based methodology for modeling this process is proposed in Chapter 8 to be investigated in further research.

6.7.5 Vehicle generation, path modeling and routing

In this section, a broad description of vehicle generation and routing issues are discussed.

The number and distribution of transit vehicles will be determined by applying a simple fleet requirement model based upon level and distribution of historical real-time passenger demand. In Chapter 7 this simple model is presented and quantified for the case study on the Orange County network. Regarding the generation of this type of transit vehicles in the context of the PARAMICS simulation, it must be noted that they are labeled as a special class of vehicle (tagged vehicles in PARAMICS). They are also generated within the "zone_action" function, which is used to determine if an additional vehicle of certain type and traveling towards a specific destination is to be released, independent of the usual OD matrices used in the simulation. When the "zone_action function is called, the API code triggers the release of a vehicle by setting the type of the vehicle to be released. This is done by calling "zone_vehicle_type_set" with the type number of the vehicle that is going to be released. With suitable counters, it is possible to release the desired number of vehicles properly distributed over the cluster areas.

In addition, the driver aggressiveness and awareness are set at proper values. Finally, using "zone_vehicle_destination_set" function, the destination zone of the transit vehicle is input. This value is set in such a way that the vehicle never reaches such a destination zone. Otherwise, it would be killed by PARAMICS in the middle of

the simulation. By observing the design features and the routing rules assigned to each tagged vehicle, it is easy to ensure this condition for all transit vehicles.

The "routing and scheduling" module of Figure 6.5 comprises several vehicle routing decisions and algorithms described in the previous chapters as well as here. From all these optimal decisions taken in real-time, a set of paths (one for each vehicle) represented by a sequence of ABSNET nodes will be generated and updated every time a vehicle crosses one of such nodes. In fact, a PARAMICS override control function "vehicle_tagged_transfer" along with the capabilities of the communication interface module, allows the modeler to realize actions every time a tagged vehicle transfers from one link to another link. Additionally, a default path (associated with each cell) and a trunk path (associated with a pair of adjacent hubs) can be also accessed by the decision maker in order to route the vehicle according to its current state and assigned task. A proper data structure is stored to keep track and control tagged vehicles after they have been generated.

PARAMICS allows the plugin to control individual vehicles route choice on a turn by turn basis. This control is provided through the use of the "routing_decision" function, which is enabled through the "routing_enable" function. The "routing_decision" is called for every tagged vehicle. When called, the plugin receives a pointer to the link where the route choice has to be made, and a pointer to the vehicle that needs to make the route choice decision. The "routing_decision" function can then make the choice of which exit the vehicle will take by returning the exit index of the desired next link. For all remaining vehicles on the network, the "routing_decision"

function returns 0, in which case PARAMICS uses its standard route choice model and make its own selection of exit.

A vehicle keeps a two-turn look ahead for its own route choice. This means that at the point a vehicle is released it looks to see what turn it should make at the end of its current link, and then at the end of the following link. Each time a vehicle is transferred from one link to another, it then updates its look ahead so that it always knows its next two turns. The way to manage these routing decisions taken ahead of the real vehicle position is through the generation of virtual vehicles. A virtual vehicle is simply a copy of the real vehicle created by PARAMICS. Such a vehicle is placed at some future point (usually 1 or 2 links ahead) to help determine both future link choice and future lane choice. Notice that all routing decisions can be made only from the position of the virtual vehicle, since the preceding vehicle route is already decided in PARAMICS and cannot be changed. This assumption is quite realistic. The position of the vehicle at the ABSNET level is updated every time the "routing_decision" is called, and will match the position of the virtual vehicle for all routing purposes.

The communication interface translates the decision node (at the ABSNET level) into a PARAMICS exit link according to the network translation mapping already described in Section 6.7.2.

The first time that "routing_decision" is called for a specific vehicle of the set type, a lookup table member is added to a hash lookup table (passing vehicle pointers and ID back and forth) in order to add it to the aforementioned data structure. Here, the vehicle is also tagged as transit vehicle.

6.7.6 Passenger and vehicle data structures special management

Once passengers are generated, they are kept in a data structure containing all their features and statistics. Passengers are initially stored in a customer list CUST. Once a specific group of passengers (pooling at certain stop) is assigned to certain vehicle, they immediately are moved to the corresponding vehicle member data structure. They are added to an assigned passenger list associated to the specific chosen vehicle. They remain in such a list until the vehicle picks them up. Then, they are split into individual members and they are transferred to the suitable list in the vehicle member data structure, whether they are going to be delivered insider the same cluster zone or to some adjacent cluster zone. The former passengers are then added to an internal delivery list while the latter are moved to a pick-up list, all list associated to the specific assigned vehicle.

When passengers transfer at a hub, they are transferred again from the vehicle (in the corresponding list) to a list of passengers at the transfer hub. This list could be organized as a structure of queues according to the queuing strategy introduced in Chapter 4, or stored simply in a simple queue if no special passenger arrangements are made at hubs.

When the passenger is picked up at a hub, he(she) is added to the vehicle external trip list to be distributed at their final destination. Once the passenger is finally dropped to his(her) final destination, the customer element is deleted from the external trip list and a record of the passenger trip is stored in a CUST statistics data structure.

6.7.7 Network connectivity constraints

At any time along the simulation period, each vehicle requires information about what route to follow. If not, it could eventually reach its PARAMICS final destination zone (which is really virtual as explained in the previous section) and disappear. Moreover, given the structure of any simulation network, it could happen that some links belonging to the network will be connected only to a reduced set of links. In such a case, if for example a vehicle were assigned to pick-up somebody on that region, it would maybe not be able to go back to its assigned path.

Consider the portion of a network in the example of Figure 6.9.

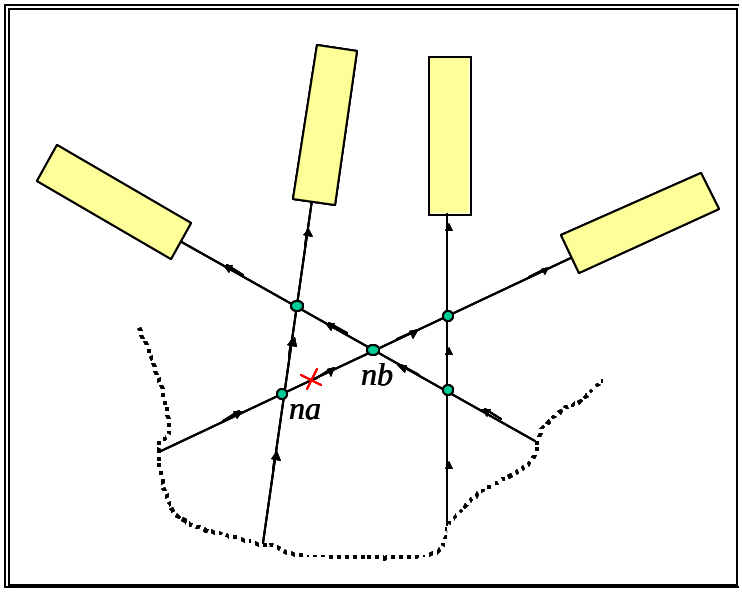

Figure 6.9 Unconnected network example

All links are one-way links and boxes represent PARAMICS destination zones. If a call is generated for example along link (na, nb), and the central dispatcher decides to send a vehicle to pick-up that customer, the vehicle would not be able to return to the network, even though (na, nb) is not a terminal node. Therefore, in order to make the simulation

run properly, no call was generated on these unconnected links. For checking that, an all-pairs shortest path algorithm was run for two cases: from every link to any node and from any node to any link. Based on the number of connections of each link based on these two criteria, some of them were discarded for candidate demand generators according to the methodology introduced in Section 6.7.3.

6.7.8 Handling vehicle stoppage and treatment of terminals

In PARAMICS, and also in the design of the ABSNET, a stop is completely defined by two variables: the link where it belongs and the distance from the downstream node of such a link. Moreover, a stop can represent any physical point on the network (a pick-up or delivery location, the current position of a vehicle, etc.), and these two attributes are enough for its complete identification.

One of the most relevant operations in the context of this simulation scheme is the vehicle stoppage at pick-ups, deliveries and transfer points. The first two cases, called normal stop are simpler than the operation at terminals. Nevertheless, the basic simulation techniques used to mimic the stoppage operation in PARAMICS are common to both scenarios.

In order to simulate a pick-up, delivery or transfer vehicle operation, there are two behavioral aspects that require certain treatment: lane changing and vehicle stoppage. The former since in reality the transfer operations should happen in the right most lane, and the latter because the vehicle needs to stop for a certain time and position in order to simulate a proper passenger transfer operation. In case of terminal

operations, an additional treatment of available sites for the vehicle to park is needed and it is discussed later in this section.

The transfer will occur somewhere along a PARAMICS link, which can be translated into the corresponding ABSNET link dictionary. Thus, the stop link and the exact stop location trough the distance from the downstream node are completely identified.

In case of the lane changing procedure, the PARAMICS override control function "move_in" was utilized in order to convey a lane-changing stimulus to move to the right, based on the target vehicle attributes and the positions and attributes of the surrounding vehicles. The stimulus is transferred to the vehicle from the time it crosses the upstream node of the stop link till the vehicle finally stop. In practical terms, the function is called every simulation time step $\Delta\tau$ for transferring such a stimulus while the vehicle is upstream from the stop location.

Two additional variables (acceptance and setting times for lane-changing) are also properly set. The "gap_acceptance" PARAMICS function is also called in order to adjust the gap of the vehicle and make the lane-changing maneuver easier and softer.

The second step is to stop the vehicle at the right location. In order to do that, a space window is identified around the stop location on the chosen link, and using the override function "vehicle_speed_set" along with the callback function "vehicle_speed", the needed deceleration is set on that vehicle to make it stop smoothly at the right position for the scheduled stop time with very high accuracy. Both stop time and position conditions are checked every simulation time step within function "vehicle_tagged_move" that is called for every vehicle at every time step.

The stop in terminals requires additional treatment. Each terminal comprises several links and each link contains several sites for stopping. The difference with the normal approach introduced above is in determining the exact position of the stop. In this case the stop location is not fixed as in the previous case and depends on the conditions of the terminal and on the number of vehicles stopped there occupying different sites on different links when the target vehicle reaches the hub. All the information regarding terminal conditions is kept on the corresponding hub data structure member.

Thus, when the target vehicle reaches the hub terminal, a site is assigned to it on a chosen link associated to such a hub. The site is set to "occupied", so that no other vehicle will use the same site spot while the target vehicle is stopped there. Once the vehicle leaves the site and finally leaves the hub, the hub data structure is updated setting the stop site as "unoccupied" and updating all customer, hub and vehicle data structures in order to update the simulated passenger transfer operation there.

In Figure 6.10 the complete stop process is graphically represented.

6.7.9 Point-to-point shortest path routine

6.7.9.1 Generalities

This section describes the algorithm utilized for computing the point-to-point shortest path travel time between any two stops belonging to the ABSNET at any time.

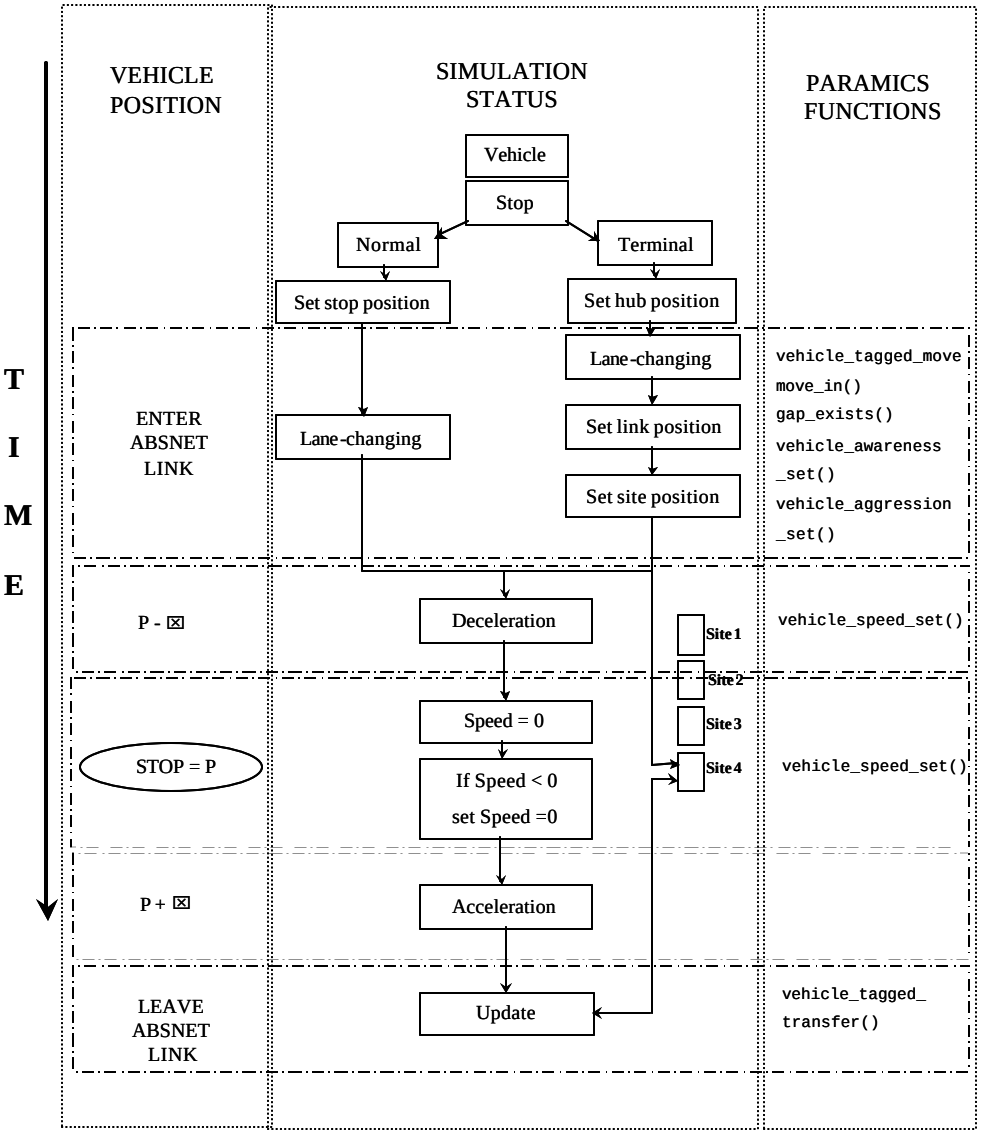

Figure 6.10 Simulating a passenger transfer operation

This particular simulation procedure requires a very efficient point-to-point shortest path routine that has to be called multiple times when deciding vehicle routes in real time according to the routing decision taken by the central dispatcher each time a new pick-up request enters the system. If the demand is very high or the network utilized is very large, the efficiency of this algorithm will be crucial for running the simulation with routing decisions taken in real time.

In fact, the insertion cost function incorporated in the "Routing and scheduling" module depend on the travel time from the current vehicle location and the pick-up location of such a request, and this will happen for all candidate vehicles every time a new request comes in.

In addition, the ultimate goal is to code a stop-to-stop shortest path algorithm, where the stops are points located in the middle of any ABSNET link. One additional difficulty is to incorporate the two components of the link cost considered by PARAMICS: link cost and turn movement cost. Thus, the algorithm has to be designed in order to manage two different network forward star data structures (coded as adjacency-list representations), one considering the cost from node to node (link cost), and another including cost from link to link (turn movement cost). In Figure 6.11 an example of the cost components included on a simple path are shown.

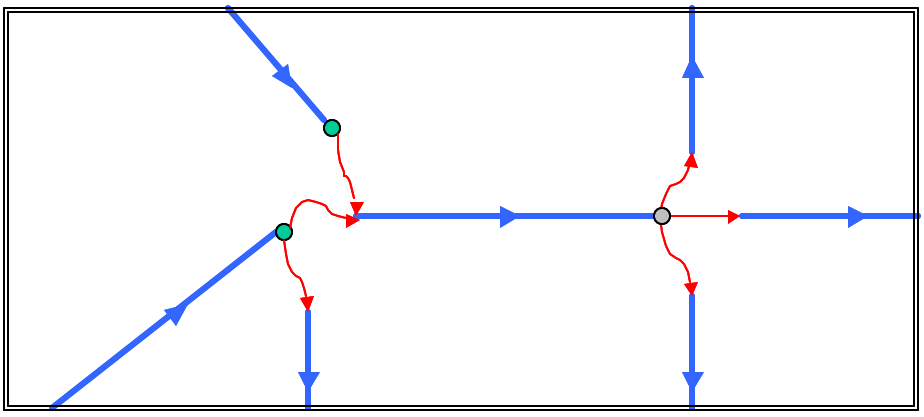

Figure 6.11 Link and turn movement cost representation

The complete link cost is computed from node to node as in the figure. Therefore, the cost for crossing a link will depend on where the vehicle is coming from. The link cost functions coded as part of this API (Section 6.7.2, cost module) follows this convention. That is, the turn movement cost happening at the beginning of the link is summed to the

downstream link cost in order to obtain the total link cost. This convention is consistent with the logic of most of the shortest path algorithms in which the predecessor to a node is normally known when computing link costs form such a node to all their neighboring nodes.

The literature provides several references for the single origin/single destination problem under different network topologies. Two-sided label setting methods for this problem were proposed originally by Nicholson (1966); see also Helgason *et al.*(1993), which contains extensive computational results. The idea of using lower estimates of the shortest distance to the destination in label correcting methods has been studied by Pearl (1984). Another alternative is to implement a combined forward/reverse auction algorithm, due to Bertsekas (1991). The implementation of an auction algorithm with graph reduction was explored by Bertsekas *et al.*,(1995). An analysis of a parallel asynchronous implementation is given by Polymenakos and Bertsekas(1994). Some variants of the auction algorithms that use slightly different price updating schemes have been proposed by Cerulli *et al.*(1994). A method that combines the auction algorithm with some dual price iterations was given by Pallotino and Scutella(1997).

In this implementation, an algorithm for solving the stop-to-stop shortest path problem is developed based on the original label setting implementation by Bertsekas (1998).

6.7.9.2 The proposed stop-to-stop shortest path algorithm

Before describing the algorithm and the application to this particular case, let us clearly define the problem and the associated cost structure according to the scheme presented

in the generalities subsection. The objective is to solve a stop-to-stop shortest path problem, in which stops happen mostly in the middle of any ABSNET links. Even though the turn movement cost is actually associated to a link, the notation will be defined in terms of nodes, since the proposed shortest path algorithm follows a node-based logic. Therefore, let us denote tc_{ijk} and c_{jk} the turn cost for the movement from link (i, j) to node (j,k) and the link cost associated to link (j,k) respectively. Consider the schemes of Figures 6.12 and 6.13 for the origin and destination locations of a trip segment (could be from the current vehicle position to a stop or between to stops, etc.).

Notice that the destination happens along a two-way link. In that case, it is assumed that the pick-up or delivery will be either on link (k,m) or on link (m,k), depending on which path turns out to be cheaper. In the origin case, the problem is easier since normally is referred to the current position of the vehicle; therefore the link is always defined, and the shortest path can be computed from the downstream link node. However, since the turn movement costs are important, the iteration should start by knowing the predecessor to the origin node j (in this case node i).

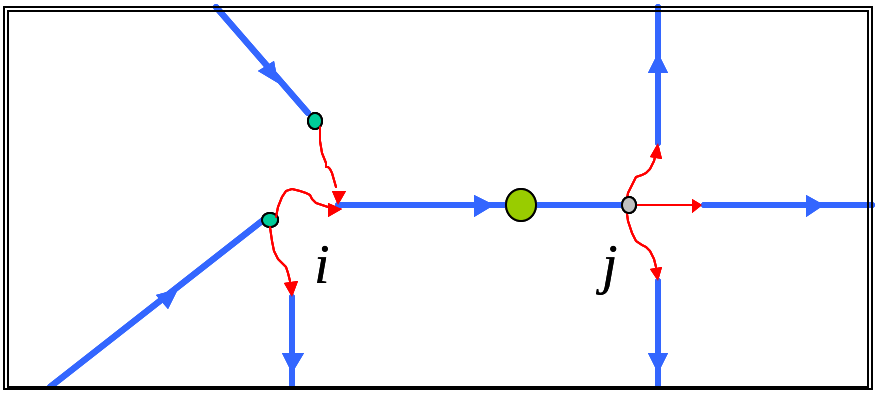

Figure 6.12 Origin stop location

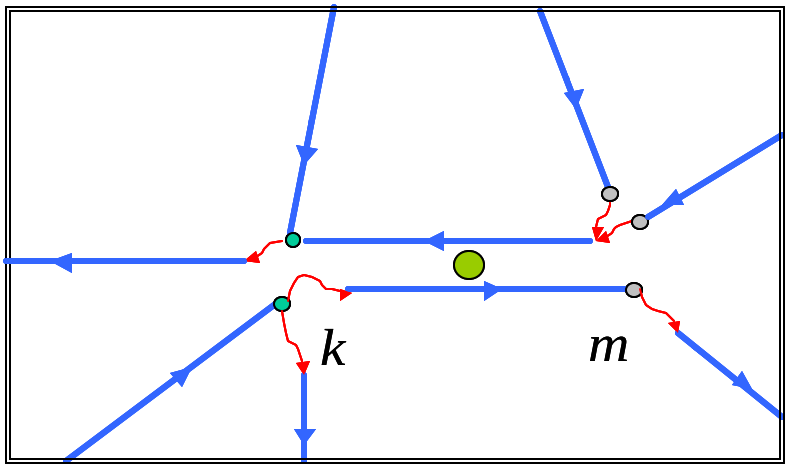

Figure 6.13 Destination stop location

The destination stop treatment is more complex in case of two ways links since vehicles could arrive from both vertices (either upstream or downstream node), and the algorithm is really a node-to-no shortest path procedure. In this case, what is needed is a 1-to-2 shortest path algorithm instead, which is a straightforward extension of the 1-to1 case as discussed next.

Let us start with the 1-to-1 case. The basic algorithm is a label correcting method proposed by Bertsekas (1998). In case of label correcting algorithms, the difficulty is that even after several paths are found to the destination t (each marked by an entrance of t into list V), one cannot be sure that better paths will not be discover later. However, in the presence of additional problem structure, the number of times various nodes will enter V can be reduced considerably, generating a very efficient point-to-point shortest path algorithm in cases where the problem itself provide information to bound the solution and reduce the tree construction spread as explained next.

Suppose that at the start of the algorithm, for each node i, an *underestimate* u_i of shortest path cost from i to t is available ($u_t = 0$ is required). For example, if all

link lengths are nonnegative $u_i = 0$ can be considered for all i. The possibility that $u_i = -\infty$ for all i corresponds to the case where no underestimate is available for the shortest path cost of i, and this case turns out to be the traditional label correcting method for finding a shortest path tree. The algorithm is as follows:

Initially

$$V = \{1\}$$

$$d_1 = 0 \ , \ d_i = \infty, \forall \ i \neq 1 \tag{6.4}$$

The algorithm proceeds in iterations and terminates when V is empty. The typical iteration (assuming V is nonempty) is as follows.

Iteration of the Generic Single Origin/Single Destination Algorithm

Remove a node i from V. For each outgoing arc $(i, j) \in \mathbf{A}$, if

$$d_i + c_{ij} + tc_{p(i)i} < \min\left\{d_j, d_t - u_j\right\} , \tag{6.5}$$

set

$$d_j := d_i + c_{ij} + tc_{p(i)i} \tag{6.6}$$

and add j to V if it does not already belong to V.

In this example d_i is the label associated to node i and $p(i)$ is the predecessor of node *i* through the best path found thus far from 1 to *i*.

The preceding iteration is the same as the typical label correcting from one to all-destinations algorithm, except that the test $d_i + c_{ij} + tc_{p(i)i} < d_j$ for entering a node j into V is replaced by the most stringent test $d_i + c_{ij} + tc_{p(i)i} < \min\left\{d_j, d_t - u_j\right\}$. (In fact, when the trivial underestimate $u_i = -\infty$ is used for all $j \neq t$ the two iterations coincide). To

understand the idea behind the iteration, note that the label d_j corresponds at all times to the best path found thus far from 1 to j. Intuitively, the purpose of entering node j in V when its label is reduced is to generate shorter paths to the destination that pass through node j. If $\mathbf{P}_j$ is the path from 1 to j corresponding to $d_i + c_{ij} + tc_{p(i)i}$, then $d_i + c_{ij} + tc_{p(i)i} + u_j$ is an underestimate of the shortest path length among the collection of paths $\mathbf{P}_j$ that first follow path $\mathbf{P}_j$ to node j and then follow some other path from j to t. However, if

$$d_i + c_{ij} + tc_{p(i)i} + u_j \geq d_t,$$

the current best path to t, which corresponds to d_t, is at least as short as any of the paths in the collection $\mathbf{P}_j$, which have $\mathbf{P}_j$ as their first component. It is unnecessary to consider such paths, and for this reason node j need not be entered in V. In this way, the number of node entrances in V may sharply reduced.

In the context of this study, the algorithm proposed by Bertsekas is very attractive since there is a logical and in most cases very tight lower bound cost u_i for the shortest path cost from 1 to i obtained by computing the all-pairs shortest path matrix under free-flow conditions. Notice that, since turn movement costs are included in the calculation, the resulting matrix will have more than two dimensions, noting that from each node i to any other node there will be as many shortest paths as incoming links to node i.

In addition, as also mentioned by Bertsekas, it is possible to further improve the algorithm proposing an efficient advanced initialization. There is no need for the typical initial conditions in (6.4).

The algorithm works correctly using several other initial conditions. One possibility is to use for each node i, an initial label d_i that is either ∞ or else it is the length of a path from 1 to i, and to take $V = \{i \mid d_i < \infty\}$. A more sophisticated alternative is to initialize V so that it contains all nodes i such that

$$d_i + c_{ij} + tc_{p(i)i} < \min\{d_j, d_t - u_j\} \text{ for some } (i, j) \in \mathbf{A}. \tag{6.7}$$

This kind of initialization can be extremely useful when a "good" path

$$\mathbf{P} = (1, i_1, \ldots, i_k, t) \tag{6.8}$$

from 1 to t is known or can be found heuristically. Then the algorithm can be initialized with

$$d_i = \begin{cases} Length\ of\ \ portion\ of\ \ path\ \mathrm{P}\ \ from\ 1\ to\ i & if\ \ i \in \mathrm{P} \\ \infty & if\ \ i \notin \mathrm{P} \end{cases}$$

$$V = \{1, i_1, \ldots, i_k\}$$

If P is a near-optimal path and consequently the initial value d_t is near its final value, the test for future admissibility into the candidate list V will be relatively tight from the start of the algorithm and many unnecessary entrances of nodes into V may be saved. In particular, it can be seen that all nodes whose shortest distances from the origin are greater or equal to the length of P will never enter the candidate list.

Therefore, and using this idea since the free-flow paths from any node to any other node are all known (for the lower bound calculations), a good path as in (6.8) would be exactly such a path, that is, the same path found from the free-flow all-pairs shortest path matrix computation, and the initial value for a label d_i for a node that

belongs to such a path has to be computed by updating the link and turn movement cost values of the sequence of links going from 1 to *i* through P.

In case of the queue management strategy, a combined SLF (Small label first method) queue insertion and a LLL (Large Label Last method) node removal strategy was utilized, defined as SLF/LLL by Bertsekas (1998). In summary

SLF strategy: Whenever a node j enters queue Q, its label d_j is compared with the label d_i of the top node i of Q. If $d_j \leq d_i$, node j is entered at the top of Q; otherwise j is entered at the bottom of Q.

LLL strategy: Let i be the top node of Q, and let

$$a = \frac{\sum_{j \in Q} d_j}{|Q|}$$

If $d_i > a$, move i to the bottom of Q. Repeat until a node i such that $d_i \leq a$ is found and is removed from Q.

The previous algorithm performs quite well, considering that normally the network costs do not change considerably with respect to the free-flow conditions in the real world, except in peak-hours in which case the algorithm anyway performs better than a two-sided label setting method such as the one proposed by Nicholson.

Recalling the original problem (stop-to-stop), and accounting for cases such as that shown in Figure 6.13, it was needed to extend the previous algorithm to the 1-to-2 case, in order to compute the shortest path to both nodes belonging to the destination link. In addition, the cost of the last turn (from last node t to the next node on the destination link) is added in the formulation in order to capture the turn cost towards the

destination in the middle of a link. Therefore, the iteration of Bertsekas algorithm turns out to be:

For the 1-to-1 case:

Iteration of the Generic Single Origin/Single Destination Algorithm

Remove a node i from V. For each outgoing arc $(i, j) \in \mathbf{A}$, if

$$d_i + c_{ij} + tc_{p(i)i} + \quad _j(t) < \min\left\{d_j, d_t - u_j\right\}, \tag{6.9}$$

set

$$d_j \coloneqq d_i + c_{ij} + tc_{p(i)i} + \quad _j(t) \tag{6.10}$$

and add j to V if it does not already belong to V.

Here,

$$_j(t) = \begin{cases} tc_{it} & \text{if} \quad j = t \\ 0 & \text{otherwise} \end{cases}$$

For the 1-to-2 case: (destinations t_1 and t_2)

Iteration of the Generic Single Origin/Single Destination Algorithm

Remove a node i from V. For each outgoing arc $(i, j) \in \mathbf{A}$, if

$$d_i + c_{ij} + tc_{p(i)i} + \quad _j(t_1) + \quad _j(t_2) < \min\left\{d_j, \max\left(d_{t_1} - u_j(t_1), d_{t_2} - u_j(t_2)\right)\right\} \tag{6.11}$$

set

$$d_j \coloneqq d_i + c_{ij} + tc_{p(i)i} + \quad _j(t_1) + \quad _j(t_2) \tag{6.12}$$

and add j to V if it does not already belong to V.

Condition (6.11) ensures that (6.12) will be set if and only if both bounds are simultaneously fulfilled, associated to t_1 as well as t_2, which means that when V is

empty, the label of both t_1 and t_2 will be the shortest path from node 1 to both destination respectively. It is also assumed that $t_1 \neq t_2$. Lower and upper bounds are defined separately for each destination as well. Therefore, from the resulting shortest path accessing to the destination spot from both vertices, one is able to decide which one is cheaper considering the extra link cost due to the relative position of the point with respect to each vertex. Bertsekas (1998) proved that if there is a path from node 1 to t, his algorithm converges to the shortest path from 1 to t. In the next section, a graphical example showing the advantages of the algorithm under relatively favorable conditions is represented.

6.7.9.3 Graphical example

Let us consider the network in Figure 6.14, and the shortest path for traveling from node 159 to node 165 under free-flow conditions:

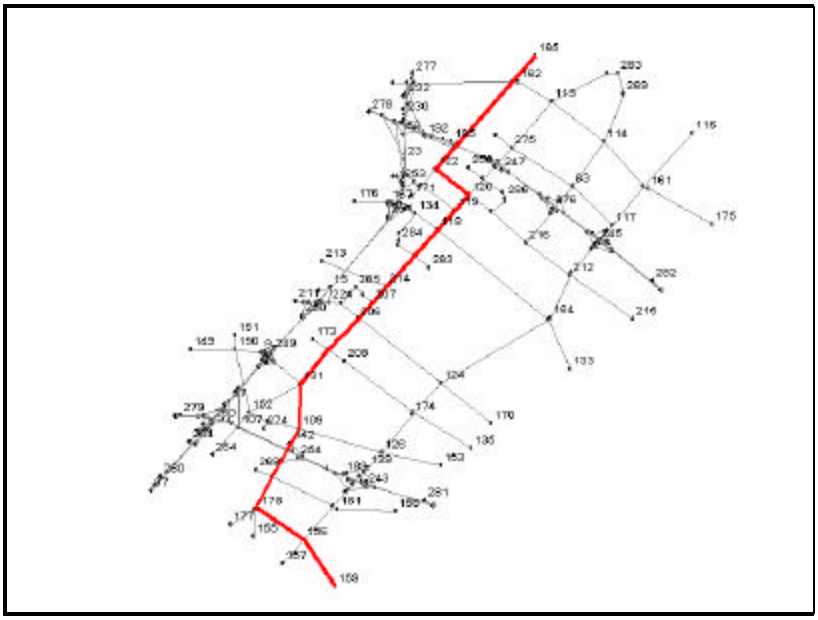

Figure 6.14 Shortest path from node 159 to node 165 (free-flow conditions)

Under these conditions, the all pair shortest path matrix is computed and stored. After loading the network, some congestion occurs, changing the previous shortest path to the one shown in Figure 6.15.

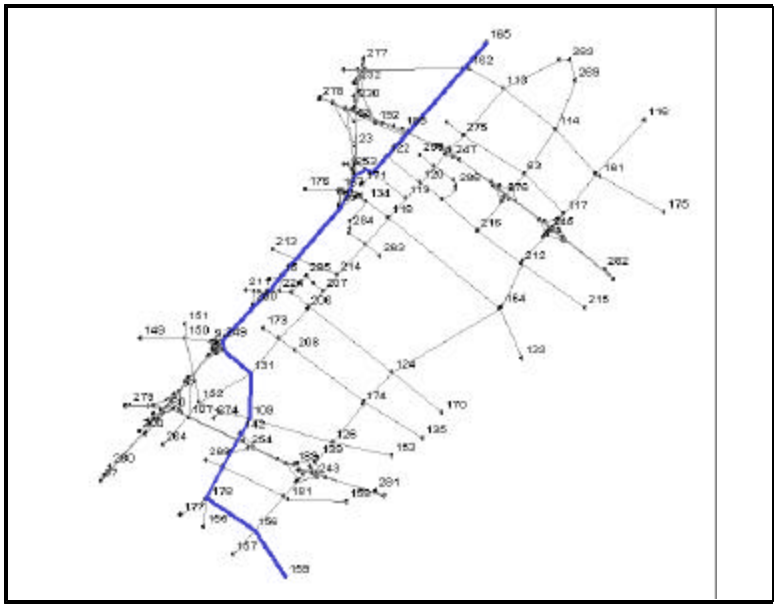

Figure 6.15 Shortest path from node 159 to node 165 (congestion)

The two-sided Dijkstra algorithm was run under the congested conditions without considering previous information about the network itself. The resulting shortest path tree obtained is illustrated in Figure 6.16

By running the proposed algorithm for the same problem but utilizing the free-flow information (Figure 6.14) in the way proposed in the previous subsection, the shortest path tree shown in Figure 6.17 is obtained.Notice the considerable savings observed from the algorithm showing its efficiency in the context of transportation problems, in which the free-flow conditions allow us to significantly bound the limits of the shortest path tree search for finding the optimal solution under congested or simply normal traffic conditions.

The presented algorithm was coded in the simulation platform for finding optimal routing of transit vehicles on the ABSNET network as shown in Figures 6.4 and 6.5

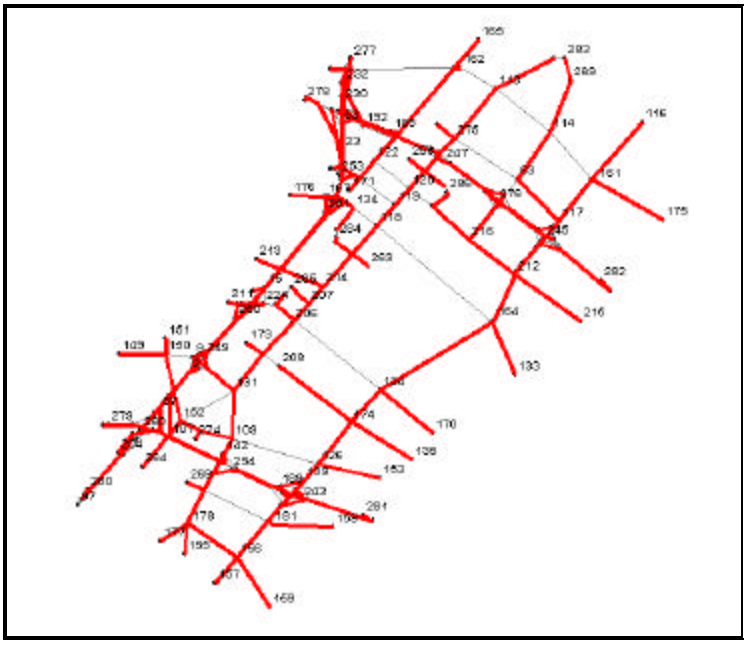

Figure 6.16 Two-sided Dijkstra shortest path tree (congestion)

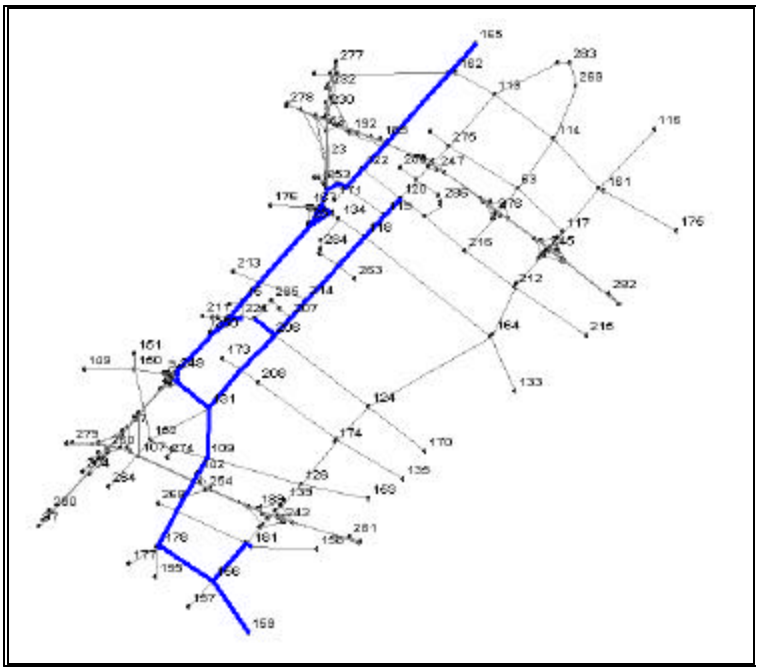

Figure 6.17 Proposed algorithm shortest path tree (congestion)

6.8 Final remarks

This chapter discusses the need for developing more comprehensive urban transportation network simulation environments that go beyond the conventional microscopic simulation models that have largely been auto-centric. The chapter provides also a candidate framework to develop such simulations and presented the application of such a framework for certain simulation contexts.

In particular, the simulation framework has been applied to the *HCPPT* study, allowing the modeler to simulate and evaluate the proposed system under various supply-demand conditions.

The idea relies on the recent interest among practitioners towards the use of micro-simulation and the renewed interest among researchers in considering such simulations as a potentially viable option in the future, especially with commercial software vendors entering the market place in a bigger way than in the past.

Much further work remains to be done. One such area concerns the general-purpose vehicle movement simulations, especially for "unusual" vehicles, such as articulated coaches, tractor-trailer trucks, and even some of the vehicle types proposed in BRT and LRT systems.

The simulation environment as discussed in this chapter would become useful for real-time evaluation of operational plans (as opposed to off-line evaluation). This would require methods to incorporate such systems into even more complicated dynamic traffic assignment models. Suffice it to say that the directions of use of simulation would drive much of future work on this topic.

7 SIMULATED CASE STUDY

7.1 Description of simulation scenarios

The objective of this Chapter is to present a more sophisticated application than that discussed in Chapter 3. The earlier results were at an abstract level based on a spatial discrete-event simulation of the *HCPPT* design. The focus in this chapter is to extract results from a microsimulation of the system based on a real network coded in PARAMICS, following the coding details addressed in Chapter 6. The process requires several critical steps. First, the PARAMICS network of a specific region has to be fed by proper vehicle demand, in order to make the system run, and also in order to generate a parametric passenger demand level according to the methodology described in Chapter 6. Next, the communication interface described in the previous chapter has to be implemented, with all the relatively complex routing rules described in Chapters 4 and 5, and also with the difficulty of routing and controlling vehicles in real simulation time at the microscopic level.

The implementation of the system at this level is computationally complicated and therefore several simplifications and assumptions were required regarding certain specific details, in particular the routing rules for the stochastic case as shown in Section 5.5. In addition, to perform a reasonable simulation of the *HCPPT* scheme, the total area can not be very small given the characteristics of the clustering and decomposed methodology. In this section, all these considerations are addressed in detail.

7.1.1 Network and demand considerations

Paramics Orange County network coding Details

A critical step in the evaluation of the *HCPPT* system is to find a suitable network for simulation. It would not be practical to simulate a small network because most trips would not fit into its boundaries. This would not reflect the demand present in the real world. Therefore, in order to be able to simulate the *HCPPT* system and to obtain reasonable results and statistics associated with its performance, a large network had to be used.

An area that encompasses most of Orange County, California, was chosen for this purpose. An Orange County network in PARAMICS was already coded in the UC Irvine Research Testbed facility, but it contained only the geometrical characteristics of the network (see Figure 7.1).

This network coding was considerably enhanced in the context of this dissertation, the final version composed by 12,691 links and 7,500 nodes. The final version of the network includes several coding details, such as the most important signals, ramps and so on, and run quite well for reasonable demand levels, as explained next. The only difficulty at this stage is to code PARAMICS zones for such a network and generate realistic vehicle demand. The process utilized to deal with this problem is explained in the next sub-section.

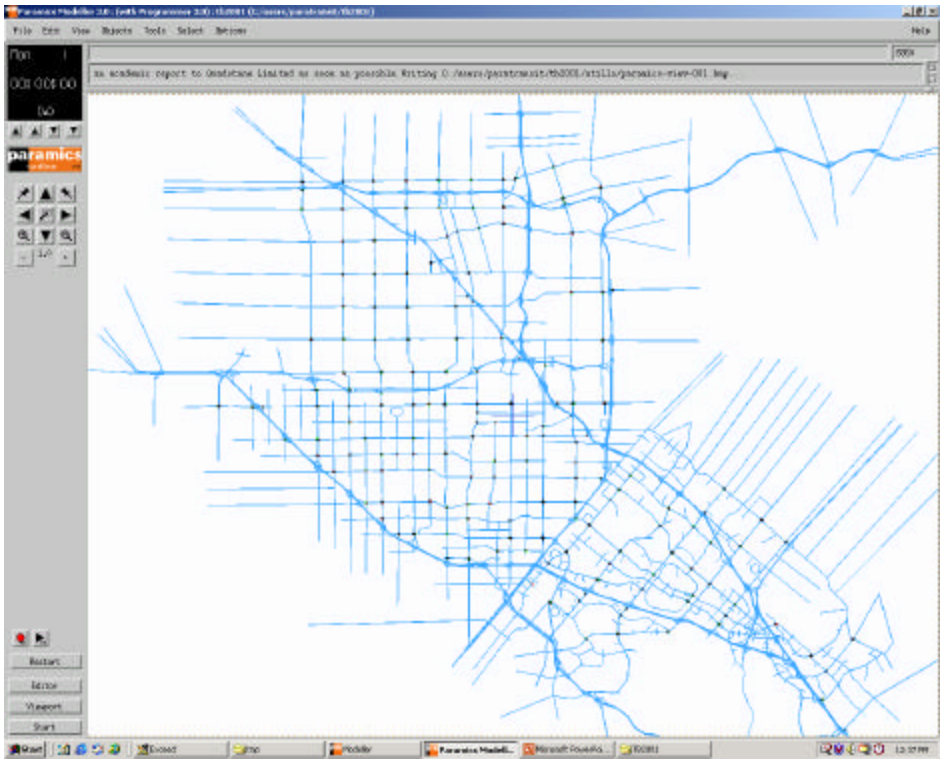

Figure 7.1 Orange County PARAMICS network

Extraction of Orange County Transportation Analysis Model (OCTAM) demand

The demand matrix needed to run the simulation had to be obtained from a planning model called OCTAM (Orange County Transportation Analysis Model). The 1998 version of OCTAM has 1704 zones. This network and its demand were originally coded in TRANPLAN, but were imported into TransCAD because of memory issues. A TransCAD representation of such a network is shown in Figure 7.2.

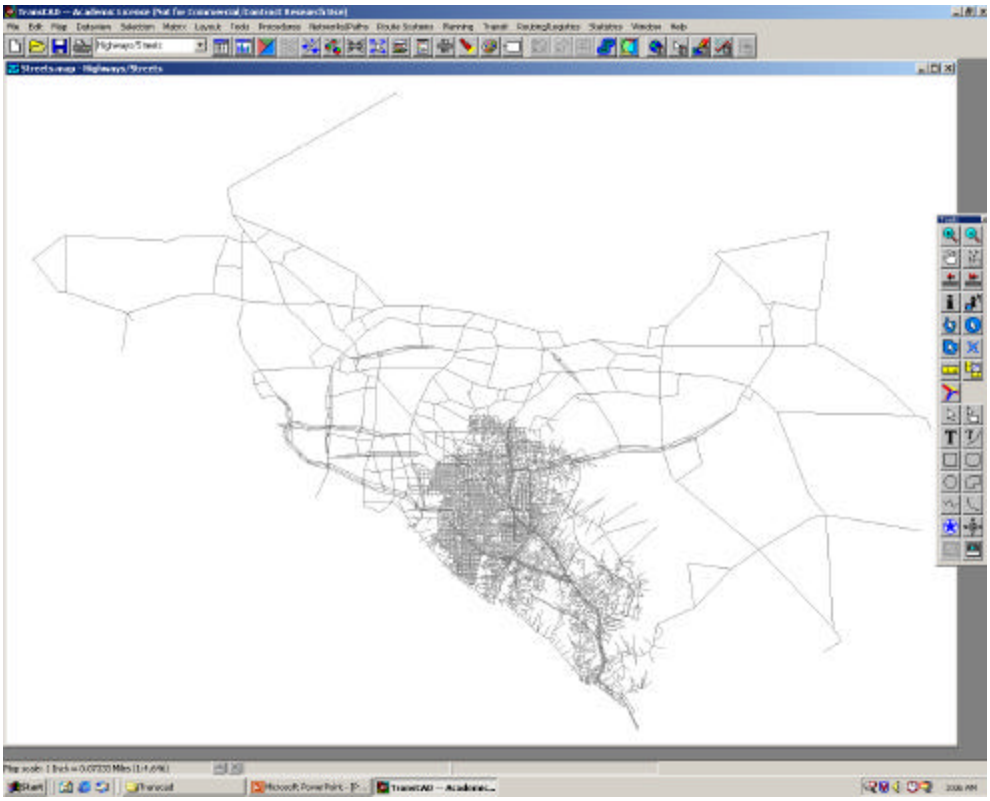

Figure 7.2 Orange County TransCAD network

A sub-area, corresponding to the PARAMICS network, was defined in the OCTAM model and its OD demand was "extracted" in the following steps.

- User equilibrium traffic assignment was performed for the whole network (see Figure 7.3). Upon successful completion, the traffic assignment procedure produced the following output files:
  - A table file containing the estimated link volumes and link costs.
  - A text file containing a summary of user inputs and model outputs.
- The subarea O-D matrix extraction was then performed. The subarea O-D matrix is defined by cross-links and internal centroids. Cross-links are links that cross the cordon line, and internal centroids are centroid nodes that are inside the cordon. An

end-node of a cross-link, whether it is a centroid or a regular node, is called an external station if it is located outside the cordon. A subarea O-D matrix is a square matrix, which means that within the matrix there is a row and a column for each of the internal centroids and the external stations. There are two ways to define a subarea, by defining a polygon or by defining an area set. We used the polygon annotation feature because it was more suited to our requirements. The shape of the subarea was coded to match the existing PARAMICS network. The subarea assignment module was used in order to arrive at the subarea OD matrix. In Figure 7.4, the extracted area at the OCTAM level along with the corresponding PARAMICS network are shown.

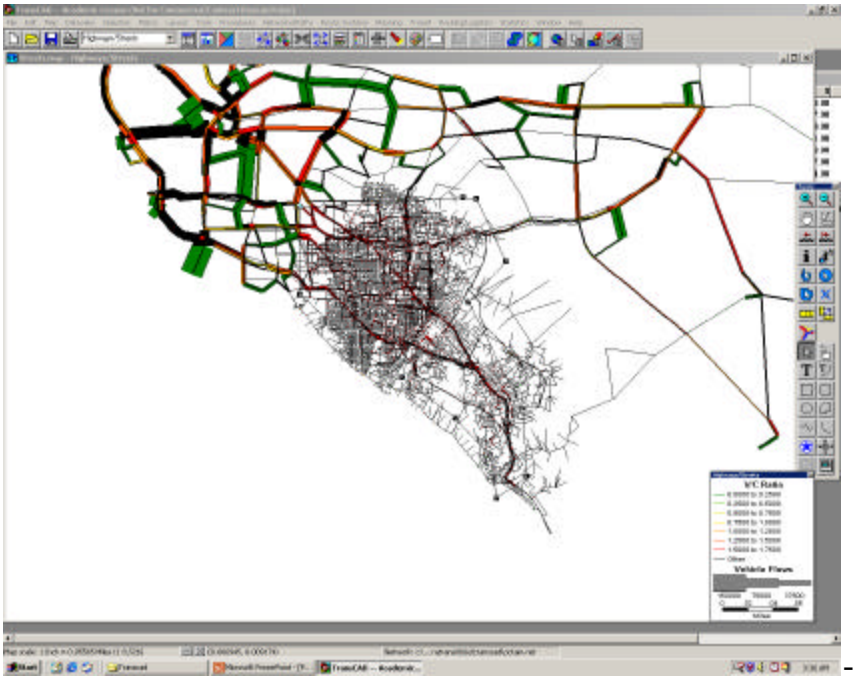

Figure 7.3 User Equilibrium assignment for the Orange County TransCAD network

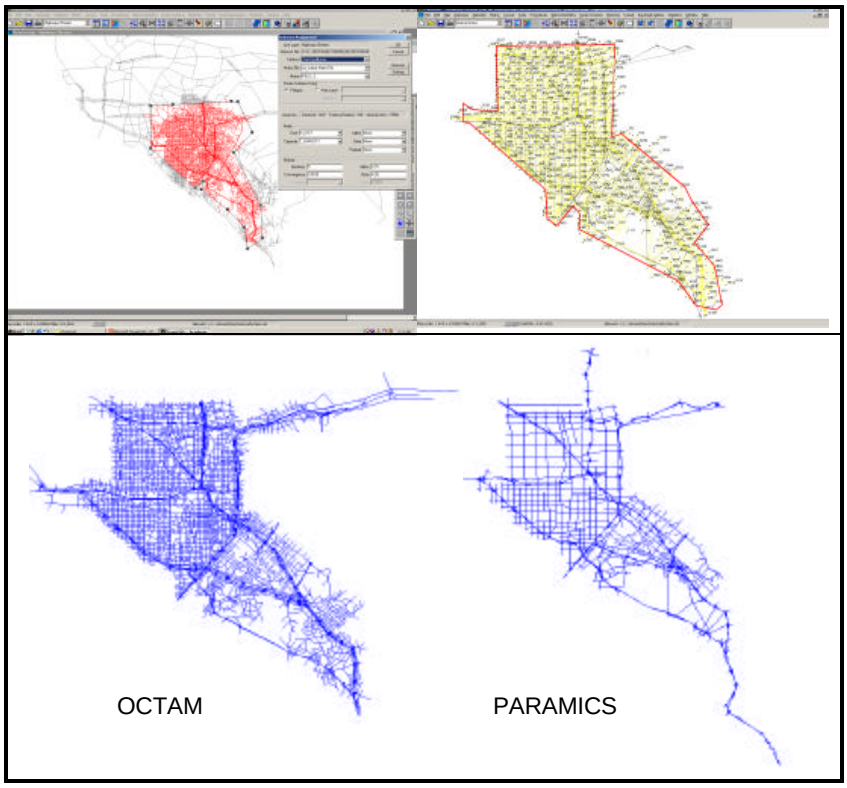

Figure 7.4 Extraction of OCTAM demand

Once the subarea had its corresponding demand, the next step was to match the demand locations of both networks. PARAMICS does not generate demand in centroids, instead the demand is generated uniformly in the links within the boundaries of the origin zone.

Each centroid in the OCTAM sub-area had to be assigned to a zone in the PARAMICS network. The PARAMICS network has a much lower detail than the OCTAM network (see the lower diagrams in Figure 7.3). Therefore each zone in the PARAMICS network had to correspond to more than one OCTAM centroid. After assigning all the OCTAM centroids to their corresponding PARAMICS zones, the demand of the PARAMICS zone was obtained by summing up the demands of all OCTAM centroids associated with it.

As stated above, the levels of detail were not the same in both networks. This made it difficult to find the corresponding locations in both networks and some adjustments had to be made to generate the appropriate demand in the correct location. For example, the level of demand generated in the boundaries of freeways is much higher than the one generated in the arterial streets. We had to make sure that the demand corresponding to the freeways was really entering into the freeways and not being diverted to the nearby arterial streets. These cases had to be identified by running simulations to see if unusually high levels of congestion in arterial streets were obtained.

Another problem encountered at this point was the lack of connectivity in some areas of the network. Sometimes, vehicles could not satisfy the O-D matrix because the origin zone and the destination zone were not connected properly, thus making it impossible for some vehicles to complete their trips. This issue was resolved by modifying the network geometry in PARAMICS. Finally, 269 zones were coded in PARAMICS, including both internal and external demand, from the OCTAM static demand. The final zoning is shown in Figure 7.5.

From the static demand matrix, the PARAMICS model was run for different levels of demand until an 80% level, recognizing that the static demand always overestimate the dynamic one, that is appropriate for a microsimulation model such as PARAMICS. In the next subsection, all *HCPPT* design considerations applied to the aforementioned Orange County region are explained, based on the PARAMICS representation as in Figures 7.1 and 7.5.

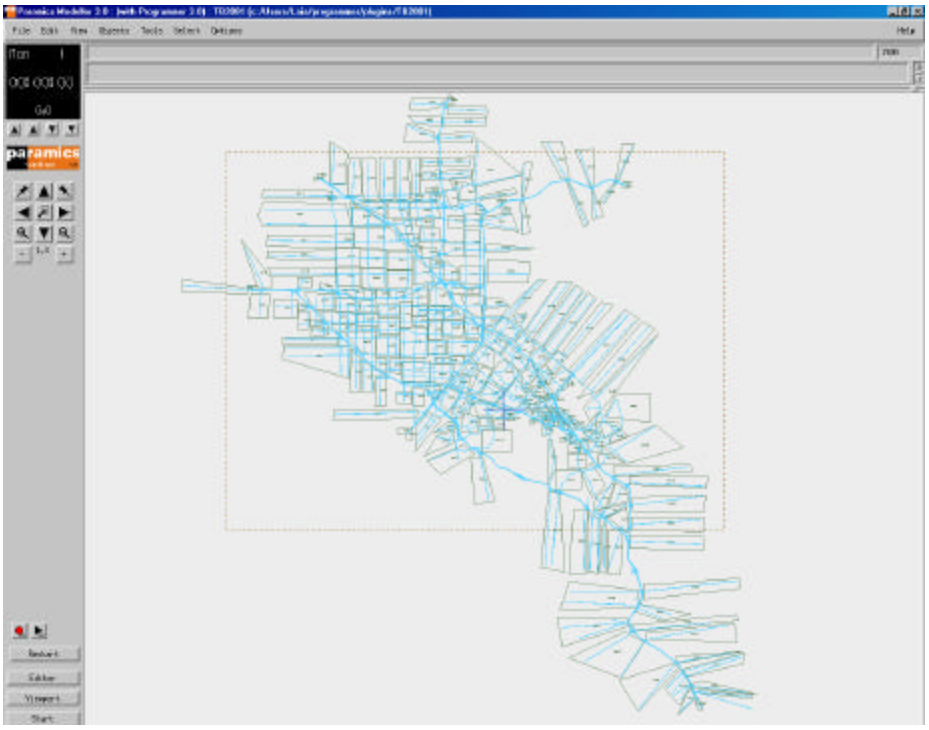

Figure 7.5 Orange County PARAMICS network zoning

7.1.2 Design considerations

7.1.2.1 Design of clusters and hub locations

The total area is around 245 square miles. Therefore, the total region was split into five hub areas according to the *HCPPT* design and dimensions explained at length in Chapter 3. The criteria used in such division were mainly motivated by topological issues. In fact, the idea behind the proposed clustering was to effectively balance the total length of roads within each hub area. Each cluster area is 40.92 square miles; therefore each of the cells is 5.84 square miles.

In addition, the distribution of the hub areas was chosen trying to maximize the coverage of the road system, constrained to the specific hexagonal shape of the cells.

Once, the best distribution was properly chosen, some additional regions (cells but of different shape) were added to the system and associated to the closest hub area.

In summary, 35 hexagonal cells are coded, in groups of 7 comprising each hub area. In addition, 7 extra regions are added to the scheme in order to cover the whole road system, and associated to the closest hub area for routing purposes, according to the clustering distribution shown in Figure 7.6.

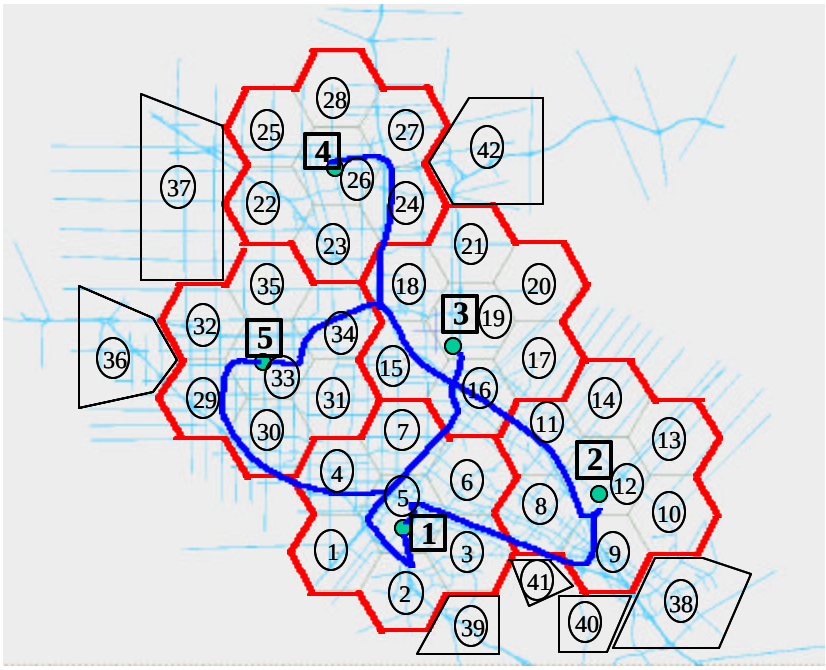

Figure 7.6 ***HCPPT*** **scheme on the Orange County region**

The hub locations was determined in such a way that they approximately matched the middle point of each hub region. Notice that in all cases, the hub terminal is located in the central cell of the corresponding hub area.

The specific design of each terminal was quite simple as shown in Figure 7.7.

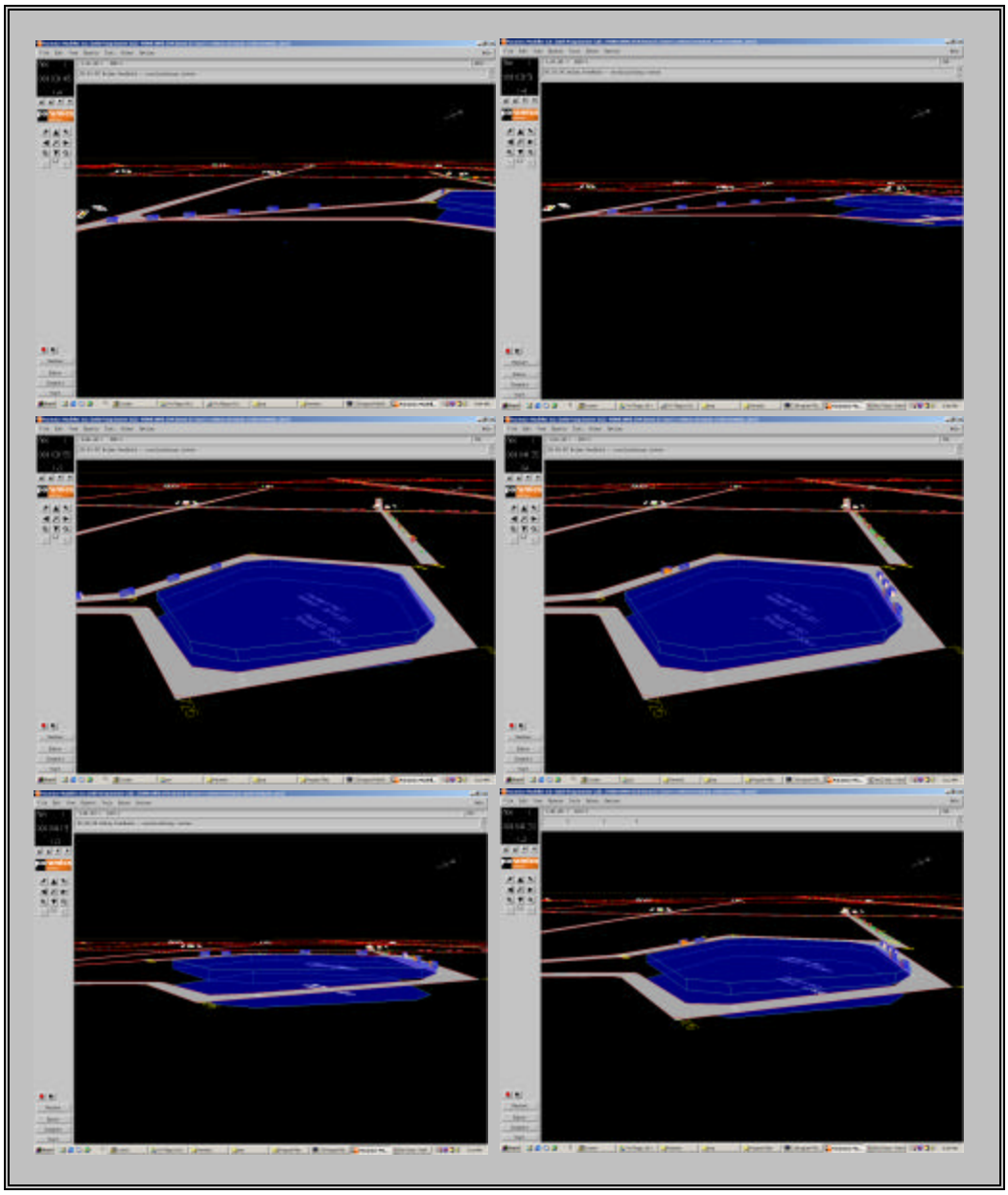

Figure 7.7 *HCPPT* hub terminal design

Notice that in the terminal, vehicles move clockwise around a terminal in order to perform all passenger transfer operations to the right-side curb. Thus the passengers can walk inside the terminal to the appropriate curb locations marked for vehicles destined to other destinations. Though such details may appear insignificant, lack of care in these aspects can cause serious passenger disutility. The above design requires a two-level

crossing at the entrance to prevent vehicle conflicts and also in order not to add extra delays to the passengers from the formation of vehicle queues at the terminal entrance.

The number of stop spots coded in PARAMICS was enough in order to guarantee at least one unoccupied spot for a random vehicle entering the terminal at any time. In addition, the sites were grouped and distributed based on the destinations of the vehicle, trying to better represent the real operation of urban bus terminals.

Terminal access was in most cases controlled by a signal, on an important arterial on the surface network.

7.1.2.2 Treatment of trunk and surface street networks

With regard to the implementation of the trunk and surface network concept, introduced in Chapter 3, in order to differentiate the reroutable from the non-reroutable portion of the *HCPPT* vehicle route, some important issues must be highlighted:

- The trunk portion of the network in most cases was chosen along the freeways (see Figure 7.6), although a freeway section is not necessarily the shortest path between two adjacent hubs, particularly under congested traffic conditions. However, the design and the experiments were carried out restricting the trunk network to the freeway sections shown in Figure 7.6.
- The entrance and exit locations, to and from the trunk portion of the route are chosen depending on the specific location of the vehicle when taking the decision of entering the trunk network. These are normally based on the shortest path between the current location and the trunk network. Some constraints were imposed

however, restricting the farthest allowable entrance point when the vehicle is moving towards its visit hub, and the closest allowable exit point when it is returning in direction to its home hub. These are again practical considerations of significant impact in the operation of real systems like this.

- The reroutable portion within each zone is composed by all arterial (surface) streets of the network, according to the clustering configuration shown in the previous section. In terms of roads, a representation of the road network split by hub area is depicted in Figure 7.8.

  In addition, the parametric generation of passenger demand from the PARAMICS zoning shown in Figure 7.5, is constrained to this set of roads. An additional constraint is added to this generation process regarding the network connectivity constraint mentioned in Section 6.7.7. That is, demand points can not be generated in places from where the vehicle can not physically return to the network. This issue is very important and it has to be checked with special care in order not to have serious problems while running the simulation, causing spurious results that may often be undetectable too.

  The length of the road system, for each cell and hub is summarized in Table 7.1. Even though the densest hub area is hub 1 in terms of road length, most of the links included in the calculation corresponds to zone connectors, and therefore are not interesting for routing and demand generation purposes. Hubs 1, 4 and 5 are quite similar. The only hub with a considerable difference in road density is hub 3, which is evident from all previous figures. This fact must be taken into account when deciding fleet distribution among hubs, as discussed later in this chapter.

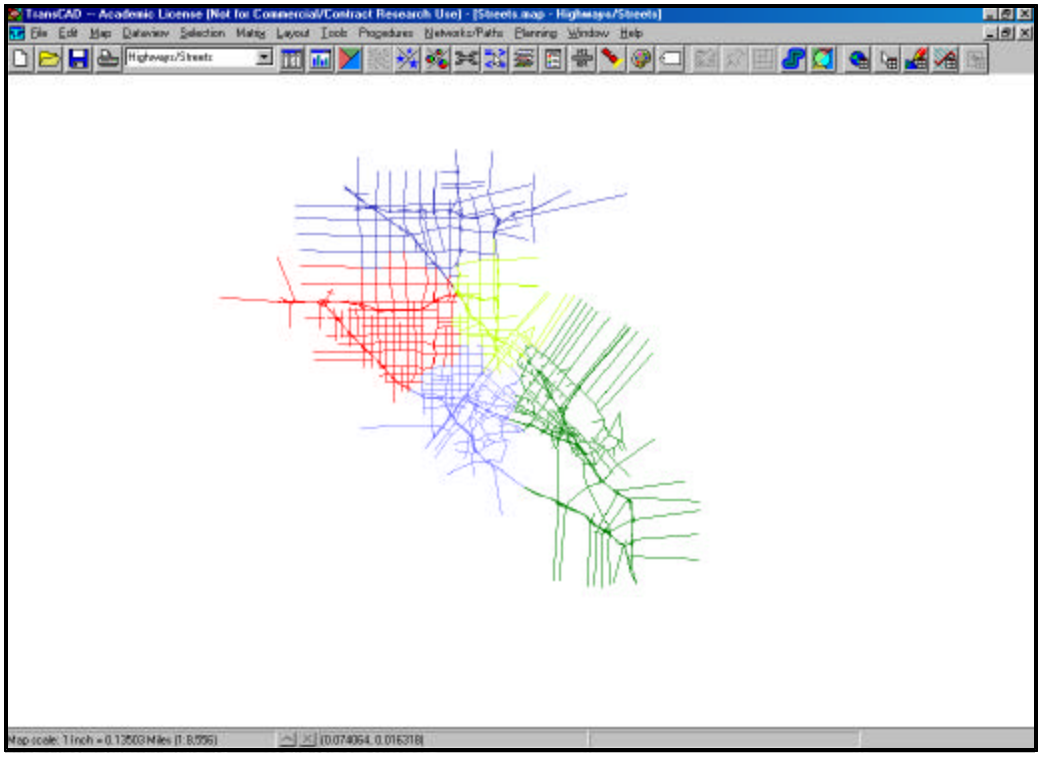

Figure 7.8 Reroutable portion of vehicle routes within each hub area

- After building the PARAMICS modeling network, and according to the network aggregation methodology discussed in Chapter 6, the corresponding simplified ABSNET is created. The ABSNET has 5,844 links and 2,994 nodes, meaning that the original microsimulation network was reduced by approximately 54% in terms of links, and about 60% in terms of nodes. The corresponding data structures for network storage are kept as linked lists for modeling and routing purposes.

Table 7.1 Length of road system at cell and hub level (meters)

	HUB 1	HUB 2	HUB 3	HUB 4	HUB 5
Total Length	562541.7	915120.61	374024.8	681305.1	640311.3
CELL					
1	65829.9	-	-	-	-
2	62231.9	-	-	-	-
3	58421.9	-	-	-	-
4	98521.4	-	-	-	-
5	116137	-	-	-	-
6	61070.3	-	-	-	-
7	71140	-	-	-	-
8	-	96339.7	-	-	-
9	-	108189	-	-	-
10	-	44771.5	-	-	-
11	-	78283.9	-	-	-
12	-	65233.7	-	-	-
13	-	42393.6	-	-	-
14	-	50709.3	-	-	-
15	-	-	78202	-	-
16	-	-	91477.1	-	-
17	-	-	40650.1	-	-
18	-	-	73882.6	-	-
19	-	-	38788.9	-	-
20	-	-	19936.8	-	-
21	-	-	31087.3	-	-
22	-	-	-	61298.8	-
23	-	-	-	64879.9	-
24	-	-	-	46434.5	-
25	-	-	-	38568.6	-
26	-	-	-	73110.3	-
27	-	-	-	60124.4	-
28	-	-	-	24433.6	-
29	-	-	-	-	88493.5
30	-	-	-	-	88144.3
31	-	-	-	-	56587.7
32	-	-	-	-	68844.3
33	-	-	-	-	82302.8
34	-	-	-	-	67840.5
35	-	-	-	-	39208.2
36	-	-	-	188127	-
37	-	-	-	-	148890
38	-	398620	-	-	-
39	29189.3	-	-	-	-
40	-	22571.7	-	-	-
41	-	8008.21	-	-	-
42	-	-	-	124328	-

7.1.3 Design of a *HCPPT* system under specific passenger demand conditions

After discussing all details about the vehicle demand generation, *HCPPT* clustering and network issues, the next issue is to generate proper passenger demand according to the methodology developed in Section 6.7.3. A modal split of 5 % was considered here for passenger demand in the system, and the corresponding passengers were generated over the road network at specific pick-up and delivery points. The passengers were generated somewhere along the network links, avoiding banned street stretches, such as very close to the intersections, places without connectivity, etc. The time between calls was assumed to be uniform over a period of one hour.

The passenger pooling process at stops was simulated using a Poisson distribution with $\lambda = 1.5$ as shown in Figure 6.8. All internal trips either traveling alone (pool size = 1) or not sharing pick-up location with any external traveler going to a different hub region were eliminated from the system, assuming that they could be served by an internal service running as a typical *DRT* within each area. The reason is that originally too many trips turned out to be internal, generating evident inefficiencies in the scheme operation as proposed here.

After making this correction, a reasonable level and distribution of passengers asking for service was generated. For example, in Table 7.2 and Figure 7.9 the distribution of the pick-up points in time and space is shown at a cell level.

Notice that there are zones where the demand is quite high, particularly those cells that belong to hub 5. Notice that even though cluster areas are quite similar in terms

of total road length as well as total surface, the demand levels are not very balanced among them. Therefore, both the distribution of vehicles among hubs and the distribution of the visit hubs among those vehicles have to be computed using a reasonable methodology based on the information available regarding the potential passenger demand shown above, that could be associated to some available historical data.

In order to do that, a very simple fleet sizing model is proposed, based on the expected level of service measured as a function of the shortest path travel time associated to each trip over time, and also as a function of a minimum desired average load. Moreover, the idea is to estimate the number of vehicles assigned to each hub but going to each adjacent hub. However, there are some customers traveling inside the same area (internal trips) and others going to a hub that is not necessarily adjacent to their home hub. In such cases, as a rule of thumb, the required vehicles for serving that portion of the demand are distributed evenly among all adjacent hub areas.

Table 7.2 Passenger demand level and distribution (call/cell)

CELL	TIME INTERVAL [min]				
	[10,20]	[20,30]	[30,40]	[40,50]	[50,60]
1	27	28	26	19	21
2	19	28	27	25	26
3	20	14	15	12	19
4	37	45	54	31	34
5	20	20	6	11	13
6	21	12	18	15	20
7	33	24	28	20	34
8	28	35	27	46	26
9	11	1	6	0	1
10	19	10	11	5	10
11	33	27	19	16	19
12	23	14	7	7	7
13	13	0	0	0	0
14	14	0	0	0	0
15	61	27	44	44	44
16	51	31	24	36	36
17	43	17	20	18	20
18	38	13	20	18	18
19	25	6	5	7	8
20	20	0	0	0	0
21	28	14	10	7	9
22	34	25	12	11	14
23	39	15	23	17	22
24	38	11	14	10	11
25	37	10	18	12	18
26	51	23	19	19	25
27	28	1	2	1	2
28	28	0	0	0	0
29	59	32	36	25	33
30	70	42	41	40	29
31	76	53	33	43	57
32	56	18	19	23	15
33	72	33	34	39	38
34	65	32	40	43	42
35	74	20	32	39	39
36	57	22	18	21	21
37	44	9	8	10	12
38	86	65	50	48	43
39	39	1	0	2	1
40	42	4	7	2	9
41	41	0	0	0	0
42	45	5	4	5	4

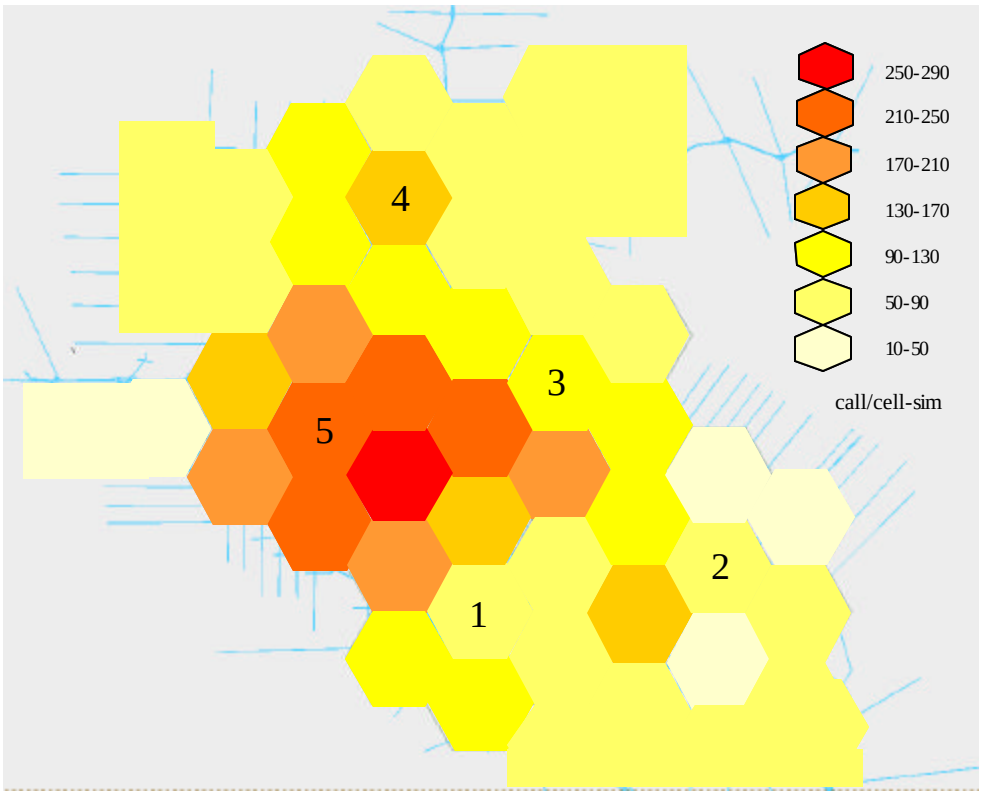

Figure 7.9 Passenger demand level and distribution

Finally, the fleet size associated with each hub area is estimated using a very simple rule as follows:

$$NV_{ij} = \frac{\rho \sum_{k \in T_{ij}} spt_k^{ij}}{DL \cdot P} + \frac{1}{N_H(i)} \left\lceil \frac{\rho \sum_{k \in T_{ii}} spt_k^{ii}}{DL \cdot P} + \sum_{l \notin H(i)} \frac{\rho \sum_{k \in T_{il}} spt_k^{il}}{DL \cdot P} \right\rceil = NV_{ij}^{d} + NV_{ij}^{i} \qquad (7.1)$$

or equivalently

$$NV_{ij} = \frac{\rho}{DL \cdot P} \left\lceil \sum_{k \in T_{ij}} spt_k^{ij} + \frac{1}{N_H(i)} \left(\sum_{k \in T_{ii}} spt_k^{ii} + \sum_{l \notin H(i)} \sum_{k \in T_{il}} spt_k^{il} \right) \right\rceil \qquad (7.2)$$

where

- spt_k^{pq}: is the shortest path travel time for traveler k going from origin hub area p to destination hub area q (secs.).
- T_{pq}: is the total demand over period P between hub area p and hub area q.
- ρ: represents a correction factor for the shortest path travel time (at this point assumed invariant between hubs), which must be greater than one, approximately capturing the expected *ride time* index defined in Chapter 3, as the ratio between the system ride time to door-to-door travel time by automobile, which is assumed to be close to the shortest path travel time.
- $H(i)$: is the set of adjacent hubs to hub i, with cardinality $N_H(i)$.
- DL: represents the desired load (pax/veh).
- P: is the total period of modeling (secs.)

Notice that this way of estimating the fleet size is an approximation, that only gives us a reasonable idea of what would be the appropriate fleet to manage the historical demand for getting some minimum desired load, assuming certain value for the level of service, captured by the indicator ρ. In Table 7.3, the previous calculation is summarized for a hourly demand pattern (P = 3600), assuming that DL = 3 and ρ = 1.5. The total number of vehicles assigned to each hub is shown in Figure 7.10.

Table 7.3 Estimation of the fleet size and distribution

Hub i	***Hub j***	***Hub not reach***	$\sum_{k \in T_{ij}} spt_k^{ij}$	NV_{ij}^{d}	$NV_{ij}^{d} + NV_{ij}^{i}$	$round(NV_{ij})$	NV_i
1	1		167123	23.2			
1	2	4	235447	32.7	46.3	47	
1	3		198279	27.5	41.1	42	
1	4		125649	17.5			
1	5		369530	51.3	64.9	65	
							154
2	1	4	459309	63.8	98.3	99	
2	2		124540	17.3			
2	3		170050	23.6	58.1	59	
2	4		150701	20.9			
2	5		221832	30.8			
							158
3	1		212882	29.6	32.5	33	
3	2		93965	13.1	16.0	17	
3	3		85111	11.8			
3	4		123861	17.2	20.2	21	
3	5		344901	47.9	50.9	51	
							122
4	1		109366	15.2			
4	2		91216	12.7			
4	3		154514	21.5	44.6	45	
4	4		132445	18.4			
4	5	2	328815	45.7	68.8	69	
							114
5	1		515203	71.6	95.0	95	
5	2		209049	29.0			
5	3		450995	62.6	86.1	87	
5	4	2	461269	64.1	87.5	88	
5	5		296685	41.2			
			167123	23.2			270

It is evident the relation between the demand and the required fleet size by looking at Figures 7.9 and 7.10. Hub 5 represents the most intensive focus of demand, and that is reflected on the fleet size of 270 vehicles, which is around twice as much as that assigned to the other hubs.

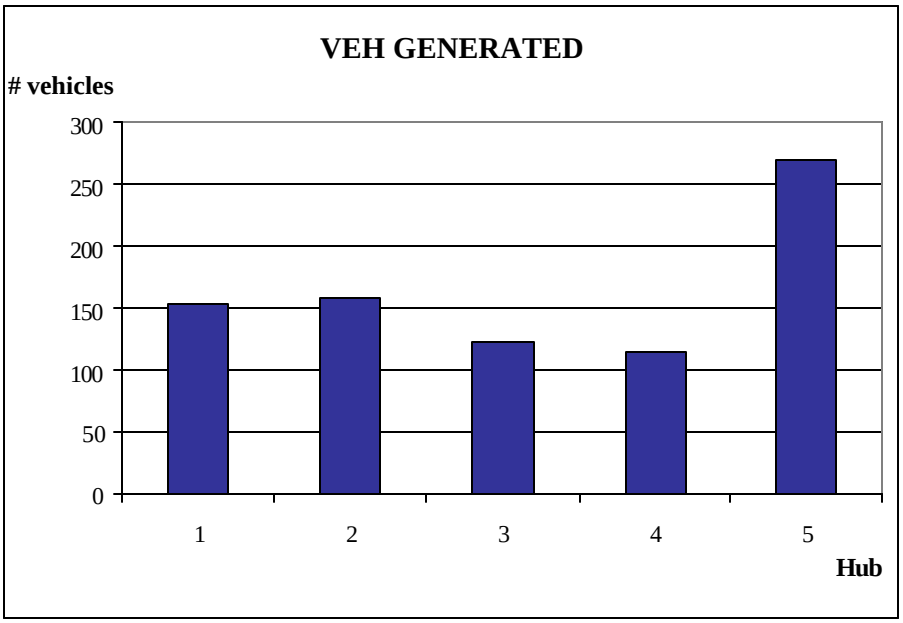

Figure 7.10 Fleet size distribution per hub

The computed fleet distribution and size is used in most of the experiments in Section 7.2. However, and in view of the results obtained in some of the simulations conducted as a first approach, some sensitivity analysis was performed based on variation of the fleet size trying to keep the above showed distribution.

In the next section, a description and classification of the simulation experiments is described, addressing the different aspects, parameters and policy indicators that were modified and tested in Section 7.2.

7.1.4 Specific routing and scheduling rules applied to different cases

In this subsection, a discussion about the specific rules and scenarios simulated as part of this research is described. The analysis is based on the methodology described in the previous chapters (mainly, 4 and 5), but always based on the basic design proposed in Chapter 3.

In Chapter 4, a set of rules are described in order to route the *HCPPT* vehicles, however, although all these rules are based on optimality issues and common sense, some details and practical issues had to be studied via simulation. Therefore, several preliminary experiments were conducted, as a first approximation in order to bind the range of some parameters used as part of the rules. Other parameters were guessed from economic theory or previous experience found in the literature. As results, a reasonable set of simulations based upon different parameters, rules and policies were obtained. As mentioned in the previous section, a sensitivity analysis was performed oriented to the supply level more than the demand. Parametric demand studies as well as specific demand models resulting from different level of service will be explored in the future and is out of the scope of this dissertation.

Before describing the experiments conducted by applying the theory developed in all previous chapters, it is necessary to mention some practical issues to make the system run. Some of them are the following:

- The simulation had to be run with relatively low demand levels (for personal vehicles), due to computational restrictions from the network size, number of vehicles and API complexity.
- All routing rules are updated every minute of simulation. This means that all new calls are accumulated in a queue of requests during intervals of one minute. After this time, calls are assigned to vehicles according to the specific routing rules in a FIFO manner. If some requests can not be served, they are added to a waiting list (queue), with a capacity of 1000 customers. If the queue exceeds this value,

additional customers are rejected. Most of the routing decision rules are computed using a matrix of shortest paths (computed from an all-to-all shortest path algorithm updated every 15 minutes) instead of running the point-to-point shortest path algorithm every time a route segment travel time has to be evaluated in the decision procedure. The point-to-point shortest path algorithm is run once the decision of assignment is taken (that is, once the vehicle and the insertion segment on that specific vehicle route are decided) in order to check the quality of the decision and also in order to effectively reroute de chosen vehicle.

- In order not to have extremely high computational delays when running the insertion algorithm, some constraints are added when defining the feasible set of vehicles and candidate segments for a specific insertion. Particularly, a parameter MAX_DEL_CAN is defined, such that, if a specific vehicle has more than MAX_DEL_CAN scheduled external deliveries (or internal deliveries already on the vehicle) at certain decision time, it will not be candidate for serving any extra pick-up. In addition, in order to reduce the length of delivery tours, a constraint on the number of people picked up at hubs is imposed, called MAX_PICK_HUB, even though the vehicle could be able to accommodate more passengers inside. After the preliminary experiments, a value of MAX_DEL_CAN = 3 and a value of MAX_PICK_HUB = 5 were considered for the final simulations.
- For each new pick-up request, only a subset of vehicles is considered as candidate for that insertion. The selection of such a set of candidates is based on a spatial consideration, since it is very easy to check. The heuristic rule is as follows: A vehicle will be considered a candidate for picking up a new group of passengers,

only if any of the next MAX_SEQ_DIST scheduled stops on its route is close enough to the origin of the pick-up request. The closeness is defined exclusively in terms of the grid distance between the pick up point and the analyzed stop. After several experiments, MAX_SEQ_DIST = 2, and maximum allowable distances of 4,000 meters (in both x and y axis) were considered suitable in terms of the trade off running time of the algorithm versus quality of the solution.

- The total running time was of 1:30 hrs of simulation. Vehicles are generated at the beginning of the simulation, and the demand is 1 hr. demand and start at t = 10 min. Vehicle statistics are measured from 0:10 to 1:15. The first 10 minutes are considered warm-up time and the last 15 are not included in the vehicle performance measures since passenger demand generation is stopped during the last 20 minute period. Simulation time constraints are basically due to computational restrictions.
- Stochastic parameters and methodology were also simplified due to computational constraints. For example, the binary decision models were calibrated offline in this first stage. In future implementations, it is possible to calibrate models online, updating them every time the predicted level of service values (basically expected travel and waiting time, and expected load) do not reproduce the observed values within certain threshold.

In addition to the above considerations, it is also pertinent to set some of the basic parameters, based on previous experience and also as obtained from the preliminary simulations. In Table 7.4, the basic parameter values for the final simulations are reported:

Table 7.4 Basic parameters used in final simulations

Parameter	Value
ATT_1	1
ATT_2	2
ACT_1	1
ACT_2	3
$TR1$	900
AWT_1	2
AWT_2	6
AWT_h	3
P_F	0
P_V	0
DTA_l	60
DTB_l	660
OC	1

These parameters are as defined in Chapter 4. Basically, they correspond to the weight given to the travel and waiting time in the cost formulation at different stages of the customer trip. Also, the weights associated to the $TR_j(k)$ component are shown. Time windows at the pick-up point (Section 4.3.1.1) are also guessed for all customers. The operator cost weight is assumed the same as that of the travel time in status 1. Dimension analysis is implicit in the formulation of Chapter 4.

In addition, given the complexity of the stochastic modeling, most of the simulation scenarios were based on the deterministic methodology. The stochastic effect was incorporated over the best deterministic scenarios to measure the incremental effect of the stochastic process after setting some basic rules and parameters based upon some practical heuristics such as those mentioned in Section 4.6.

A summary of the strategies developed is shown in Table 7.5.

Table 7.5 Summary of final simulations

	SIMULATION #										
VARIABLE	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***
eps1	900	900	900	900	900	900	300	900	1200	1200	1200
eps2	300	300	300	300	300	300	300	300	300	300	300
repositioning	0	0	0	0	0	0	0	0	0	0	0
max_seq_dist	2	2	2	2	2	2	2	2	2	2	2
max_pick_hub	4	4	4	4	4	4	4	4	4	5	5
max_del_can	3	3	3	3	3	3	3	3	3	3	3
max_wait_list	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
MAXXC	3000	5000	7000	3000	5000	5000	5000	5000	5000	5000	3000
MAXYC	3000	5000	7000	3000	5000	5000	5000	5000	5000	5000	3000
Max_rer_time	600	600	600	600	600	600	600	600	600	600	600
TLIM	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200
MIN LF	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
PLMAX	3	3	3	3	3	3	3	3	3	3	3
status_veh_rer_time	1	1	1	1	1	1	1	1	1	2	2
pick_state2	0	0	0	1	1	1	1	1	1	0	0
Max_Sim_Time_Veh	4500	4500	4500	4500	4500	4500	4500	4500	4500	4500	4500
time_bet_stops								480	480	360	360
Min_Veh_Rer_Rate										0.5	0.5
gamma_A											
gamma_T											
theta_PI											
min_adj											
veh removed											
Run type	D	D	D	D	D	D	D	D	D	D	D

Table 7.5 (contd) Summary of final simulations

	SIMULATION #										
VARIABLE	***12***	***13***	***14***	***15***	***16***	***17***	***18***	***19***	***20***	***21***	***22***
eps1	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200
eps2	300	300	300	300	300	300	300	300	300	300	300
repositioning	0	0	0	1	1	1	0	1	1	1	1
max_seq_dist	2	2	2	2	2	2	2	2	2	2	2
max_pick_hub	5	5	5	5	5	5	5	5	5	5	5
max_del_can	3	3	3	3	3	3	3	3	3	3	3
max_wait_list	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
MAXXC	5000	3000	4000	4000	4000	4000	4000	4000	4000	4000	4000
MAXYC	5000	3000	4000	4000	4000	4000	4000	4000	4000	4000	4000
Max_rer_time	600	600	600	600	600	600	600	600	600	600	600
TLIM	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200
MIN LF	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
PLMAX	3	3	3	3	3	3	3	3	3	3	3
status_veh_rer_time	2	2	2	2	2	2	2	2	2	2	2
pick_state2	1	0	1	1	1	1	1	1	1	0	0
Max_Sim_Time_Veh	4500	4500	4500	4500	4500	4500	4500	4500	4500	4500	4500
time_bet_stops	360	360	360	360	360	360	360	360	360	360	360
Min_Veh_Rer_Rate	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
gamma_A							1				
gamma_T							2				
theta_PI							-1				
min_adj							0				
veh removed	50				50			75	100	75	100
Run type	D	D	D	D	D	D	D	D	D	D	D

Table 7.5 (contd) Summary of final simulations

	SIMULATION #										
VARIABLE	***23***	***24***	***25***	***26***	***27***	***28***	***29***	***30***	***31***	***32***	***33***
eps1	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200
eps2	300	300	300	300	300	300	300	300	300	300	300
repositioning	0	1	1	1	1	0	0	1	1	1	1
max_seq_dist	2	2	2	2	2	2	2	2	2	2	2
max_pick_hub	5	5	5	5	5	5	5	5	5	5	5
max_del_can	3	3	3	3	3	3	3	3	3	3	3
max_wait_list	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
MAXXC	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000
MAXYC	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000
Max_rer_time	600	600	600	600	600	600	600	600	600	600	600
TLIM	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200
MIN LF	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
PLMAX	3	3	3	3	3	3	3	3	3	3	3
status_veh_rer_time	2	2	2	2	2	2	2	2	2	2	2
pick_state2	0	1	1	1	1	0	0	1	1	1	1
Max_Sim_Time_Veh	4500	4500	4500	4500	4500	4500	4500	4500	4500	4500	4500
gamma_A	1	1	1	1	1	1	1		1	1	
gamma_T	2	2	2	2	2	2	2		2	2	
theta_PI	-1	-1	-1	-1	-1	-1	-1		-1	-1	
min_adj	0	0	0	0	0	0	0		0	0	
veh removed			50	100						75	50
Run type	D	D	D	D	D	D	D	S	S	S	S

Originally, more than 50 simulations were conducted. In the next section, only the best 30 are shown with some detail. All strategies and parameters used here were carefully tested in order to have an idea of the impact of each small rule on the overall system performance. The first two columns of the table are the values assumed for the parameters $_1$ and $_2$ added to the cost function in (4.28) and (4.52). The row repositioning indicates if the vehicle repositioning strategy is implemented or not. Most of the other parameters were already explained. MAXXC and MAXYC corresponds to the maximum search area for deciding candidate segments and vehicles (finally, after the first experiments, it was set at 4000 meters). The last four parameters are associated to the queuing strategy summarized in Section 4.6.1.1.

Status_veh_rer_time is a parameter that identifies the strategy utilized in the transition between not-reroutable and reroutable portions of the vehicle trip. If this parameter is equal to 1, means that the first strategy in Section4.6.2.2 is implemented. If the value is 2, it refers to the second strategy, involving a critical value of the cumulative time on board $TR_j(k)$ in the decision. This last strategy performs much better than any other, and that is why it was used in most experiments.

Pick_state2 is a parameter that indicates if the vehicle stays as a candidate for picking-up new customers while it is moving towards the trunk portion of the route, in which case this factor is 1, 0 otherwise.

For sensibility analysis, a percentage of the fleet it was removed, proportionally to the fleet distribution discussed above in Section 7.1.3. The number of vehicles so

removed for sensitivity studies are shown in one row of the table for each simulation run. Run type D is for deterministic cost function and type S for stochastic runs.

7.2 Simulation Results on the Impacts of *HCPPT*

7.2.1 Overview

In this section, the final results of the simulation experiments are presented. In the next subsection, the calibrated models needed for running the stochastic case are shown. Next, a summary of the simulation results with a pertinent analysis are shown and discussed at length.

7.2.2 Calibration of Stochastic Models

As mentioned in Chapter 5, it has been assumed that the process of vehicle-customer assignment can be described as a discrete (binary) regression model.

In this particular modeling framework, y can be defined as 1 if the dispatching module decides to assign a combination vehicle-segment-customer, 0 otherwise. The right-hand side variables, or independent variables, were chosen as the ones that should better explain the minimizing cost behavior of the dispatching module when taking assignment decisions, given the feasibility constraints of the problem discussed in Chapter 4.

In this experiment, only a logit binary model was considered in the calibration procedure. In most of cases, even though only the observations happening within the catchment area were considered, the number of observations where the insertion is

rejected is considerably higher than those observation of favorable cases (the insertion was accepted). This situation creates some problems in the calibration of a binary probit model, since there is no way to perform the matrix inversion according to the method explained in Appendix to Chapter 5. In the case of the logit model, there is no such problem if the cost (or disutility) associated with the rejected alternatives is big enough for those insertion alternatives to have no real chance of getting accepted (Ben-Akiva and Lerman, 1985; McFadden, 1978).

As stated in Chapter 5, by assuming that the cumulative distribution of the error (μ_i) is the logistic distribution, we obtain the logit model. Analytically

$$P_{AS_j}(ATTR_j(k,k+1), DCT(q)) = \text{Prob}(y_i = 1) \quad = \text{Prob}(\mu_i > -\beta' x_i) \\ = 1 - F(-\beta' x_i) \tag{7.3}$$

where

$$\beta' x_i = \beta_0 + \beta_1 \frac{SP_j(k,k+1)}{E[PS]} + \beta_2 TR_j(k) + \beta_3 WSP_j(k,k+1) + \beta_4 DCT(q) \tag{7.4}$$

In this case, under the logit model assumptions

$$F(-\beta' x_i) = \frac{exp(-\beta' x_i)}{1+\exp(-\beta' x_i)} = \frac{1}{1+\exp(\beta' x_i)} \tag{7.5}$$

hence,

$$P_{AS_j}(ATTR_j(k,k+1), DCT(q)) = 1 - F(-\beta' x_i) = \frac{\exp(\beta' x_i)}{1+\exp(\beta' x_i)} \tag{7.6}$$

The collection of observations is performed under the same scheme as that detailed in Chapter 5. The expected pool size $E[PS]$ is taken directly from the assumed Poisson distribution (equals to 1.5).

In a first attempt, 10 different models like that of expression 7.4 were estimated, one different model per hub area, and under two different conditions of operation, with and without implementing the queuing strategy at hub terminals. The data for the estimation is taken from simulations 17 and 27, both described in Table 7.5. The estimation of the coefficients from a sample size obtained during the most intensive operation interval (say from 30 to 40 minutes of simulation) and the corresponding tests of significance are presented in Tables 7.6 and 7.7.

The description of the statistics and significance tests is the usual.

$-2\left[L(0)-L(b)\right]$ is the statistics used to test the null hypothesis that all the parameters are zero, and it is asymptotically distributed as ^{2} with K degrees of freedom. In this case, K = 5, and the critical value at the 0.01 level of significance is 15.1. Therefore, by observing the value of the statistics, it is possible to reject the null hypothesis in all cases.

In addition $-2\left[L(c)-L(b)\right]$ is used to test the null hypothesis that all the parameters other than the constants are zero. It is asymptotically distributed as ^{2} with K-1 degrees of freedom. In this case, K-1 = 4, and the critical value at the 0.01 level of significance is 13.28. Therefore, again it is possible to reject the null hypothesis in all cases.

Notice that ρ^2 and $\bar{\rho}^2$ are reasonable in all cases.

Finally, an asymptotic t-statistics greater than 2 in absolute value is expected for each coefficient estimate to consider a robust explicative contribution of each specific variable to the model. That happens with almost all variables, except when estimating β_2, which is the parameter associated to the variable $TR_j(k)$, indicating that it is not clear if $TR_j(k)$ is really an explanatory variable in this process. There are some exceptions, particularly when modeling with terminal operation, however in those cases the value of the coefficient itself turns out to be not significant.

With regard to the sign of the coefficients and the importance of each of the variables considered in the model, notice that the follow exactly the expected behavior, validating the structure of the model as a whole. Actually, the only variable that should favor the chance of having an insertion on a specific segment is the available space on the vehicle while entering that segment (positive sign). All other variables have negative contributions, because they are introduced to incorporate the penalties of such an insertion for a routing decision. The exception is model 1 with terminal operation for the case of $TR_j(k)$, but the coefficient in that cases is not significant at all.

Table 7.6 Estimation of a binary logit choice model (with terminal operation)

Model Hub 1

	Coefficient	*std_error*	*t-statis.*
β_0	-4.2453	0.4174	-10.1702
β_1	0.4994	0.0923	5.4123
β_2	0.0001	0.0001	0.8245
β_3	-0.1141	0.0944	-1.2082
β_4	-0.0012	0.0003	-4.6541

Summary statistics

# of cases	5721
$L(0)$	-3965.5
$L(c)$	-686.963
$L(b)$	-653.972
$-2[L(0)-L(b)]$	6623.045
$-2[L(c)-L(b)]$	65.9804
ρ^2	0.8351
$\bar{\rho}^2$	0.8338

Model Hub 2

	coefficient	*std_error*	*t-statis*
β_0	-4.0924	0.5678	-7.207
β_1	1.0619	0.1542	6.8858
β_2	-0.0006	0.0002	-3.5393
β_3	-0.3197	0.1006	-3.1782
β_4	-0.0015	0.0004	-3.377

Summary statistics

# of cases	3038
$L(0)$	-2105.78
$L(c)$	-498.974
$L(b)$	-435.368
$-2[L(0)-L(b)]$	3340.826
$-2[L(c)-L(b)]$	127.2117
ρ^2	0.7933
$\bar{\rho}^2$	0.7909

Model Hub 3

	Coefficient	*std_error*	*t-statis*
β_0	-2.2801	0.5494	-4.1505
β_1	0.6825	0.1374	4.9672
β_2	-0.0007	0.0001	-4.574
β_3	-0.5017	0.1035	-4.8465
β_4	-0.0005	0.0004	-1.1193

Summary statistics

# of cases	2903
$L(0)$	-2012.21
$L(c)$	-480.784
$L(b)$	-434.684
$-2[L(0)-L(b)]$	3155.045
$-2[L(c)-L(b)]$	92.2007
ρ^2	0.784
$\bar{\rho}^2$	0.7815

Model Hub 4

	coefficient	*std_error*	*t-statis*
β_0	-3.6352	0.4517	-8.0482
β_1	1.1147	0.1158	9.626
β_2	-0.0005	0.0001	-4.3936
β_3	-0.5106	0.0785	-6.5071
β_4	-0.0015	0.0003	-4.7734

Summary statistics

# of cases	6169
$L(0)$	-4276.03
$L(c)$	-946.14
$L(b)$	-819.21
$-2[L(0)-L(b)]$	6913.627
$-2[L(c)-L(b)]$	253.8496
ρ^2	0.8084
$\bar{\rho}^2$	0.8072

Table 7.6 (contd) Estimation of a binary logit choice model (with terminal operation)

Model Hub 5			
	coefficient	*std_error*	*t-statis.*
β_0	-5.1955	0.311	-16.7059
β_1	1.1423	0.0725	15.7646
β_2	-0.0001	0.0001	-1.5321
β_3	-0.399	0.0577	-6.9096
β_4	-0.0012	0.0002	-7.0443

Summary statistics	
# of cases	20368
$L(0)$	-14118
$L(c)$	-2175.75
$L(b)$	-1939.22
$-2[L(0)-L(b)]$	24357.61
$-2[L(c)-L(b)]$	473.0602
ρ^2	0.8606
$\bar{\rho}^2$	0.8623

Table 7.7 Estimation of a binary logit choice model (without terminal operation)

Model Hub 1

	coefficient	*std_error*	*t-statis.*
β_0	-3.4707	0.381	-9.1087
β_1	0.5059	0.0901	5.6162
β_2	0	0.0001	-0.3748
β_3	-0.4035	0.0819	-4.9237
β_4	-0.0009	0.0003	-3.628

Summary statistics

# of cases	5966
$L(0)$	-4135.32
$L(c)$	-707.88
$L(b)$	-666.681
$-2[L(0)-L(b)]$	6937.271
$-2[L(c)-L(b)]$	82.3934
ρ^2	0.8388
$\bar{\rho}^2$	0.8376

Model Hub 2

	coefficient	*std_error*	*t-statis*
β_0	-3.9092	0.3115	-12.5507
β_1	0.6078	0.0731	8.3171
β_2	-0.0001	0.0001	-1.1711
β_3	-0.2363	0.0583	-4.0561
β_4	-0.0011	0.0002	-5.3513

Summary statistics

# of cases	9238
$L(0)$	-6403.29
$L(c)$	-1223.33
$L(b)$	-1138.59
$-2[L(0)-L(b)]$	10529.41
$-2[L(c)-L(b)]$	169.4926
ρ^2	0.8222
$\bar{\rho}^2$	0.5214

Model Hub 3

	coefficient	*std_error*	*t-statis*
β_0	-3.726	0.275	-13.5507
β_1	0.6419	0.0639	10.0519
β_2	-0.0001	0.0001	-1.0776
β_3	-0.3033	0.052	-5.8283
β_4	-0.0013	0.0002	-6.7689

Summary statistics

# of cases	12528
$L(0)$	-8683.75
$L(c)$	-1723.23
$L(b)$	-1597.44
$-2[L(0)-L(b)]$	14712.61
$-2[L(c)-L(b)]$	251.567
ρ^2	0.816
$\bar{\rho}^2$	0.8155

Model Hub 4

	coefficient	*std_error*	*t-statis*
β_0	-4.1048	0.2581	-15.9036
β_1	0.7605	0.0597	12.7322
β_2	-0.0001	0.0001	-1.0371
β_3	-0.2819	0.048	-5.8714
β_4	-0.0014	0.0002	-8.0402

Summary statistics

# of cases	15746
$L(0)$	-10914.3
$L(c)$	-2179.12
$L(b)$	-1994.6
$-2[L(0)-L(b)]$	17839.4
$-2[L(c)-L(b)]$	369.0589
ρ^2	0.8172
$\bar{\rho}^2$	0.8168

Table 7.7 (contd) Estimation of a binary logit choice model (without terminal operation)

	Model Hub 5		
	coefficients	*std_error*	*t-statis.*
β_0	-4.632	0.2252	-20.5657
β_1	0.8648	0.0514	16.8105
β_2	0	0	-0.0894
β_3	-0.3299	0.0427	-7.7188
β_4	-0.0013	0.0001	-9.3966

Summary statistics	
# of cases	29850
$L(0)$	-20690.4
$L(c)$	-3407.81
$L(b)$	-3125.58
$-2\left[L(0)-L(b)\right]$	35129.72
$-2\left[L(c)-L(b)\right]$	564.4543
ρ^2	0.8489
$\bar{\rho}^2$	0.8487

In view of all previous comments, the estimation was repeated but now removing $TR_j(k)$ from the model, obtaining a functional form like this:

$$\beta' x_i = \beta_0 + \beta_1 \frac{SP_j(k,k+1)}{E[PS]} + \beta_2 WSP_j(k,k+1) + \beta_3 DCT(q) \qquad (7.7)$$

In this case, the obtained models are similar than those obtained above, and in all cases the previous conditions and comments are fulfilled.

7.2.3 Summary and analysis of simulation results

In this section, a summary of the results of all simulations described above is presented. First at all, a general description of the performance of each run is shown in Figures 7.11 to 7.17, in terms of waiting time at both, home and hub, total ride time, for both internal and external trips, and average load considering the reroutable and non-reroutable portion of the trips. Let us first take a look to the averages waiting times at home and hub.

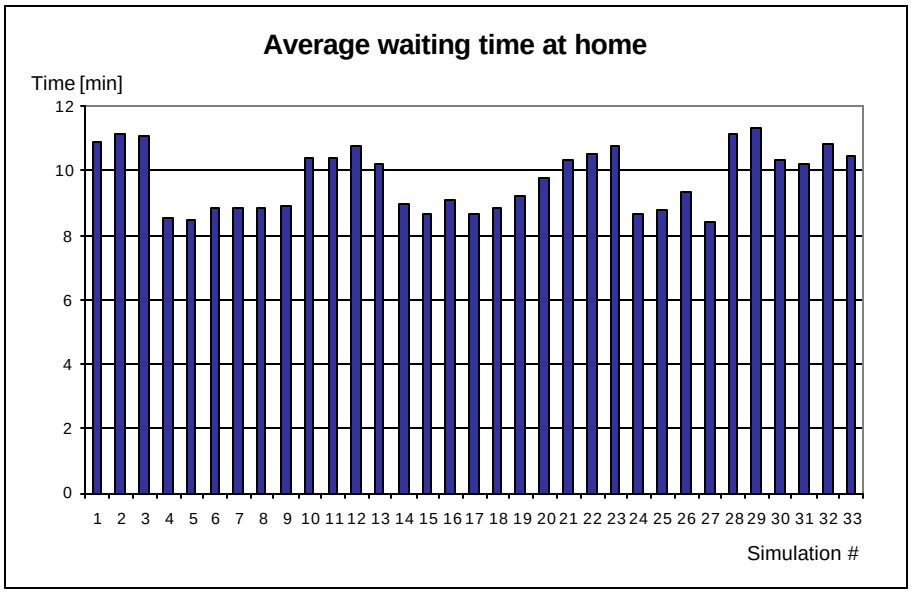

Figure 7.11 Average waiting time at home

For a real-time response system like *HCPPT*, average results in terms of waiting times seem quite reasonable, with values around 10 minutes at home, for a system where calls are entering without previous notice, and similar values of waiting at terminals.

The stochastic runs (30 to 33) do not seem to yield much better results in terms of waiting times. That fact is possibly due to the limitations in the proper modeling of the stochastic component of the waiting time (see Section 4.3.2.3), and also due to a lack

of calibration of the parameters involved in the waiting time portion of the cost formulation.

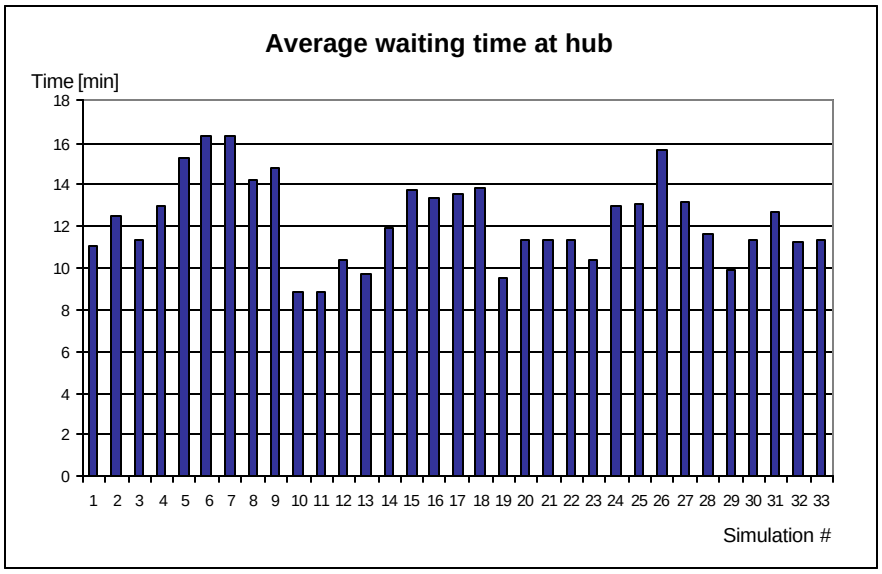

Figure 7.12 Average waiting time at hub

All simulations that incorporate the queuing strategy developed in Section 4.6.1.1 (for example, simulations 23 to 29, 31 and 32 in the stochastic case) in general present an improvement in waiting times, in some cases significant both at home and at the hub.

It is always an improvement to incorporate vehicles in state 2 as part of the candidate set in terms of waiting time. When vehicles are removed, waiting times increase, but not very much. This comment is important, particularly when analyzing the vehicle load next.

Let us now analyze the ride times associated with all the final simulations.

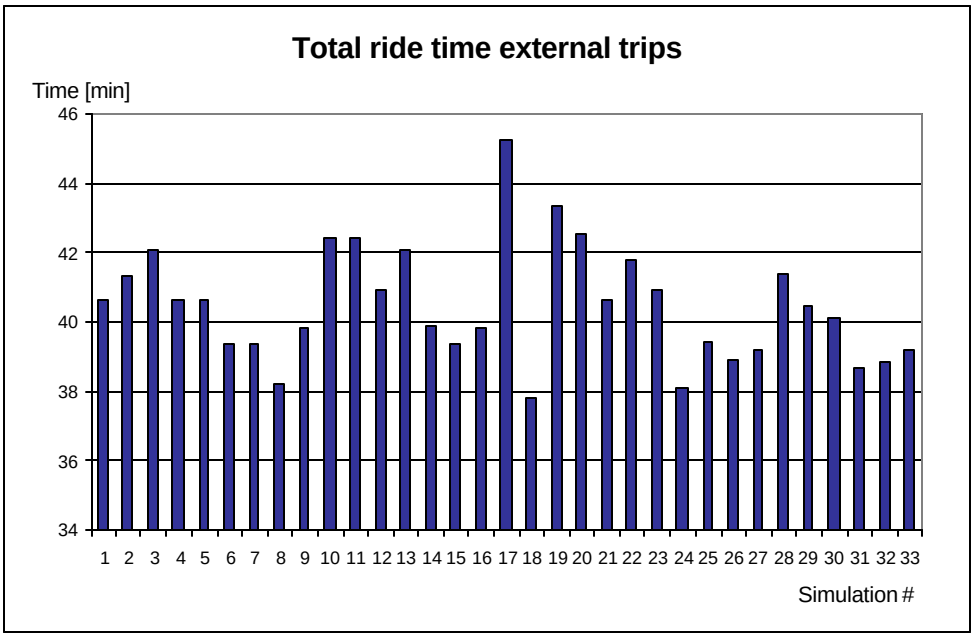

Figure 7.13 Total ride time for external trips

In this case, stochastic simulations perform very well in terms of ride times when compared with deterministic runs under the same conditions.

In addition, notice that incorporating queuing strategies always improves the solution in terms of ride times. From figure 7.14, it is possible to conclude that the operation complements the service of internal trips quite well, reaching very reasonable ride times. The impact of queuing strategies is clear when we compare simulations 17 (without queuing) and 18 (with queuing) under the same conditions.

Another important strategy is the repositioning of vehicles when no other task is assigned to them (Section 4.6.2.1). These simulations always show better system performance when such repositioning is attempted. The use of a proper rule for deciding when the vehicles must enter the trunk network (Section 4.6.2.2) also improves the performance and was thus included in most simulations.

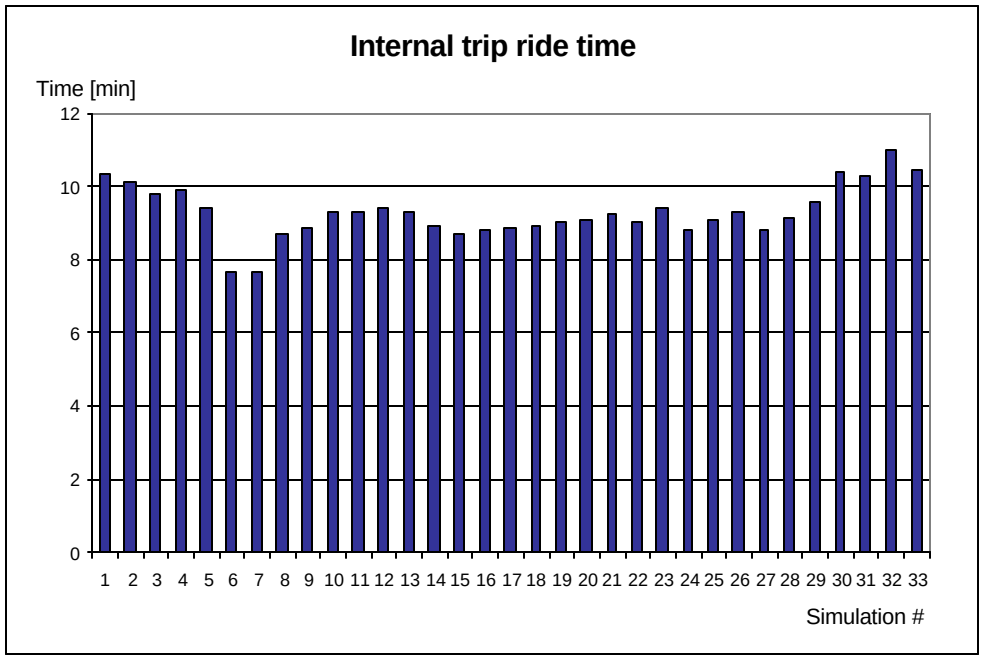

Figure 7.14 Total ride time for internal trips

As in Chapter 3, the system performance measures discussed by Black (1995) are adopted in order to analyze the system design efficiency and performance under different simulation conditions and different demand levels as well. Let us first define the *level-of-service index* at hub level $_i$ as follows

$$_i = \frac{\text{Average passenger waiting time at pick-up location}}{\text{Average door-to-door ride time}} \qquad (7.8)$$

where the denominator ride time is calculated as the average *door-to-door* ride time from hub i to all the adjacent hub destinations. Another performance index is introduced here, the *ride time index* ρ defined at system level. That is

$$\rho = \frac{\text{Average vehicle ride time}}{\text{Average door - to - door ride time}} \qquad (7.9)$$

The *door-to-door* ride time is the time of travel when no other passenger is picked up. The average *door-to-door ride time* is computed from particular vehicles in PARAMICS in a separate calculation. The average between hubs is an average of the value obtained between cells. For computing the value assigned to each pair of cells, the corresponding PARAMICS zones belonging to each cell had to be identified.

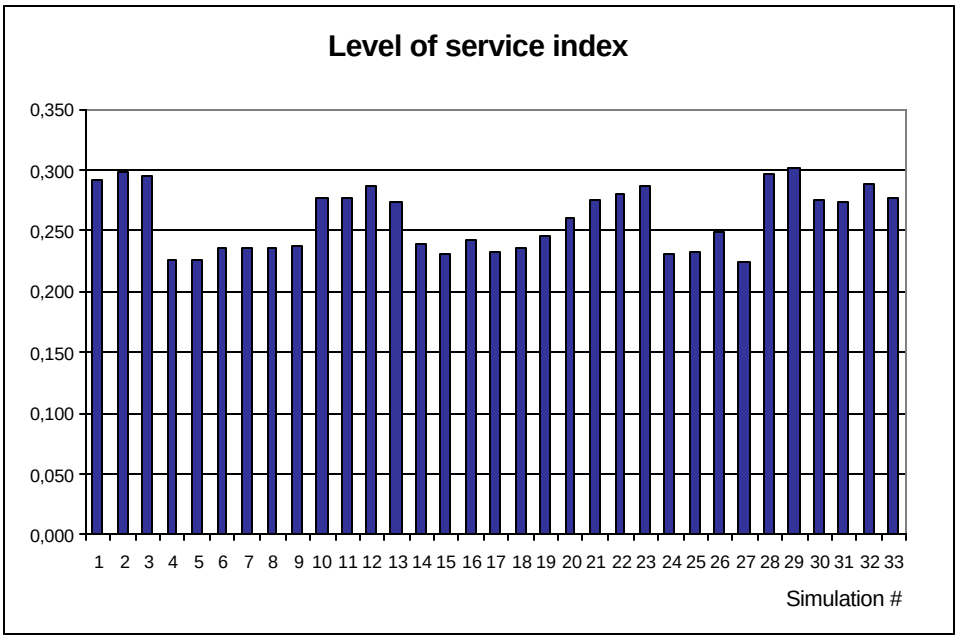

Figure 7.15 Level of service index

Waiting times up to 11 minutes at pick-up are very reasonable for this system. It is comparable to taxi services and possibly somewhat better, and it is much better than the reported values for the *dial-a-ride* services of the past. It is also quite better than the walking and waiting times in fixed route transit services. In addition, waiting time can be

rescheduled to other activities if the pick-up is at home, unlike in the traditional transit services. This is a significant factor to consider in the "real" cost the passengers perceive towards the waiting times.

The (*level of service index*) values represent the importance of the waiting time at home compared with the ride time itself. Previous *DRT* systems have shown values up to even 2 or 3 (i.e., waiting times were two to three times travel times) which were rather unacceptable. The simulations have consistently found values between 0.2 and 0.3. = 0.26 means that for a 20 minute trip, the user must wait around 5 minutes until service. The transfer waiting times at hubs are low enough; no more than 15 minutes in the worse case, which shows that a high coverage system can indeed provide acceptable transfer options. However, there is a tradeoff between these two indices, reflected through the adopted routing strategy. Different simulations can yield results that decrease the home waiting or hub transfer time at the expense of each other.

The *ride time index*, on the other hand, shows the rate between *door-to-door* service ride time and the resulting ride time provided by our system. So, $\rho = 1.10$ means that the ride time provided is 10 % more than that provided by a competitive *door-to-door* service. On the other hand, for higher ρ values does not exceed 1.206 (see Figure 7.16), which is in any case, better than the values reported in previous *DRT* experiments, and better than the values reported in Chapter 3 for the spatial simulation.

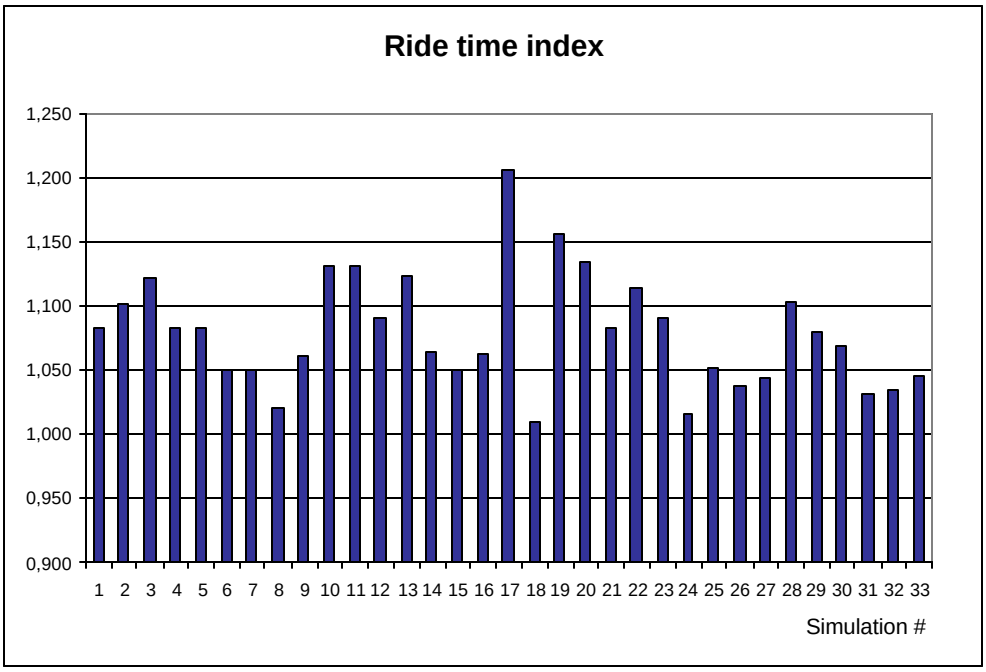

Figure 7.16 Ride time index

Finally, the productivity of the system operating under these conditions is measured by the average load (in passengers/vehicle), split into reroutable and non-reroutable portions. A summary of that measure for the whole system is shown in Figure 7.17.

As expected, in all simulations the non-reroutable portion shows greater load than the reroutable portion load, showing the advantage of implementing a system like *HCPPT*, where vehicles can feed other vehicles at transfer points, increasing the overall productivity of the entire system.

The highest average load (3.13 pax/veh) is obtained for a simulation where 75 vehicles were removed from the total fleet. In such a case, the total ride time did not change considerably compared with others that could be considered better (42.44 minutes versus the minimum ride time which is 37.82 minutes).

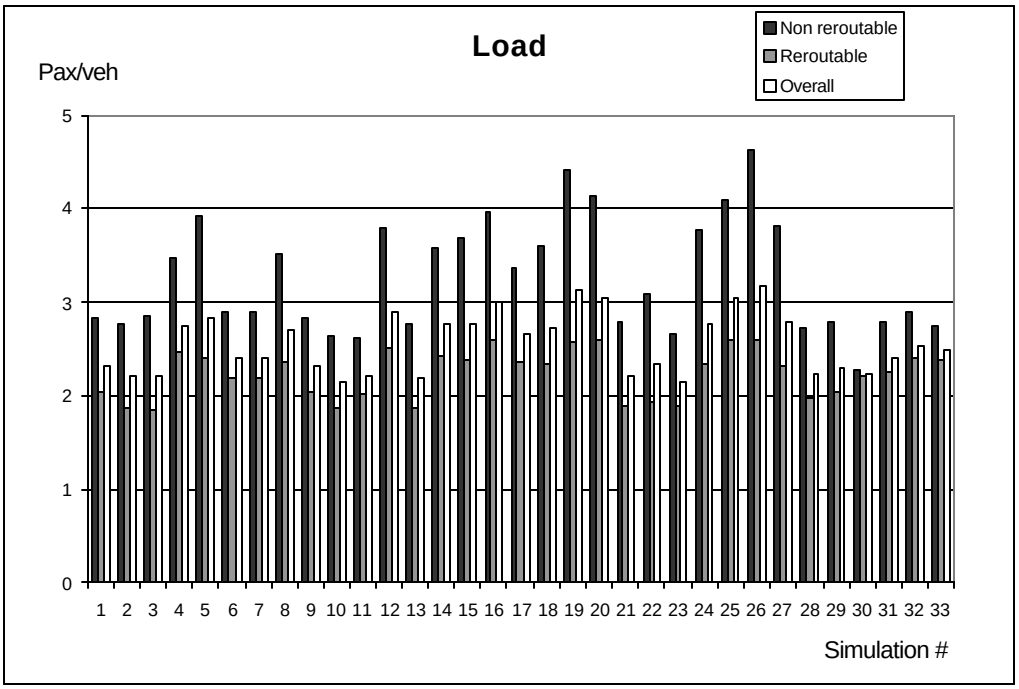

Figure 7.17 Productivity of the system (Average load)

Therefore, this implies that the fleet size aspects should be explored further along with additional routing strategies. Stochastic runs are not particularly good in terms of productivity, since the cost function does not explicitly consider this effect in the stochastic formulation.

7.3 Final remarks

In this chapter, a summary of all experiments performed in the context of this dissertation is presented.

This application over a real network coded at a microscopic level was very successful, in both methodological and numerical terms. Further research is needed in order to adjust and calibrate some of the parameters for improving the obtained solutions. In addition, a sensitivity analysis from the demand standpoint has to be carried

out, in order to study the sensibility of the design itself under different demand conditions. Such experiments are very useful in order to decide the critical conditions needed for the proposed system to work properly. In terms of running times of the algorithm, a summary of the required computation, split into the pick up decision part and the whole decision every time a new group of calls has to be routed, is presented in Figure 7.18.

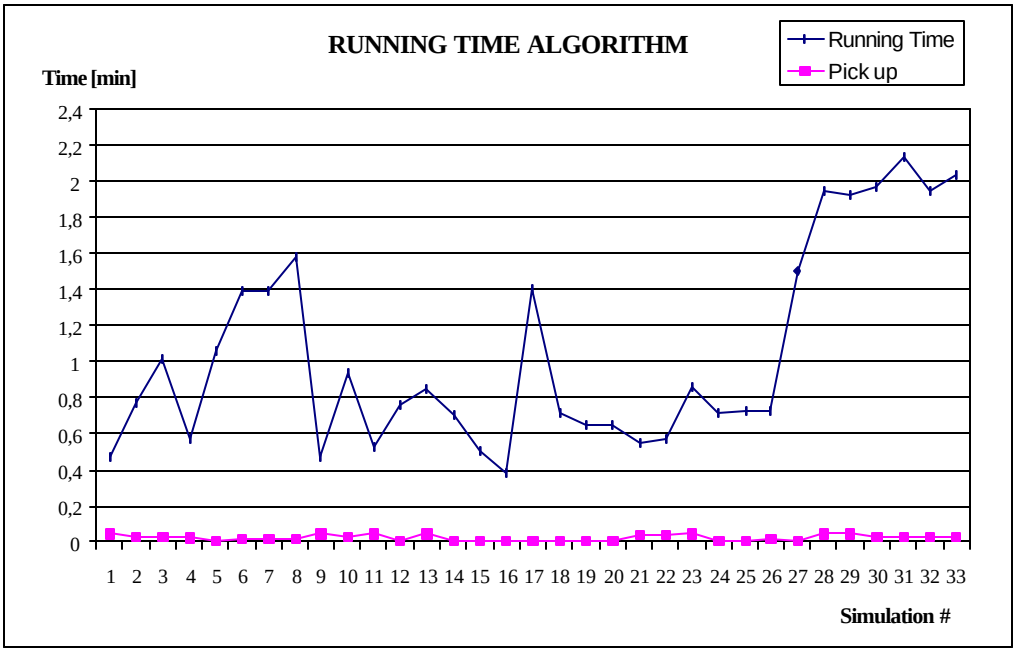

Figure 7.18 Algorithm running times summary

As shown in the figure, running times are negligible, due to the simplicity of the insertion algorithms as well as the additional simplifications and practical considerations added to the simulations. This fact is very important when we consider running this kind of strategies in real-time as required by the scheme to work. As expected, stochastic algorithms are more time consuming than the deterministic ones.

In the final chapter, a brief economic analysis is preformed in view of the results discussed here.

8 SUMMARY AND CONCLUSIONS

8.1 Summary of the Dissertation

This dissertation research proposes the development and evaluation of a new concept for high-coverage point-to-point transit systems (*HCPPT)*. The system is a radically new operational scheme for mass transit that relies on a large number of small transit vehicles routed using advanced communication technology and real-time stochastic control algorithms.

Overall, three major contributions can be identified as the core of this research: the proposed scheme design, the development of sophisticated routing rules that can be updated in real-time to implement and optimize the operation of such a design, and the implementation of a multi-purpose simulation platform in order to simulate and evaluate such a design under real network conditions.

In Chapter 3, the details of the concept were described, outlining its features and the flexibility of potential operational schemes. The design is based on Jitney or Shuttle-style operations with a large number of deployed vehicles under a coordinated transit system that uses advanced information supply schemes with fast routing and optimization schemes. The system design is rather innovative in the vehicle operation scheme and ensures that no more than one transfer is needed for the travelers, by using transfer hubs as well as reroutable and non-reroutable portions in the vehicles' travel plans. It yields flexibility for demand-side benefits from options such as price incentives

for time-bound "passenger-pooling" at the stops without destination constraints to the users.

In Chapter 4 and 5, the real-time routing algorithms are presented. This is a system particularly suited for fast real-time routing schemes based on optimized vehicle selection algorithms and decomposed local solutions for the "pick-up-and-delivery" logistics. A strict optimization formulation and solution for such a problem is computationally prohibitive in real-time. The design proposed in this dissertation is effectively geared towards a decomposed solution using detailed rules for achieving vehicle selection and route planning. If real-time update of probabilities based upon modeling the future dispatch decisions is included, then this scheme can be considered as a form of quasi-optimal predictive-adaptive control problem.

In Chapter 6, a multi-purpose simulation platform developed for this research is presented. This chapter discussed the need for developing more comprehensive urban transportation network simulation environments that go beyond the conventional microscopic simulation models that have largely been auto-centric. A candidate framework to develop such simulations and presented the application of such a framework for certain simulation contexts. In particular, the simulation framework has been applied to the *HCPPT* study, allowing the modeler to simulate and evaluate the proposed system under various supply-demand conditions. There were two motivating factors for such microscopic simulations. One, the need to study the "devil in the details" in a system that has a lot of details and parameters, with the performance being intricately dependent on real-time routing rules which in turn depended heavily on network dynamics, making options such as Monte Carlo travel time simulations useless.

The second motivation is the recent interest among practitioners towards the use of micro-simulation and the renewed interest among researchers in considering such simulations as a potentially viable option in the future, especially with commercial software vendors entering the market place in a bigger way than in the past. A proper simulation of the system appeared to be a necessity to convince those who may not easily be convinced by the theoretical arguments presented at the outset of this dissertation.

However, the final simulations of *HCPPT* required point-to-point vehicle simulation, which is not possible with off-the-shelf simulators. The simulation framework did utilize a well-known microscopic traffic simulator, though it was significantly modified for demand-responsive vehicle movements and passenger tracking, this eventually leading to the development of a multiple vehicle-class general purpose microscopic simulator that could model a variety of fleet systems.

In Chapter 7, the implementation of the operational scheme was evaluated via detailed simulations, carefully examining the routing rules and strategies developed. Illustrative results from a simulated case study in Orange County are then shown. The final application required a highly detailed network treatment, the consideration and study of supply issues, terminal operations and in general the implementation of all relevant simulation details. After conducting several runs, under different routing and operational strategies, simulation studies showed that with enough deployed vehicles, the system can be substantially better, even competitive with personal auto travel, compared to the often-unsuccessful traditional *DRT* systems and the existing fixed route

public transit. Furthermore, *HCPPT* can be incrementally implemented by contracting out services to existing private operators.

8.2 Conclusions and Policy Issues

This dissertation originated from a simple idea for design and operation of a transit system that could be somewhat competitive with the automobile. In a first attempt, the system was simulated and evaluated under very simplistic conditions (see application and spatial simulation in Chapter 3), obtaining more promising results in terms of the level of service and productivity than in previous *DRT* systems as reported in the literature.

After this first experiment, the design appeared to be sufficiently attractive and interesting for it to be studied more carefully. The next task was then to develop a much more elaborate design along with sophisticated routing rules. This led to significant research that started with insertion based heuristics for vehicle routing and culminated in the development of a real-time stochastic-predictive control scheme for a transit system. (as in Chapters 4 and 5). In fact, the approach is rather different than in conventional transit research, in that it views the *HCPPT* system to be effectively a real-time feedback control scheme applied to transit with notions of optimality and distributed solution schemes.

Finally a case study was undertaken with a large Orange County network with the simulations yielding, under several conditions, even better results than those obtained in Chapter 3 for the spatial simulation application. Results were not economically evaluated in the context of this study, however, there are some very important insights to

be considered in further economic evaluations and implementations of a system of this characteristics.

Next, some of the most relevant issues from the whole process are addressed:

- One of the major characteristics of the scheme is the "high-coverage" feature, in the sense of having a great number of available vehicles everywhere at any time, which make it distinguishable from other *DRT* designs. This particular issue makes the system highly responsive to service requests in real-time. That is also one of the reason of why the system was studied for serving dynamic demand generated in real-time with no previous knowledge of both the location and time of generation, by the dispatching algorithm.

  In Figure 7.11 (Chapter 7) it is clear that the system is responding to the service request for trips to adjacent hub areas quite competently. That is, the customer has to wait for around 10 minutes for service for a trip of 40 minutes ride time, which is quite competitive with any other mode with high level of service, even the personal automobiles. For trips within the bug area, however, the waiting time to ride time ratio (*level of service* index as defined before) does not seem to be very attractive, since the internal trip ride time is quite low (about 10 minutes too). However, the system was originally designed to serve external trips more than internal trips, and for them it performs quite well. It is not useful to keep reducing the average (and individual) waiting time at home from the values reported in the Chapter 7 experiments by adding more vehicles, because the vehicles would then be idle for much more time, thus sacrificing the good productivity levels obtained in most

simulation . This is one of the reasons of why internal trips traveling alone (without pooling with at least one external passenger) were removed from the demand data set when preparing the simulation experiment. It must be noticed anyway that the internal travel times are very comparative with an alternative straight door-to-door service. This responsive feature could have a very dramatic impact on the potential demand for such a service, considering that the waiting time at hubs is also very reasonable (in some cases close to 10 minutes too).

In addition, the discomfort associated with all situations where the customers has to wait, whether it is at home or inside a terminal, could be diminished by adding information systems that greatly reduce the user uncertainty. At home, for example, the company could implement an Internet web page with information in real-time about the exact location and the expected time of arrival of the vehicle assigned to every specific customer waiting for being picked-up. In case of terminals, it is important to keep displays with vehicle locations and expected arrival times by destination everywhere. In addition, all this information should be clear and simple to be understood by any passenger. Terminals should also be equipped with coffee shops, comfortable seats, a closed and protected waiting space in case of bad whether, etc.

- An important comment that could arise from the results shown in Chapter 7, is the importance of the specific strategy to be applied at different stages of the scheme, regarding both vehicle availability for inserting a new pick-up and passenger management at hub level. It is important to implement strategies that can keep the

terminal queues almost empty, as in most cases in which vehicles in state 2 were allowed to pick-up new calls as a strategy.

The queuing strategy at terminals is also a critical factor in terms of reducing both ride time and waiting time at hubs. In this dissertation, a very simple scheme was implemented in order to manage passenger assignment according to spatial and temporal considerations. More work is possible in improving the strategy and also in calibrating some parameters in the formulation of the heuristic rule. The schemes provided here for queue management are however a very good starting point, the results showing considerable improvement in efficiency and level of service. It must be noticed that the implementation of such a scheme requires a very important investment in technology, since the passenger-vehicle assignment at any particular hub is decided based on information (location, occupancy, expected time of arrival, etc.) regarding all vehicles coming from any other adjacent hub. In addition, the decision of leaving the trunk network by those vehicles could change in real-time depending on the number and features of passengers arriving at hub over time.

Other strategies, such as repositioning of vehicles in real time when no other task is assigned to them, and also the decision on either sending vehicles to the trunk network or keeping them within the reroutable area, are not simple and depend on several factors, particularly the use and access to information technology. These strategies showed some impact; however, the real contribution of each heuristic by itself is hard to measure in the relatively limited set of simulations here.

- The adaptive-predictive control scheme developed in Chapter 5 and implemented in the last reported simulations (see Table 7.5) showed a contribution in capturing correctly the travel time between stops. That is why there is an improvement in external ride times when considering the stochastic formulation under the same conditions of the deterministic case. However, there is no perceivable improvements in productivity and waiting times due to stochastic control schemes. In case of productivity, it is not clear how the introduction of stochastic terms in the incremental cost formulation impacts occupancy, the primary indicator of productivity as defined here. On the other hand, in case of waiting time, it is necessary to make a sensitivity analysis with respect to the parameters associated with the waiting time components in the formulation. The only way to accomplish that is by constructing a scenario isolating the reroutable part of the system, probably with less vehicles running, in order to capture the real impact of each component of the unknown cost function in the formulation of Chapter 5. A possible approach to deal with such a problem is roughly presented in Section 8.3.2 next.
- With regard to the economic impact of the implementation of the system, notice that a system like this should be actually able to attract potential demand from the automobile. A careful and involved study must me performed before making a stronger statement on this. However, if a reasonable demand level can feed the system, it is possible to obtain reasonable productivity levels at least in view of the simulation results shown in Chapter 7. With occupancy levels higher than 3 pax/veh, along with very reasonable waiting and ride times, it is possible to offer very reasonable fare levels to the customer, much less than what a user would pay by

taking a taxi for the same trip, but with possibly even better service than in taxis in some cases, if we disregard any disutility from having to share the vehicle with other passengers.

Let us suppose that the operational cost of *HCPPT* is about the same as that of a traditional *DRT* system, for an average trip of about 10 miles (that is what the proposed scheme is covering between adjacent hub zones). The actual expense in current DRT systems are up to nearly $20 per vehicle trip (Dessouky, 2002). For the same trip, a taxi service would charge about $ 20 per passenger.

Nowadays, a fixed-route transit system shows operational costs of about $ 4.0 per passenger. State subsidies up to 75% or more are normal in the fixed route US urban transit which change $ 1.0 or less per passenger trip. With the productivity levels reached by HCPPT, one could charge a fare of about $20/3 \approx \$7.0$ per passenger, using the costs mentioned above for DRT type services. Another simple argument why the system could be operated with such fares is that taxi systems are already able to break-even with fares up to $1.5 or $2 per passenger mile. With nearly 3 times as many passengers due to higher occupancy and with lesser idle time for the driver, it is not impossible to bring the costs down to 0.75 or lesser per mile per passenger. With any reasonable level of subsidy (say even 30 %), a fare of $ 5.0 per a trip of 10 miles is quite possible, the crux of the argument here primarily being based on occupancy. The system will very attractive when compared with even a personal auto option but without the parking related worries. In addition, level of service indices are also very competitive, showing that this scheme could attract enough demand to make it work. Traditional DRT systems are not comparable, since

they reach low occupancies (about 1.39 on average as shown by Black, 1995) as well as poor level of service and ride time indices, resulting in low demand levels and very few vehicles running on the service network at any time. All these preliminary numbers make the idea very interesting and candidate to be deeper studied, particularly regarding demand-side potential as well as acceptability by operators and contractors. These issues are analyzed in the next section as future lines of research.

On the other hand the very simplistic cost analysis above is certainly not sufficient to justify the system. Myriad other issues need to be considered. The most important would be issues such as union rules of the drivers, liability matters, and ADA requirements. Despite all that, there are clearly several arguments in favor of implementing such systems incrementally using existing taxi operators to operate *HCPPT* services for a public transit agency. For various reasons, the taxi companies may at first object to public transit agencies encroaching into their territory, but it is also possible that they would view it as an opportunity to derive significantly higher utilization of drivers and vehicles. Being even part-time partners in a public/private scheme with a coordinated structure and with more public credibility may be viewed as advantageous by existing taxi and shuttle operators. The public transit agency can avoid huge capital costs in implementing the system by contracting out the operations as well. Once the higher demand for such systems start materializing it can be expanded further.

The point remains that the possibilities are rather endless in HCPPT thanks to its flexibility. Much further research is needed before it can be implemented in practice, however.

8.3 Further Research

In this section, the most important issues to be explored in the future as a continuation of this research are outlined. Some of them have been thought and developed deeper than others in the context of this dissertation. Regardless of the analysis devoted at any case, in this section at least the potential lines of research needed for each topic are addressed .

In the next subsection, one of the most relevant issues regarding the success of the implementation of the *HCPPT* scheme and out of the scope of this dissertation is discussed: The treatment and modeling of the potential demand for this system. Next, other issues related to optimality of the routing rules under a control theory scheme are briefly discusses, finalizing with some simulation issues not completely developed in this dissertation.

8.3.1 Treatment of the demand

8.3.1.1 Relevance

The proposed scheme has potential to significantly add to the state-of-the-art in transit systems concepts. The model, if it can be proved feasible from the demand and institutional stand point, could then be incrementally implemented in many urban contexts by contracting existing private transport operators to be part of a coordinated

transit system. Improved efficiency of the transportation system and significant improvements in mobility and accessibility is a direct result if the idea is practicable. Therefore, it is clear that a very attractive line of research could yield the demand-side insights that would drive the development of such systems or even their demise.

Performance and Supply side studies as part of this research have demonstrated that many of the initial intuitions on how to handle the problems with *DRT* schemes of the past using modern information technology have been proven correct so far in simulations. The demand study is what needs to come next.

8.3.1.2 Methodology

An initial proposal focuses carefully on the demand and institutional sides, to support or disprove our claims on the feasibility of *HCPPT*. The focus in this proposed research is on three aspects – (1) the acceptability of the system from the transit agency side, as well as the private operator agency side if it is implemented with contracted-out service; (2) the acceptability from the operators, primarily drivers and other transit unions workers; and (3) the detailed demand-side responses towards the concept from potential travelers. As the system does not exist yet, Delphi techniques, stated preference (SP) surveys, and finally SP study of revealed behavior of people (passengers) making decisions in a laboratory simulator environment are proposed. The Delphi survey is for qualitatively measuring the feasibility of the *HCPPT* system taking into account the opinions of a variety of experts taken from Federal Agencies, Transit Operators, Driver Syndicates, etc. The design of the questionnaires will follow the general rules described in the literature, but oriented to an exploratory and qualitative survey, providing a useful set of

insights into practical feasibility. With respect to the design of a potential user survey experiment, a brief background regarding SP and RP (revealed preference) studies is presented later in this Chapter. The decision behavior of the users in a laboratory simulation is technically not RP which should be in the real-world decision context. Several of the econometric model building aspects are however identical to RP, in the case of laboratory discrete choice experiments.

The Delphi method

One of the key issues in the kind of system proposed here ever becoming a reality, is the institutional inertia against change in transit paradigms. No models exist that are directly applicable in finding to what extent a new system such as *HCPPT* is possible. The system can be implemented in incremental manner when it would be augmenting existing systems in marginal ways and slowly developing towards more comprehensive systems. What institutional issues are involved in this, needs to be studied, and a Delphi method of asking selected participants in repeated interviews, would yield insights into several of such factors. This study is really a Delphi survey study, however, the techniques involved are quite varied. For extensive reviews of techniques for Delphi please see Adler *et al*. (1996), Helmer (1977), and Brockoff (1975).

Stated and Revealed Preferences Methods

Given a set of beliefs about the attributes possessed by product alternatives, consumers develop a preference ordering for products, and depending upon budget (or other constraints) make decisions about whether to purchase similar behavior is often assumed

for travel decisions, as is well-known. This process framework gives an explanation of the choice behavior in a physically observable way, where psychological variables are also included in building and calibrating utility (preference) functions.

Depending on the analyst's objectives, explanatory variables at one level can serve as instruments for measures at other levels of the consumer's behavior, providing an opportunity to reduce specification errors and improve efficiency. The popularity of SP methods is now seen in the many travel surveys, which incorporate some from preference or choice experiment, sometimes to complement revealed preference (RP) data and sometimes as the only empirical means of studying a particular behavioral response.

A stated preference analysis follows a controlled experiment, with a series of survey questions eliciting a response to alternative combinations of levels of attributes. A good experiment is one that has a sufficiently rich set of attributes and choice contexts, together with enough variation in the attribute levels necessary to produce meaningful behavioral responses in the context of the strategies of study. This study should use well-developed survey techniques (for details, see J.J. Louviere, D.A. Hensher and J.D. Swait, 2000, which is an excellent and updated review of all SP/RP techniques).

Supplier response: Exploratory survey experiment

In this dissertation an elaborate simulation model was elaborated. The Orange County network was used as the basis, with detailed microscopic simulations of vehicle movements, dwell times, passenger generation, and congestion-effects. This yields the

kind of empirical study tool that agencies can use for studying the effectiveness of a demand-responsive transit system like *HCPPT*. However the real contact with agencies, operators, drivers, etc., through a qualitative Delphi type-survey experiment will provide a better understanding of the necessary conditions required to implement such a scheme.

The Delphi method presented above makes use of a panel of experts, selected based on the areas of expertise required. In this case, agency representatives, public and private transit operators, and driver representatives could be chosen. The notion is that well-informed individuals, calling on their insights and experience, are better equipped to predict the future than theoretical approaches or extrapolation of trends. Their responses to a series of questionnaires are anonymous, and they are provided with a summary of opinions before answering the next questionnaire. It is believed that the group will converge toward the "best" response through this consensus process.

Detailed simulations can show the kind of social and economic factors to be considered in the design of the questionnaires. Under possible public-private cooperation, the existence of subsidies in the operation of the system and the possible institutional support are issues to be explored in the development of these Delphi panel experiment designs. All these factors are fundamental in order to compare the operation of such a system against a more typical *DRT* scheme, which are usually low-coverage systems for specific purposes (service for the elderly, handicapped, etc.).

Private provision of public transport is a topic of much recent discussion. As a matter of fact, several case studies of the role of politics as against economics are currently available, showing how transit services get supplied (J. Richmond, 2001). *HCPPT* is a scheme that may developed with private jitney, shuttle or even a taxi

companies, under proper regulations, fare structure and operational rules governed by a transit public agency. A qualitative measurement of the public acceptability regarding all these preliminary ideas should be explored in the context of the aforementioned Delphi study.

User response: Exploratory survey and development of behavioral models

The evaluation of the efficacy of such a system requires careful simulation of passenger demand, to find out if the required occupancy (perhaps at least around passengers per vehicle) can be achieved under service with enough vehicle coverage. Conventional demand models such as common disaggregate models do not capture the passenger demand-behavior well enough, as they arguably are not sensitive enough towards travelers' waiting time disutility. The reason for this is that such models have not been calibrated with data sets that include the time dynamic dimension finely enough (no system of this kind we describe exists out there). In this context, it becomes useful to conduct laboratory simulated experiments on traveler decisions, using the full-screen graphical interfaces and displays of transit operations to hypothetical travelers who "experience" travel. The simulation capabilities already developed here are ideal for creating decision "scenarios" for travelers.

In this study, the potential domain of demand for this system need to be identified, along with the most significant factors and their specific weights in the user mode decisions (that is, which aspects of the design are most ridership-sensitive) as well as the impact on modal split after its implementation. Note that the proposed scheme is

intrinsically different from a typical *DRT* one, since it is oriented towards a different stratum of users. Indeed, it is supposed to attract more travelers from automobiles than from conventional transit systems being implemented nowadays.

In addition, we need to also study the effect of "passenger-pooling" or "stop-pooling" where passengers join at the same pick up spot, regardless of their destination location. Here the idea is to measure the attractiveness of such a policy along with the use of fare incentives. Several of the money-based and disutility-based behaviors can be modeled using game techniques, as well.

Stated Preference User Response Study

A traditional SP survey design is not for models and parameters to be used directly for demand-side predictions (for which SP surveys may not yield reliable results), but rather in finding qualitative conclusions on overall acceptability and attractiveness. Furthermore, the SP survey results will be very useful in determining the relative importance of the factors involved, which is of significant use in further refining the system designs.

The set of factors and their relative importance (weight) will strongly depend upon user characteristics. Traditional *DRT* schemes, which are absolutely not competitive with auto travel, have been oriented to the service of small communities or passengers with specific requirements (elderly, disabled). This kind of user probably attaches importance to factors such as the existence of safety devices, acquaintanceship, quietude, social interaction, etc. (see Flannelly *et al*., 1991, Wegman, 1989, Kumar *et al*., 1991)

However, the objective of implementing a high-coverage *DRT* system like this is mainly to attract a different type of user. This will not happen unless the level of supply of vehicles is significantly high, and the time between call and service is significantly low to make even those who don't consider conventional transit as an option to select this system over personal automobiles. For them, we could expect a survey revealing people who associate importance to different factors such as travel time, travel time variability, waiting time and so on. These factors are precisely those that were envisioned when this system was originally designed.

Dynamic interactive simulator laboratory experiments of user behavior

In order to develop these preliminary behavioral models, a methodology for quantifying commuter behavior models from laboratory SP discrete choice survey is proposed. This model should be based on simulated travel conditions instead of real travel conditions. The major feature of this methodology is taking advantage of the simulation results in order to present the information needed to make a travel decision by an interviewee. This information will allow the interviewee to take a real decision of travel mode among the available choices, including the *HCPPT* system as a real alternative. The methodology is also useful in minimizing the bias resulting from asking about a hypothetical mode as a real alternative.

Extensive diary surveys of commuter behavior should be conducted based on simulation results. These surveys will provide data to characterize the day-to-day dynamics of commuter decisions in the laboratory, based on mode-switching behavior resulting from simulated *HCPPT* attributes, such as expected pick-up time, waiting time

at home and hub locations, ride time, etc. The simulation results will strongly depend on the time of service request (that is, whether it is a peak hour or off-peak hour trip), workplace conditions, socioeconomic characteristics and traffic system characteristics. Mahmassani and Jou (2000) gives an example of a study with a similar methodology.

With regard to the behavioral model itself, a joint estimation of two different day-to-day demand models is proposed, (see Jou and Mahmassani, 1998 for a similar estimation methodology). Given the contemporaneous correlation across the users, it does not seem reasonable to use the typical logit model, which assumes that the unobserved components of utility are independently and identically distributed. Instead, an estimation of a probit model is proposed, which has been derived by relaxing these assumptions about the unobserved components of utility. The two models to be estimated will be both based on simulation results and they correspond to the following user behavior decision experiments:

- The first probit model should be calibrated using appropriate variance-covariance structures, and the utility functions will be based on mode switching behavior, considering multiple choice in travel alternatives. Traditional estimation methods can be used, such as based on a Clarke approximation or a Monte Carlo method that uses random values for the unobserved component of the utility.
- The second model to be calibrated is a simple binary probit model, which should capture the behavior of accepting or rejecting the expected pickup time and travel time in the *HCPPT* system. In this case, the estimation is quite easier, and a standard maximum likelihood methodology could be used without major complications.

8.3.2 Adjustment of Control Strategies and Optimality

One fundamental improvement over the adaptive-predictive-control approach (APC) introduced in Chapter 5, would be to study the behavior of the optimal control formulation and to show optimality of such a scheme for different application areas related to systems like *HCPPT*. In other words, we need to develop guidelines on how to strive towards optimality in real-time transportation scheduling and routing problems depending on its characteristics.

Given a particular dynamic routing problem, some questions that immediately arise are the following:

- How to optimize this system?
- What kind of algorithms should be used?
- What kind of real-time control scheme would have the best performance?
- What type of cost function is more suitable for this type of problem?

The introduced algorithms are based on incremental costs formulations, which intuitively makes sense since the local incremental algorithm at any time is analogous to the gradient at a given point in classical optimization algorithms. The answers to these questions obviously need further analysis, but at this preliminary stage of this research, it is possible to propose some aspects that should be explored in the context of a future study, regarding

- Cost function formulation: Depending on the objectives of the system the cost function formulation could be modified. In most of the commercial applications the operator cost should be considered as an explicit variable, on the other hand, for evacuation purposes the operator cost may not be considered.
- Control schemes: Based upon the desired objective, the best control scheme should differ for each application. As stated in Chapters 5 and 7, the adoption of a predictive adaptive control scheme for real-time scheduling and routing could have promising results.
- Control policies: In this case, the parameters utilized should be different for each case. For instance, in the case of the *HCPPT* system, the area corresponding to a hub could maybe be increased depending on the demand level. For a more flexible operation, the main operations and decision ought to happen within the reroutable area and the number of vehicles should decrease along trunk corridors. The zoning and area size could also be sensitive to changes in demand. In theory, hub locations could also change in real-time with the demand pattern, which is not really feasible under a scheme like this, but could be possible to be implemented for different type of systems

All these questions, control schemes and optimality issues have to be further explored in order to adjust parameters, improve the cost formulations and ultimately, in order to improve the overall performance of the proposed transit system.

8.3.3 Improvement in Simulation Scheme

There are some additional issues that could be improved in the simulation itself. The most important aspects that require further analysis and development are mentioned next:

- The "passenger pooling" modeling proposed in Chapter 6 is based on some known discrete random variable distribution. However, in this context, it is possible to find a more appropriate manner of modeling the passenger-pooling behavior, incorporating thresholds regarding maximum allowable walking distance as well as maximum allowable waiting time, in order to decide how to group passengers over the simulation network.
- Several processes can be improved in the simulation platform. Nowadays, the most critical restrictions are the computation time and the necessary capabilities (in terms of RAM an power of the machines) in order to run the microsimulation along with the API developed here. In addition, the size of the network as well as the number of vehicles loaded at any time significantly restricted the implementation and functionality of the algorithms and routing procedures. In the near future it will be possible to implement better algorithms. For the stochastic cases in particular, it will be possible to relax some constraints and simplifications of the implemented algorithms, which should improve the solutions obtained.

- Implementation of a more sophisticated insertion algorithm (Section 4.3.3). In addition, the re-scheduling procedure (Section 4.4) has to be incorporated in the decision rules of assignment passenger-vehicle.

8.3.4 Other Issues

There are additional sources of interesting research from the different thoughts arose from this dissertations. The most important ones identified in this dissertation are the following:

- The feasibility of the proposed system and the efficiency of the algorithms developed in the context of this dissertation could be contrasted against an implementation of the multiple pickup and delivery problem with time windows in real-time, under similar conditions (benchmark scenario). The formulation and computational implementation of the benchmark scenario as well as the comparison with the heuristic solution obtained by running the algorithms developed here in the context of the *HCPPT* are very challenging issues.
- Development of new cost functions for the high-coverage type systems and estimation of the cost involved in the deployment schemes, essentially through the use of standard figures available for capital and operating expenses for such service, based on vehicle miles and vehicle hours (labor costs).
- Exploration of the details of coordinated public and private (contracted out) operations of the system and development of behavioral models for augmenting the

demand generation models. Qualitative insights from Delphi and Phone interview surveys will be very useful.

- Use of information technology and identification of technological devices. Initial qualitative evaluation of the relative benefits of alternate information technologies and devices.

REFERENCES

Abdulhai *et al*. (2002). Microsimulation Modeling and Impact Assessment of Streetcar Transit Priority Options: The Toronto Experience. *81ST Annual Meeting of the Transportation Research Board, Washigton D.C*., January 2002

Abramovich, D.Y. and L. G. Bushnell (1999). Report on the Fuzzy versus Conventional Control Debate. *IEEE Control Systems Magazine* **19**, No 3, pp. 8-10.

Adler M. and E. Ziglio (1996), *Gazing into the Oracle: The Delphi method and its application to social policy and public health.* London, Jessica Kingsley Publishers

Astrom, K.J. (1970). Introduction to Stochastic Control Theory. Academic Press. New York. USA.

Banks, J., J. Carson, and B. Nelson (1995). *Discrete-Event System Simulation*. Second Edition, Prentice Hall.

Bell, W., L.M. Dalberto, M.L. Fisher, A.J Greenfield, R. Jaikumar, P. Media, R.G. Mack and P.J. Prutzman (1983). Improving the Distribution of Industrial Gases with an Online Computarized Routing and Scheduling Optimizer, *Interfaces* **13**, pp. 4 - 23.

Bertsekas, D.P. (1991), An Auction Algorithm for Shortest Paths, *SIAM Journal on Optimization*, vol. 1, pp. 425-447.

Bertsekas, D.P., S. Pallotino, and M.G. Scutella (1995), Polynomial Auction Algorithms for Shortest Paths, *Computational Optimization and Applications*, vol. 4, pp. 99-125.

Betmead, R.R., M. Gevers and V. Wertz (1991). *Adaptive Optimal Control -The thinking man`s GPC*. Prentice Hall Inc., Englewood Cliffs, N. J.

Black, A.(1995). *Urban Mass Transportation Planning*, McGraw-Hill Series in Transportation, New York.

Bloomberg L. and J. Dale (2000). Comparison of VISSIM and CORSIM Traffic Simulation Models on a Congested Network.. *Transportation Research Record* **1727**, TRB, National Research Council, Washigton, D. C., pp 52-60.

Bodin, L., B. Golden, A. Assad and M. Ball (1983). Routing and scheduling of vehicles and crews. The state of the art, *Computers and Operations Research* **10**, pp. 69-211.

Borndörfer, R., M. Grötschel, F. Klostermeier and C. Küttner (1997). *Telebus Berlin: Vehicle scheduling in a dial-a-ride system*, in: Computer-Aided Transit Scheduling Proceedings, pp. 391-422.

Brockhoff, K.(1975): *The (Performance) of Forecasting Groups in Computer Dialogue and Face-to-Face Discussion*, in: Linstone, H.A.; Turoff, M. (Eds.): The Delphi Method. Techniques and Applications, Teading S. 291-321

Camacho, E.F. and C.Bordons (1995). *Model Predictive Control in Process Industry*. Springer - Verlag. London.

Cerulli, R., R. De Leone, and G. Piacente (1994), A Modified Auction Algorithm for the Shortest Path Problem, *Optimization Methods and Software*, vol. 4, pp. 209-224.

Chira-Chavala, T., and C. Venter (1997). Costa and Productivity Impacts of a Smart Paratransit System, *Transportation Research Record* **1571**, pp. 81-88.

Choa F., Milam R. T. and David Stanek (2002). CORSSIM, PARAMICS AND VISSIM: What the Manuals Never Told You. ITE Conference.

Clarke, D.W., C. Mohtadi and P. S. Tuffs (1987). Generalised Predictive Control - Part I. The basic Algorithm. *Automatica*. **23**, No 2, pp.137-148.

Clarke, D.W., C. Mohtadi and P. S. Tuffs (1987). Generalised Predictive Control - Part II. Extensions and Interpretations. *Automatica* **23**, No 2, pp. 149-160.

Crainic T. G., F. Malucelli and M. Nonato (2001). Flexible many-to-few + few-to-many = an almost personalized transit system. *TRISTAN IV*, São Miguel Azores Islands, pp. 435-440.

Cormen, T., C. Leiserson, and R. Rivest (1999). *Introduction to algorithms*, The MIT Press.

COMSIS Corporation (1988) "Cost Analysis Methodology for demand-responsive Service", Report prepared for Mass Transit Administration and the Maryland Dept. of Transportation.

Cullen, F.H., J.J. Jarvis and H.D. Ratliff (1981). Set partitioning based heuristics for interactive routing, *Networks* **11**, pp. 125-143.

Cutler, C.R. and B. C. Ramaker (1980). Dynamic Matrix Control - a Computer Control Algorithm. In *Joint Automatic Control Conference*. San- Francisco, USA.

Daganzo, C. (1978). An approximate analytic model of many-to-many demand responsive transportation systems, *Transportation Research* **12**, pp. 325-333.

Daganzo, C. (1984). Checkpoint dial-a-ride systems, *Transportation Research* **18B**(4/5), pp. 315-327.

Dawkins, J. and P.A.N. Briggs (1965). A method for using weighting functions as system description in optimal control. *Proceedings IFAC Symposium*. Teddington, U.K.

Desrochers, M., J.K. Lenstra, M.W. Savelsbergh, and F. Soumis (1988) Vehicle Routing with Time Windows: Optimization and Approximation. In B.L. Golden and A.A. Assad, editors, *Vehicle routing, methods and studies*, pp. 65 - 84, North Holland, Amsterdam.

Desrosiers, J., Y. Dumas and F. Soumis (1986). A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows, *American Journal Math. Management Sci.* **6**, pp. 301-325.

Desrosiers, Y. Dumas, M. Salomon and F. Soumis (1995). Time Constrained Routing and Scheduling. In *Handbooks in Operations Research / Management Science, Volume 8, Network Routing*, M. Ball, T. Magnanti, C. Monma, and G. Nemhauser (eds.), pp. 35-139, Elsevier Science Publishers B.V.

Dessouky, M. (2002). Benchmarking Demand Responsive Transit Systems, *presentation at Annual Meeting of the PATH Program*, Richmond, CA.

Dial, R.B. (1995). Autonomous dial-a-ride transit introductory overview, *Transportation Research* **3C**(5), pp. 261-275.

Dumas, Y., J. Desrosiers and F. Soumis (1991). The pickup and delivery problem with time windows, *European Journal of Operational Research* **54**, pp. 7-22.

M.L. Fisher and R. Jaikumar (1982). A Generalized Assignment Heuristic for Solving the VRP, *Networks* **11**, pp. 109.

Fischetti, M. and Toth P. (1989). An additive bounding procedure for combinatorial optimization problems, *Operational Research* **37**, pp. 319-328.

Flannelly K.J., M.S. Mc Leod, L. Flannelly and RW Behnke (1991). Direct comparison of commuter's interests in using different modes of transportation. *Transportation Research Record*, **No. 1321,** pp 90-96.

Fu, L. (2002). A simulation model for evaluating advanced dial-a-ride paratransit systems, *Transportation Research* **36A**, pp.291 - 307.

Gendreau, M. and J. Potvin (1999). Dynamic vehicle routing and dispatching, *Fleet Management and Logistics*, pp.115-126.

Gerland, H. (1991). FOCCS – Flexible operation command and control system for public transport. *Proceedings of Seminar H held at the PTRC 19th Summer Annual Meeting*, Sussex, England, Vol. P348, pp. 139-150.

Hanke, J.E., and A.G. Reitsch (1995). *Business Forecasting*. Prentice Hall, Englewood Cliffs, NJ 07632.

Helgason, R.V., J.L. Kennington, and B.D. Steward (1993), The One-to-One Shortest-Path Problem: An Empirical Analysis with the Two-Tree Dijkstra Algorithm, *Computational Optimization and Applications*, vol. 1, pp. 47-75.

Helmer O. (1977) Problems in futures research , *Futures*, **9(1),** pp.17-31.

Hickman, M. and K. Blume (2000). A method for scheduling integrated transit service, *8th International Conference on Computer-Aided Scheduling of public Transport (CASPT),* Berlin, Germany

Laura L. Higgins, L., J. Laughlin and K.F. Turnbull (2000). Automatic Vehicle Location and Advanced Paratransit Scheduling at Houston METROLift **,** *Proceedings of the 79th Transportation Research Board Annual Meeting, Washington D.C, January 2000, paper* 00-0956

Ho, Yu-Chi (1999). The No Free Lunch Theorem and the Human-Machine Interface. *IEEE Control Systems Magazine* **19**, No 3, pp. 88-90.

Horn, M.E.T. (2002). Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research* **10C**, pp. 35-63.

Horn, M.E.T. (2001). Multi-modal and demand-responsive passenger transport systems: a modeling framework with embedded control systems. *Transportation Research* **36A,** pp.167-188.

Inga Note (2001). Integration of a Center-Running Guided Busway into an Arterial Street. Presented at the 2001 VISSIM User Group Meeting, Seattle.

Ioachim, I., J. Desrosiers, Y. Dumas, M. Salomon, and D. Villeneuve (1995). A request clustering algorithm for door-to-door handicapped transportation, *Transportation Science* **29(1)**, pp. 63 - 78.

Jansson, J. O. (1980) A simple bus line model for optimization of service frequency and bus size, *Journal of Transport Economics and Policy* **14,** 53-80.

Jayakrishnan R., H. Mahmassani and T-Y Hu (1994) An evaluation tool for advanced traffic information and management systems in urban networks, *Transportation Research,* Vol. **2C**, pp. 129-147.

Jaw, J., A. Odoni, H. Psaraftis and N. Wilson (1986). A heuristic algorithm for the multi-vehicle many-to-many advance-request dial-a-ride problem, Working paper MIT-UMTA-82-3, M.I.T., Cambridge, MA.

Jaw, J., A. Odoni, H. Psaraftis and N. Wilson (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows, *Transportation Research* **20B**(3), pp. 243-257.

Jou, R. C., H. Mahmassani (1998). *Day-to-day dynamics of urban commuter departure time and route switching decisions: joint model estimation*. In: de Dios Ortuzar, J., Hensher, D., Jara-Diaz, S. (Eds.) Travel Behavior Research: Updating the State of Play, Ch. 20. Elsevier Science, Oxford.

Katende, E. and A. Jutan (1995). *Non-linear Predictive Control*. ACC Proc., FP10, Seattle, USA.

Khattak, A. and M. Hickman (1998). Automatic Vehicle Location and Computer Aided Dispatch Systems: Commercial Availability and Deployment in Transit Agencies, *Journal of Public Transportation* **2**, No. 1, pp. 1-26.

Kikuchi, S. (1984). Scheduling of demand-responsive transit vehicles, *Journal of Transportation Engineering* **110**(6), pp.511-520.

Kikuchi, S. and R. Donnelly (1990). Scheduling demand-responsive transportation vehicles using fuzzy-set theory, *Journal of Transportation Engineering* **118**(3), pp.391-409.

Kumar, A. and M. Moilov (1991). Vanpools in Los Angeles. *Transportation Research Record*, **No. 1321**, pp. 103-108.

Law and Kelton (1991), *Simulation Modeling and Analysis*, 2nd ed., McGraw Hill, New York.

Liaw, C., White, C. and J. Bander (1996). A decision support system for the bimodal dial-a-ride problem, *IEEE Transactions on System, Man and Cybernetics* **26**, pp. 552-565.

Lin, S. (1965). Computer solutions of the traveling salesman problem, *Bell System Technical Journal* **44**, pp. 2245 - 2269.

Lin, S. and B.W. Kernigham (1973). An effective heuristic algorithm for the traveling salesman problem, *Operations Research* **21**, pp. 498 - 516.

Liu, R., R.M. Pendyala and S. Polzin (1997). Assessment of intermodal transfer penalties using SP data, *Transportation Research Record* **1607**, pp. 74 - 80.

Liu, R., R.M. Pendyala and S. Polzin (1998). Simulation of the effects of intermodal transfer penalties on transit use, *Transportation Research Record* **1623**, pp. 88 - 95.

Louviere J.J., Hensher D.A.. and Swait JD (2000). *State choice methods*, Cambridge University Press.

Madsen, O.B.G., Raven, H.F., and J.M. Rygaard (1995). A heuristics algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives, *Annals of Operations Research* **60**, pp. 193-208.

Mahamassani, H.S., Jou, R.C. (2000) Transferring insights into commuter behavior dynamics from laboratory experiments to field surveys. *Transportation Research* **34A**, pp. 243-260.

Malucelli, F., M. Nonato and S. Pallottino (1999). *Demand Adaptive Systems: some proposals on flexible transit*. In Operational Research in Industry, T.A. Ciriani, *et al.*, Editors, McMillan Press: London, pp. 157-182.

Martin Sanchez, J.M. and Jose Rodellar (1996). *Adaptive Predictive Control: From the concepts to plant optimization.* Prentice Hall. London.

Model Predictive Control: Techniques and Applications (1999). *Materials of IEE Workshop. IEE*. London. April 27-30.

Multisystems Inc., Bus Rapid Transit Simulation Model Research and Development. Report USDOT/SBIR Phase 1. October 2000.

Nanry,W. and J.W. Barnes (2000). Solving the pickup and delivery problem with time windows using reactive tabu search, *Transportation Research* **34B**, pp. 107-121.

Newell, G.F. (1980). *Traffic flow on transportation networks*, MIT Press, Cambridge, Mass.

Nicholson, T.(1966), Finding the Shortest Route Between two Points in a Network, *The Computer Journal*, vol.9, pp 275-280.

Oh, Jun-Seok, C.E. Cortés, R. Jayakrishnan and Der-Horn Lee (2000). Microscopic Simulation with Large-Network Path Dynamics for Advanced Traffic Management and Information Systems, *Proceedings of the 6th International Conference on Applications of Advanced Technologies in Transportation Engineering, Singapore, June 2000*

Pallottino, S., and M.G. Scutella (1997), Dual Algorithms for the Shortest Path Tree Problem, *Networks*, vol. 29, pp. 125-133.

Pearl, J. (1984). Heuristics, *Addison-Wesley, Reading, MA.*

Polymenakos, L.C., and D.P. Bertsekas (1994), Parallel Shortest Paths Auction Algorithms, *Parallel Computing*, vol. 20, pp. 1221-1247.

Potter, B. (1976). Ann Arbor's dispatching system, *Transportation research Record* **608**, pp. 89-92.

Psaraftis, H.N. (1980). A dynamic programming solution to the single many-to-many immediate request dial-a-ride problem, *Transportation Science* **14**(2), pp. 130-154.

Psaraftis, H.N. (1983a). An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows, *Transportation Science* **17**(3), pp. 351-357.

Psaraftis, H.N. (1983b). Analysis of an $O(n^2)$ heuristic for the single vehicle many-to-many Euclidean dial-a-ride problem, *Transportation Research* **17B**(2), pp 133-145.

Psaraftis, H.N. (1986). Scheduling large-scale advance-request dial-a-ride systems. *American Journal of Math. And Mgmt. Sciences*, Vol. 6, No ¾pp . 327-367.

Psaraftis, H.N. (1988). Dynamic Vehicle Routing Problems. In B.L. Golden and A.A. Assad, editors, *Vehicle routing, methods and studies*, pp. 223 - 248, North Holland, Amsterdam.

Richalet, J., A. Rault, J. L. Testud, and J. Papon (1978). Model Predictive Heuristic Control: Application to Industrial Processes. *Automatica* **14(2)**, pp. 413-428.

Richalet, J., A. Rault, J. L.Testud, and J. Papon (1976). Algoritmic Control of Industrial Processes. In *4th IFAC Symposium on Identification and System Parameter Estimation*. Tbilisi, USSR.

Rosenkrantz, D.J., R.E. Stearns and P.M. Lewis II. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal of Computing*, Vol. 6, 1977, pp. 563-581.

Roy, S., J.M. Rousseau, G. Lapalme and J. Ferland (1984). Routing and Scheduling of Transportation Services for the Disabled: *Summary Report. Technical Report CRT-473A*, Centre de recherche sur les transports, Université of Montréal.

Ruland, K.S. and E.Y. Rodin (1997). The pickup and delivery problem - Faces and Branch-and-Cut algorithm. *Computers and Mathematics with Applications* **33 (12)**, pp. 1-13

Salomon, M. and J. Desrosiers (1988). Time window constrained routing and scheduling problems, *Transportation Science* **22**(1), pp. 1-13.

Savelsbergh, M.W.P. and M. Sol (1995). The general pickup and delivery problem, *Transportation Science* **29**(1), pp. 17-29.

Savelsbergh, M.W.P. and M. Sol (1998). DRIVE: dynamic routing of independent vehicles, *Operations Research* **46**(4), pp. 4747-490.

Schneider, J. B. and S. P. Smith(1981). Redesigning Urban Transit Systems: A Transit-Center-Based Approach. *Transportation Research Record* **798**, 56-65.

Sexton, T.R. and L.D. Bodin (1985a). Optimizing single many-to-many operations with desired delivery times: I Scheduling, *Transportation Science* **19**, pp. 378-410.

Sexton, T.R. and L.D. Bodin (1985b). Optimizing single many-to-many operations with desired delivery times: II Routing, *Transportation Science* **19**, pp. 411-435.

Smartest. Contract Number RO-97-SC.1059. Project funded by European Comission under the Transport RTD Programme of the 4th Framework Programme, 2000

Soeterboek, R. (1992). *Predictive Control: Unified Approach*. Prentice Hall.

Stein, D.M. (1978). Scheduling dial-a-ride transportation systems, *Transportation Science* **12**(3), pp. 232-249.

Stone, J.R., A. Nalevanko and J. Tsai (1993). Assessment of software for computerized paratransit operations, *Transportation Research Record* **1378**, pp. 1-9.

Teal, R.F. (1993). Implications of technological developments for demand responsive transit, *Transportation Research Record* **1390**, pp. 33-42.

Toth, P., and D. Vigo (1997). Heuristic algorithms for the handicapped person transportation problem, *Transportation Science* **31**, pp. 60-71.

Turvey, R. and H. Mohring (1975) Optimal bus fares, *Journal of Transport Economics and Policy* **9,** pp. 319-327.

Venglar, Fambro and Bauer (1995). Validation of Simulation Software for Modeling Light Rail Transit. *Transportation Research Record* **1494**, TRB, National Research Council, Washigton, D. C., pp 161-166.

Wegmann, F.J. (1989). Cost effectiveness of private employer ridesharing programs: An employers assessment. *Transportation Research Record*, **No. 1212**, pp. 88-100

Wilson, N.H.M. and Weissberg H. (1976). Advanced dial-a-ride algorithms research project: Final report. Report R76-20, Department of Civil Engineering, M.I.T., Cambridge, MA.

Wilson, N.H.M. and Colvin N.H. (1977). Computer control of Rochester dial-a-ride system. Report R77-31, Department of Civil Engineering, M.I.T., Cambridge, MA.

Wilson, N.H.M. and Miller E. (1977). Advanced dial-a-ride algorithms research project phase II: Interim report. Report R77-31, Department of Civil Engineering, M.I.T., Cambridge, MA.

APPENDIX TO CHAPTER 4

A4.1 Computation of the cumulative travel time on board

In this appendix section, let us focus on the analytical expression for the expected travel time experienced by all passengers on board when vehicle j is leaving a future stop k, called $TR_j(k)$. As usual, the current decision time is t, so let us define tP_l, for any group of passengers Z_l, the clock time at which they have been picked-up, if they were picked-up before the current decision time t. Hence, the travel interval between their pick-up time and the current time is just $tPI_l = t - tP_l$.

The calculations are split into three pieces: $TRK1_j(k)$ for those customers assigned to be picked-up by vehicle j, belonging to set $ZA_j(0)$; $TRK2_j(k)$ for all passengers belonging to set ZB_j and still on board when the vehicle leaves stop k; and $TRU_j(k)$ for the expected passengers to be assigned into the route between the current vehicle location v_j and stop k. The last part corresponds to the random component associated to future insertions.

Analytically for computing $TRK1_j(k)$,

$$TRK1_j(k) = \sum_{s=1}^{k-1} \left\{ \beta_j(s) \left(PS_{z_j(s)} - \sum_{i=s+1}^{k} \gamma_j(z_j(s), i) \right) \sum_{r=s}^{k-1} E[tS_j(r, r+1)] \right\} \qquad \text{(A4.1)}$$

where

$$\beta_j(s) = \begin{cases} 1 & if \;\; z_j(s) > 0 \;\wedge\; zp_j(s) = 0 \\ 0 & otherwise \end{cases} \tag{A4.2}$$

and

$$\gamma_j(l,i) = \begin{cases} 1 & if \;\; z_j(i) = l \;\wedge\; zp_j(i) > 0 \\ 0 & otherwise \end{cases} \tag{A4.3}$$

Expressions (A4.1), (A4.2) and (A4.3) depend on the one-to-one mappings defined in section 4.2, matching elements in the sequence CS_j and the corresponding group of customers. Condition (A4.2) is 1 when the stop s corresponds to a pick-up, and it is 0 in all other cases. Condition (A4.3), on the other hand, is 1 only when the i^{th} element in the sequence is associated to group of customers l, and it is a delivery. Expression (A4.1) is valid only if $k > 1$.

In case of $TRK2_j(k)$, the analytical expression turns out to be

$$\begin{aligned} TRK2_j(k) = & \sum_{r=1}^{LP_j(0)} \left\{ tPI_{zb_j(r)} + \sum_{s=0}^{k-1} E[tS_j(s,s+1)] \right\} + \\ & + \sum_{r=k}^{N_j} \delta_j(r) \left\{ tPI_{zb_j(zp_j(r))} + \sum_{s=0}^{k-1} E[tS_j(s,s+1)] \right\} \end{aligned} \tag{A4.4}$$

where

$$\delta_j(r) = \begin{cases} 1 & if \;\; z_j(r) = 0 \;\wedge\; zp_j(r) > LP_j(0) \\ 0 & otherwise \end{cases} \tag{A4.5}$$

In this case, the one-to-one mapping between elements in ZB_j and the corresponding group of customers has been included. Expression (A4.5) will become 1 if the r^{th} stop in

sequence CS_j match a delivery associated to the $zp_j(r)$ element of ZB_j, corresponding to the $zb_j(zp_j(r))$ group of customers.

Finally, for the random component $TRU_j(k)$, let us first define the unknown component of the vehicle load at stop k as $LU_j(k) = LPU_j(k) + LDIU_j(k)$, where both the unknown pick-ups and internal deliveries have been already defined in (4.39). Thus, $TRU_j(k)$ can be computed as follows

$$TRU_j(k) = \frac{LU_j(k)}{\sum_{r=0}^{k-1} LU_j(r)} \sum_{r=0}^{k-1} LU_j(r)\, E[tS_j(r, r+1)] \qquad \text{(A4.6)}$$

Thus, the cumulative travel-time of passengers on board when vehicle leaves stop k, $TR_j(k)$ can be calculated using expression (A4.7):

$$TR_j(k) = \begin{cases} TRK1_j(k) + TRK2_j(k) + TRU_j(k) & \text{if } k > 1 \\ TRK2_j(k) + TRU_j(k) & \text{if } k = 1 \\ TRK2_j(0) & \text{if } k = 0 \end{cases} \qquad \text{(A4.7)}$$

For the deterministic case in section 4.3.1, $TR_j(k) = TRK1_j(k) + TRK2_j(k)$, but replacing the expected travel time by simply its deterministic value. For the probabilistic case in section 4.3.2 on the other hand, $TRK_j(k) = TRK1_j(k) + TRK2_j(k)$ and $TRU_j(k)$ is directly computed using (A4.6).

In addition, for the current vehicle position at k = 0, the cumulative time is always known, and matches the component associated to passengers on board.

In such a case, $TR_j(0)$ can be split into external and internal trips based cumulative time. The former is the one used section 4.6.2.2 as part of the heuristics proposed to decide when to send vehicles to the trunk corridor. Analytically

$$TR_j(0) = TR_j^E(0) + TR_j^I(0) \tag{A4.8}$$

where

$$TR_j^E(k) = \sum_{r=1}^{LP_j(0)} tPI_{zb_j(r)} \qquad TR_j^I(k) = \sum_{r=LP_j(0)+1}^{L_j(0)} tPI_{zb_j(r)} \tag{A4.9}$$

A4.2 Computation of the expected travel time when vehicle leaves the reroutable portion of its route

In this section, a methodology to calculate the expected travel time that vehicle j takes from the last scheduled stop within the reroutable portion of its route to the next hub position in its schedule is conducted.

At any time, vehicle *j'* tasks while collecting and distributing passengers are determined by a sequence of stops CS_j as follows

$$CS_j = \{0,1,2,\ldots,N_j, N_j+1, N_j+2, N_j+3\} \tag{A4.10}$$

Stop 0 corresponds to the current position of vehicle j. The sequence is defined by N_j customer based stops, and three more terms accounting for the transition from the reroutable portion to the trunk line. Vehicle j does not cover the entire sequence when joining the trunk line. If it has to stop at its home hub (stop N_j+1) before proceeding towards its visit hub (stop N_j+3), it will skip stop N_j+2 of sequence in (A4.8), which represents the optimal entrance point along the trunk corridor from last stop N_j. On the

other hand, if vehicle j is scheduled to go directly towards its destination hub, it will skip the home hub stop $N_j + 1$.

The objective of this section is to compute $E[tS_j(N_j, N_j + 3)]$, that is, the expected surface travel time when vehicle is scheduled to go to its visit hub. This computation makes sense only in the stochastic case of section 4.3.2. In addition, if vehicle has to stop at its home hub first, the case becomes a normal reroutable segment expected travel time calculation. Notice that in such a case the next hub to stop will be always the home hub (assuming that all already assigned calls are not rescheduled to a different vehicle).

If the vehicle is expected to go to the visit hub, it could eventually deviate towards its home hub if a future pick-up insertion were assigned to it, containing at least one passenger whose destination hub were different from that of the vehicle.

Let us define g as the probability that the vehicle is deviated towards its home hub given that originally (at time t when vehicle is at position 0) it was scheduled to go straight towards its visit hub.

Then, such a probability can be estimated as follows

$$g = \left[PM(HD_j)\right]^K \tag{A4.11}$$

with $K = \sum_{k=0}^{N_j - 1} a_j(k, k+1) + a_j(N_j, N_j + 3)$ and $a_j(k, k+1) = E[ES] \cdot E[PI_j(k, k+1)]$

In addition, $PM_l(HD_j)$ represents the probability that an external trip destination hub matches the vehicle destination hub, that is for group Z_l

$$PM_l(HD_j) = \text{Prob}\left(HZD_l(k) = HD_j,\ \forall k \in ZE_l\right) \tag{A4.12}$$

This probability is assumed known from the system.

A graphic representation pf this decision process is shown next:

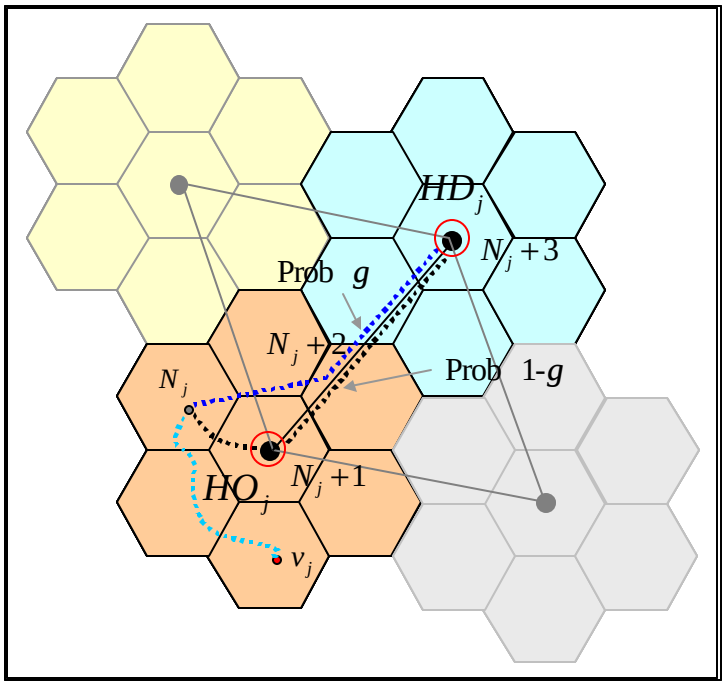

Figure A4.1 Transition from reroutable to non-reroutable portions of vehicle route

Then, under these conditions, the expected total travel time $E[tC_j(N_j, N_j+3)]$ for traveling from last stop N_j till the vehicle visit hub becomes a linear combination of two travel options. Analytically

$$\begin{aligned} E[tC_j(N_j, N_j+3)] &= g\left\{E[tS_j(N_j, N_j+)] + tTN(a_j(N_j+2), a_j(N_j+3))\right\} + \\ &+ (1-g)\left\{E[tS_j(N_j, N_j+1)] + HT(HO_j) + tTN(ah(HO_j), ah(HD_j))\right\} \end{aligned} \tag{A4.13}$$

or equivalently

$$E[tC_j(N_j, N_j+3)] = \gamma\, E[tS_j(N_j, N_j+2)] + (1-\gamma)E[tS_j(N_j, N_j+1)] + \\ + \gamma\left\{tTN(a_j(N_j+2), a_j(N_j+3)\right\} + (1-\gamma)\left\{HT(HO_j) + tTN(ah(HO_j), ah(HD_j))\right\} \tag{A4.14}$$

which can be written as

$$E[tC_j(N_j, N_j+3)] = E[tS_j(N_j, N_j+3)] + E[tT_j(N_j, N_j+3)] \tag{A4.15}$$

where

$$\begin{aligned} E[tS_j(N_j, N_j+3)] &= \gamma\, E[tS_j(N_j, N_j+2)] + (1-\gamma)E[tS_j(N_j, N_j+1)] \\ E[tT_j(N_j, N_j+3)] &= \gamma\left\{tTN(a_j(N_j+2), a_j(N_j+3)\right\} + \\ &\quad + (1-\gamma)\left\{HT(HO_j) + tTN(ah(HO_j), ah(HD_j))\right\} \end{aligned} \tag{A4.16}$$

Note that the surface (reroutable) travel time is function of the probability γ, which is also function of the expected number of insertions. However, the expected number of insertions on the other hand is function of the segment travel times. Thus, for the last segment

$$E[PI_j(N_j, N_j+3)] = g_1\left(E[tS_j(N_j, N_j+3)]\right) \tag{A4.17}$$

$$= g_1\left(\ \gamma E[tS_j(N_j, N_j+2)] + (1-\gamma)E[tS_j(N_j, N_j+1)]\ \right)$$

and

$$\gamma = g_2\left(\ ProbM(HD_j), \sum_{k=0}^{N_j-1} E[PI_j(k, k+1)], E[PI_j(N_j, N_j+3)]\ \right) \tag{A4.18}$$

Expressions (A4.17) and (A4.18) justify a iterative process as described in the following chart

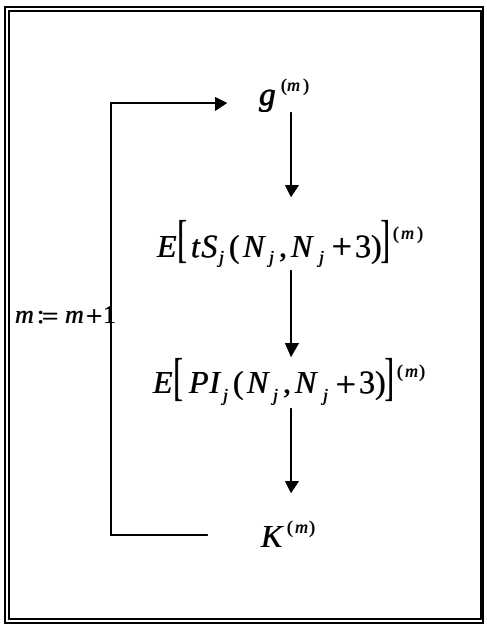

Figure A4.2 Iterative process to compute the expected travel time

Once a desired convergence level is reached, say after iteration *M*, we can compute $E[PI_j(N_j, N_j+3)]^{(M)}$ and $g^{(M)}$. Thus, $E[tS_j(N_j, N_j+3)]$ can be calculated consistently using those final values in expression (A4.14).

APPENDIX TO CHAPTER 5

A5.1 The probit and logit models (source Maddala, 1985)

In this section, a brief introduction to the probit analysis model by Goldberger (1964) is introduced, according to the excellent discussion and summary by Maddala (1985).

Let us assume that there is an underlying response variable y_i^* defined by the regression relationship

$$y_i^* = \boldsymbol{b}' x_i + m_i \tag{A5.1}$$

In practice, y_i^* is unobservable. What we observe is a dummy variable y defined by

$$\begin{aligned} y &= 1 \quad \textit{if } y_i^* > 0 \\ y &= 0 \quad \textit{otherwise} \end{aligned} \tag{A5.2}$$

In this formulation, $\boldsymbol{b}' x_i$ is not $E[y_i / x_i]$ as in the typical linear probability model; it is $E[y_i^* / x_i]$. From the relations (A5.1) and (A5.2) the following relation is obtained

$$\begin{aligned} Prob(y_i = 1) &= Prob\,(m_i > -\boldsymbol{b}' x_i) \\ &= 1 - F(-\boldsymbol{b}' x_i) \end{aligned} \tag{A5.3}$$

where F is the cumulative distribution function for m.

In this case the observed values of y are just realizations of a binomial process with probabilities given by (A5.3) and varying from trial to trial (depending on x_i). Hence, the likelihood function is

$$L = \prod_{y_i=0} F(-\beta' x_i) \prod_{y_i=1} [1 - F(-\beta' x_i)] \tag{A5.4}$$

The functional form for F in (A5.4) will depend on the assumptions made about μ_i in (A5.1). If the cumulative distribution of μ_i is the logistics, the *logit model* is obtained. In this case,

$$F(-\beta' x_i) = \frac{exp(-\beta' x_i)}{1 + \exp(-\beta' x_i)} = \frac{1}{1 + \exp(\beta' x_i)} \tag{A5.5}$$

Hence,

$$1 - F(-\beta' x_i) = \frac{\exp(\beta' x_i)}{1 + \exp(\beta' x_i)} \tag{A5.6}$$

Note that in this case there is a closed-form expression for F, because it does not involve integrals explicitly. Not all distributions permit such a closed-form expression. For instance, in the *probit model* (or, more accurately, the *normit* model) μ_i are assumed $IN(0, \sigma^2)$. Then,

$$F(-\beta' x_i) = \int_{-\infty}^{-\beta' x_i/\sigma} \frac{1}{(2\pi)^{1/2}} \exp(-\frac{t^2}{2}) dt \tag{A5.7}$$

It can be easily seen from (A5.7) and the likelihood function (A5.4) that only β/σ can be estimated, and not β and σ separately. Hence, $\sigma = 1$ is assumed to start with.

Because the cumulative normal distribution and the logistics distribution are very close to each other, except at the tails, we are not likely to get very different results using (A5.6) or (A5.7), that is the logit or the probit method, unless the samples are so large (so that there are enough observations at the tails). However, the estimates of β from the two methods are not directly comparable. Because the logistics distribution has a

variation $\pi^2/3$, the estimates of β obtained from the logit model have to be multiplied by $3^{1/2}/\pi$ to be comparable to the estimates obtained from the probit model (where σ is normalized to be equal to 1).

Amemiya (1981) suggested that the logit estimates be multiplied by $1/1.6 = 0.625$, instead of $(3^{1/2}/\pi)$ saying that this transformation produces a closer approximation between the logistic distribution and the distribution function of the standard normal. He also suggested that the coefficients of the linear probability model $\hat{\beta}_{LP}$ and the coefficients of the logit model $\hat{\beta}_L$ are related by the relationships

$$\hat{\beta}_{LP} \approx 0.25\hat{\beta}_L \text{ except for the constant term}$$

$$\hat{\beta}_{LP} \approx 0.25\hat{\beta}_L + 0.5 \text{ for the constant term}$$

Thus, if one needs to make $\hat{\beta}_{LP}$ comparable to the probit coefficients, it is needed to multiply them by 2.5 and subtract 1.25 from the constant term.

An alternative way of comparing the models would be to (a) calculate the sum of squared deviation from predicted probabilities, (b) compare the percentages correctly predicted, and (c) look at the derivatives of the probabilities with respect to a particular independent variable. Let x_{ik} be the kth element of the vector of explanatory variables x_i, and let β_k be the k^{th} element of β. Then the derivatives for the probabilities given by the linear probability model, probit model, and logit model are, respectively,

$$\frac{\partial}{\partial x_{ik}}(x_i'\boldsymbol{\beta}) = \beta_k \tag{A5.8}$$

$$\frac{\partial}{\partial x_{ik}}\Phi(x_i'\boldsymbol{\beta}) = f(x_i'\boldsymbol{\beta})\beta_k \tag{A5.9}$$

$$\frac{\partial}{\partial x_{ik}}L(x_i'\boldsymbol{\beta}) = \frac{\exp(x_i'\boldsymbol{\beta})}{[1+\exp(x_i'\boldsymbol{\beta})]^2}\beta_k \tag{A5.10}$$

These derivates will be needed for predicting the effects of changes in one of the independent variables on the probability of belonging to a group. In the case of the linear probability model, these derivatives are constant. In the case of the probit and logit models, they have to be calculated at different levels of the explanatory variables to get an idea of the range of variation of the resulting changes in the probabilities.

Logit model

For the logit model, the likelihood function (A5.4) can be written as

$$L = \prod_{i=1}^{n}\left(\frac{1}{1+\exp(\boldsymbol{\beta}'x_i)}\right)^{1-y_i}\left(\frac{\exp(\boldsymbol{\beta}'x_i)}{1+\exp(\boldsymbol{\beta}'x_i)}\right)^{y_i} = \frac{\exp(\boldsymbol{\beta}')\sum_{i=1}^{n}x_i y_i}{\prod_{i=1}^{n}[1+\exp(\boldsymbol{\beta}'x_i)]} \tag{A5.11}$$

Let us define $t^* = \sum_{i=1}^{n} x_i y_i$. To find the maximum-likelihood (ML) estimate of $\boldsymbol{\beta}$, let us take log to both sides of (A5.11) getting

$$\log L = \boldsymbol{\beta}' t^* - \sum_{i=1}^{n}\log[1+\exp(\boldsymbol{\beta}'x_i)] \tag{A5.12}$$

Hence, $\frac{\partial \log L}{\partial \boldsymbol{b}} = 0$ gives

$$S(\boldsymbol{b}) = -\sum \frac{\exp(\boldsymbol{b}' x_i)}{1+\exp(\boldsymbol{b}' x_i)} x_i + t^* = 0 \qquad \text{(A5.13)}$$

These equations are nonlinear in $\boldsymbol{b}$. Hence either the Newton-Raphson method or the scoring method has to be used to solve the equations. The information matrix is

$$I(\boldsymbol{b}) = E(-\frac{\partial^2 \log L}{\partial \boldsymbol{b} \partial \boldsymbol{b}'}) = \sum_{i=1}^{n} \frac{\exp(\boldsymbol{b}' x_i)}{[1+\exp(\boldsymbol{b}' x_i)]^2} x_i x_i' \qquad \text{(A5.14)}$$

Starting with some initial value of $\boldsymbol{b}$, say $\boldsymbol{b}_0$, $S(\boldsymbol{b}_0)$ and $I(\boldsymbol{b}_0)$ are computed. Then the new estimate of $\boldsymbol{b}$ is, by the method of scoring,

$$\boldsymbol{b}_1 = \boldsymbol{b}_0 + [I(\boldsymbol{b}_0)]^{-1} S(\boldsymbol{b}_0) \qquad \text{(A5.15)}$$

In practice, both $I(\boldsymbol{b}_0)$ and $S(\boldsymbol{b}_0)$ are divided by *n*, the sample size. This iterative procedure is repeated until convergence. In the present case it is clear that $I(\boldsymbol{b})$ is positive definite at each stage of iteration. Hence, the iterative procedure will converge to a maximum of the likelihood function, no matter what the starting value is. If the final converged estimates are denoted by $\hat{\boldsymbol{b}}$, then asymptotic covariance matrix is estimated by $[I(\hat{\boldsymbol{b}})]^{-1}$. These estimated variances and covariances will enable us to test hypotheses about the different elements of $\hat{\boldsymbol{b}}$.

After estimating $\boldsymbol{b}$, estimated values of the probability that the ith observation is equal to 1 can be obtained. Denoting these estimated values by $\hat{p}_i$, then

$$\hat{p}_i = \frac{\exp(\hat{\boldsymbol{b}}' x_i)}{1+\exp(\hat{\boldsymbol{b}}' x_i)} \tag{A5.16}$$

Equation (A5.16) shows that

$$\sum \hat{p}_i x_i = \sum y_i x_i \tag{A5.17}$$

Thus, if x_i includes a constant term, then the sum of the estimated probabilities is equal to $\sum y_i$ or the number of observations in the sample for which $y_i = 1$. In other words, the predicted frequency is equal to the actual frequency. Similarly, if x_i includes a dummy variable, say 1 for female, 0 for male, the predicted frequency will be equal to the actual frequency for each sex group. Similar conclusions follow for the linear probability model by virtue of the fact that equations (A5.17) are the least-squares normal equations in that case.

In any case, after estimating $\hat{\boldsymbol{b}}$ and then $\hat{p}_i$ by the logit model, it is always good practice to check whether or not equations (A5.17) are satisfied.

Probit model

For the probit model, expression (A5.7) is substituted in equation (A5.4).

Let us denote by $f(\cdot)$ and $\Phi(\cdot)$ the density function and the distribution function, respectively, of the standard normal. Then for the probit model the likelihood function corresponding to (A5.11) is

$$L = \prod_{i=1}^{n} [\Phi(\boldsymbol{\beta}' x_i)]^{y_i} [1 - \Phi(\boldsymbol{\beta}' x_i)]^{1-y_i} \tag{A5.18}$$

and the log-likelihood is

$$\log L = \sum_{i=1}^{n} y_i \log \Phi(\boldsymbol{\beta}' x_i) + \sum_{i=1}^{n} (1 - y_i) \log [1 - \Phi(\boldsymbol{\beta}' x_i)] \tag{A5.19}$$

Differentiating *log L* with respect to $\boldsymbol{\beta}$ yields

$$S(\boldsymbol{\beta}) = \frac{\partial log\, L}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{[y_i - \Phi(\boldsymbol{\beta}' x_i)]}{\Phi(\boldsymbol{\beta}' x_i)[1 - \Phi(\boldsymbol{\beta}' x_i)]} f(\boldsymbol{\beta}' x_i) x_i \tag{A5.20}$$

The ML estimator $\hat{\boldsymbol{\beta}}_{ML}$ can be obtained as a solution of the equations $S(\boldsymbol{\beta}) = 0$.

These equations are nonlinear in $\boldsymbol{\beta}$, and thus we have to solve them by an iterative procedure. The information matrix is

$$I(\boldsymbol{\beta}) = E(-\frac{\partial^2 \log L}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'}) = \sum_{i=1}^{n} \frac{[f(\boldsymbol{\beta}' x_i)]^2}{\Phi(\boldsymbol{\beta}' x_i)[1 - \Phi(\boldsymbol{\beta}' x_i)]} x_i x_i' \tag{A5.21}$$

As with the logit model, we start an initial value of $\boldsymbol{\beta}$, say $\boldsymbol{\beta}_0$, and compute the values $S(\boldsymbol{\beta}_0)$ and $I(\boldsymbol{\beta}_0)$. Then the new estimate of $\boldsymbol{\beta}$ is, by the method of scoring

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 + [I(\boldsymbol{\beta}_0)]^{-1} S(\boldsymbol{\beta}_0) \tag{A5.22}$$

Note that $I(\boldsymbol{\beta})$ is positive definite at each stage of the iteration. Hence, the iterative procedure will converge to a maximum of the likelihood function no matter what the

starting value is. If the final converged estimates are denoted by $\hat{\boldsymbol{b}}$, then the asymptotic covariance matrix is estimated by $[I(\hat{\boldsymbol{b}})]^{-1}$. These can be used to conduct any tests of significance.

A5.2 Catchment area definition and computation of the weighted available space in the proximity of a vehicle route segment

In this section two relevant issues are addressed: first, the definition of the catchment area associated to a segment $(k,k+1) \in CS_j$ on the reroutable portion of the vehicle j's route is computed. Second, and associated to the first concept, it is analytically formulated a measure for the weighted available space (in seats) in the proximity of vehicle segment $(k,k+1)$ by the time vehicle j is expected to arrive there. This variable is called $WSP_j(k,k+1)$.

Catchment area concept

The definition of the catchment area is quite simple. In order to simplify the models and have a consistent formulation of the branched process methodology discussed in Chapter 5, it is advisable to define a spatial area around every predefined vehicle segment, in which the dispatching module is more likely to insert a new call to that specific vehicle segment. In the formulation of this chapter, it has been assumed that insertions beyond a defined catchment area are very unlikely to happen, and therefore they have been overlooked.

Analytically, it has been assumed that an elliptical form is a reasonable way to define a catchment or influence area associated to a specific segment $(k, k+1)$ of the vehicle sequence, as shown in Figure A5.1.

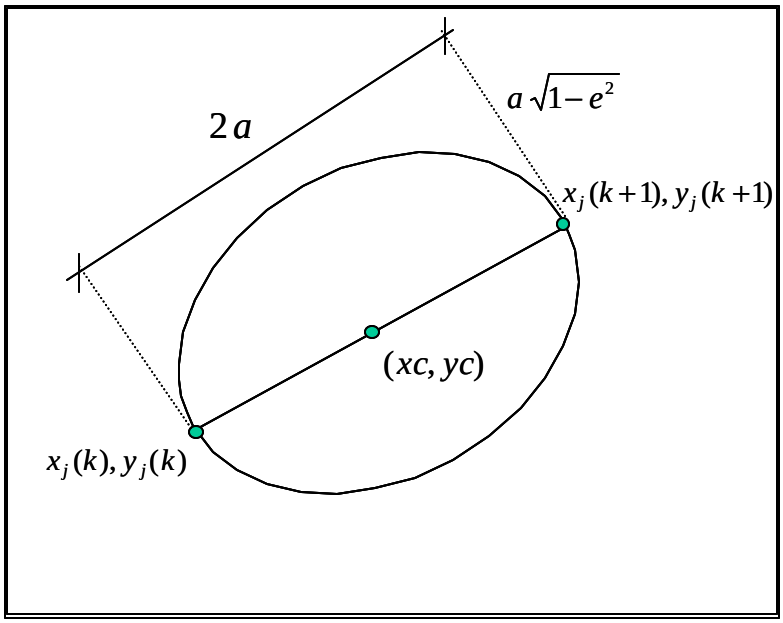

Figure A5.1 Segment $(k, k+1)$ catchment area

The ellipse is completely defined by its extreme points determined by stops k and k+1 on the longest axis. As mentioned in the chapter text, the eccentricity of the ellipse (e) is a parameter that could be calibrated based on the observed dispatch decisions, and eventually it could be part of the adaptive procedure as well.

Theoretically, the catchment area should depend on the real network topology as well as on network traffic conditions, however for simplicity, it is assumed that the spatial and geometrical features of the ellipse are enough to capture the boundaries of the dispatch decision scope.

The coordinates of both stops are completely defined by their spatial coordinates $(x_j(k), y_j(k))$ and $(x_j(k+1), y_j(k+1))$. Let us define the Euclidean distance between both stops as

$$D_j(k,k+1) = \sqrt{(x_j(k+1) - x_j(k))^2 + (y_j(k+1) - y_j(k))^2} \tag{A5.23}$$

In addition, the coordinates of the center point (xc, yc) are given by

$$xc = \frac{x_j(k) + x_j(k+1)}{2} \qquad yc = \frac{y_j(k) + y_j(k+1)}{2} \tag{A5.24}$$

Note that stops k and k+1 have any position with respect to the origin point of the system of coordinates; therefore, the proper translation and rotation corrections have to be introduced to the standard equation of the ellipse in order to account for such a change of coordinates. Finally, the corrected equation of the ellipse shown in Figure A5.1 is written for an arbitrary point (x, y). Thus, (x, y) belongs to the ellipse curve if and only if the following condition is fulfilled

$$\frac{4}{D_j(k,k+1)^2}\left[x'^2 + \frac{y'^2}{(1-e^2)}\right] = 1 \tag{A5.25}$$

where

$$x' = \left(\ x - xc\ \right)\frac{\left(x_j(k+1) - x_j(k)\right)}{D_j(k,k+1)} + \left(\ y - yc\ \right)\frac{\left(y_j(k+1) - y_j(k)\right)}{D_j(k,k+1)}$$

$$y' = \left(\ y - yc\ \right)\frac{\left(x_j(k+1) - x_j(k)\right)}{D_j(k,k+1)} + \left(\ x - xc\ \right)\frac{\left(y_j(k+1) - y_j(k)\right)}{D_j(k,k+1)}$$

and $0 \le e < 1$. When $e = 0$ the ellipse becomes a circle. For an arbitrary point (xs, ys) to belong to the catchment area of segment $(k, k+1) \in CS_j$, the following condition has to be satisfied

$$\frac{4}{D_j(k,k+1)^2}\left[xs'^2 + \frac{ys'^2}{(1-e^2)}\right] \le 1 \qquad \text{(A5.26)}$$

where

$$xs' = \left(xs - xc\right)\frac{\left(x_j(k+1) - x_j(k)\right)}{D_j(k,k+1)} + \left(ys - yc\right)\frac{\left(y_j(k+1) - y_j(k)\right)}{D_j(k,k+1)}$$

$$ys' = \left(ys - yc\right)\frac{\left(x_j(k+1) - x_j(k)\right)}{D_j(k,k+1)} + \left(xs - xc\right)\frac{\left(y_j(k+1) - y_j(k)\right)}{D_j(k,k+1)}$$

If (A5.26) is fulfilled, then spatial location (xs, ys) belongs to the catchment area defined above. Otherwise, it does not.

The total surface of the catchment area is equal to $AR_j(k,k+1) = \pi \frac{D_j^2(k,k+1)}{4}\sqrt{1-e^2}$.

The eccentricity e is the only parameter that could be calibrated as part of the learning process developed in this chapter, by observing the dispatch algorithm decisions. Thus, if there are not many vehicles available at certain time, the dispatching module could start sending vehicles to pick-up calls generated far from the straight line between k and k+1, in which case $e \to 0$. In other cases, the observed decisions taken by the algorithm could slightly deviate already scheduled segments if the density of vehicles were extremely high in some areas for certain periods, in which case an

eccentricity value close to one could be reasonable. A simple way to guess a value for this parameter without adding extreme complexity to the computation and based on the observed evolution of the dispatch decisions could be to adjust e such that p % of the observed insertions happens within the corresponding catchment area, according to equation (A5.26). The limit p would depend on the accuracy of the system performance, say 90 or 95 %.

In the next subsection, a methodological approach is used in order to compute the indicator $WSP_j(k,k+1)$, which measures the available vehicle capacity (measured in seats) in the proximity (meaning in space-time) of a specific vehicle segment $(k,k+1)$. The computations will explore competitive routes only within the catchment area of such a segment according to the definition proposed above.

The indicator $WSP_j(k,k+1)$ plays a very important role in the prediction process described in this chapter. In fact, the final objective of the APA process is to model the dispatch algorithm mechanism and to update such a model from observed dispatch decisions in real-time. An important premise is that part of such a decision process is based on the conditions of other vehicles competing for customers within a reasonable influence area when analyzing certain vehicle segment under the APA framework. In other words, if there are a lot of vehicles close by (with available seats on them), a future insertion should be less likely to happen than in a scenario with just few vehicles around, impacting the predicted expected number of insertions on that vehicle segment. This part of the dispatch process is mostly captured by $WSP_j(k,k+1)$ in the APA learning methodology described in the chapter body.

Next, the deterministic case is formulated for the sake of simplicity. A deterministic value of $WSP_j(k,k+1)$ will be needed when taking data from the observed system as explained in Section 5.6. The extension to the stochastic case is straightforward, and will be addressed later in this Appendix.

Weighted available space calculation: determinstic case

The deterministic expression for $WSP_j(k,k+1)$ is useful to clarify the basic concept, and is directly utilized when taking data from the system as described in Section 5.6.

Let us consider another vehicle m with the same home hub as that of vehicle j, that is $HO_m = HO_j$. Vehicle m has assigned to follow a stop sequence CS_m and currently is within its reroutable portion.

Unlike the formulation shown above, in this case a combined space-time condition has to be satisfied for a point, and more specifically for a segment (in this case, a segment $(s,s+1) \in CS_m$) to be considered in the calculation of $WSP_j(k,k+1)$.

Geometrically, a space-time segment $(s,s+1)$ (or a portion of such a segment) in the route of vehicle m will be considered as part of the $WSP_j(k,k+1)$ calculation if and only at least one of the stops defining it falls within the volume of the geometrical body (in space-time dimensions) depicted in Figure A5.2.

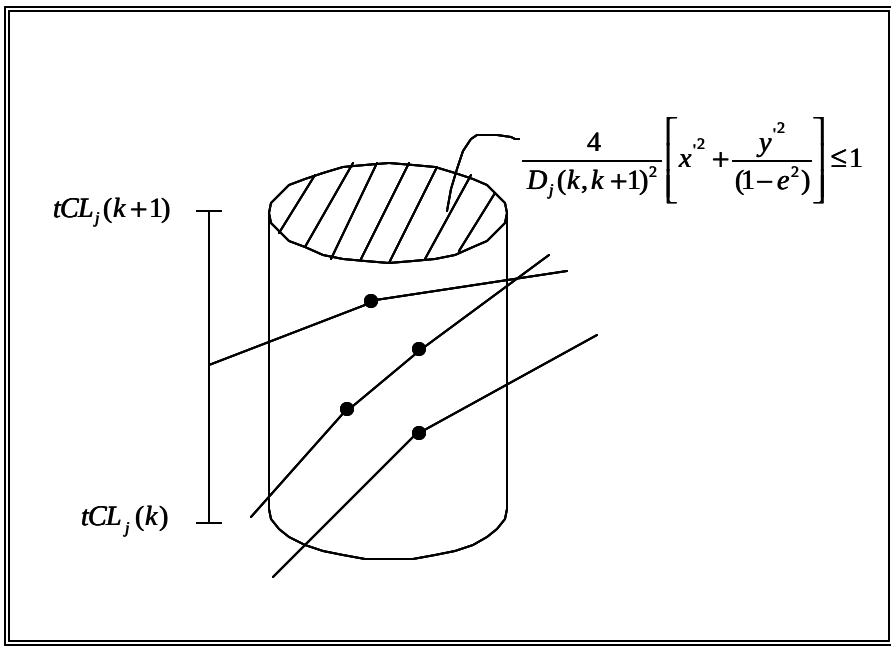

Figure A5.2 Geometrical body associated to vehicle *j*'s segment $(k, k+1)$

The vertical axis represents time, and the horizontal plane accounts for two-dimensional space *x* and *y*. As defined in Chapter 4, $tCL_j(k)$ is the clock time at which vehicle *j* is expected to arrive to stop *k*.

First, the time condition is checked. The order in which the conditions are checked is chosen just because time conditions are easier to compute. Analytically, there are three cases.

Case 0: if $tCL_m(s+1) < tCL_j(k) \quad \vee \quad tCL_m(s) > tCL_j(k+1)$ go to next segment-vehicle

or set $WT_{mj}(s, s+1) = 0$

else

Case 1: if $tCL_m(s) < tCL_j(k)$:

1.1)

$$tCL_j(k) < tCL_m(s+1) < tCL_j(k+1) \Rightarrow WT_{mj}(s,s,+1) = tCL_m(s+1) - tCL_j(k)$$

1.2) $tCL_m(s+1) \geq tCL_j(k+1) \Rightarrow WT_{mj}(s,s+1) = tCL_j(k+1) - tCL_j(k)$

Case 2: $tCL_j(k) \leq tCL_m(s) < tCL_j(k+1)$

2.1)

$$tCL_j(k) < tCL_m(s+1) < tCL_j(k+1) \Rightarrow WT_{mj}(s,s,+1) = tCL_m(s+1) - tCL_j(s)$$

2.2) $tCL_m(s+1) \geq tCL_j(k+1) \Rightarrow WT_{mj}(s,s+1) = tCL_j(k+1) - tCL_j(s)$

$WT_{mj}(s,s+1)$ is the weight in time of segment $(s,s+1)$ associated to vehicle m route, in the calculation of $WSP_j(k,k+1)$.

Once the time condition is fulfilled for a portion of segment $(s,s+1)$ (basically, Case 1 or Case 2), the spatial condition has to be checked as well. In equations:

Case 1': if $[x_m(s), y_m(s)]$ and $[x_m(s+1), y_m(s+1)]$ both satisfy (A5.26)

$$\Rightarrow WS^2_{mj}(s,s+1) = \left(x_m(s+1) - x_m(s)\right)^2 + \left(y_m(s+1) - y_m(s)\right)^2$$

Case 2': else if neither $[x_m(s), y_m(s)]$ nor $[x_m(s+1), y_m(s+1)]$ satisfy (A5.26)

$$\Rightarrow WS^2_{mj}(s,s+1) = 0$$

Case 3': if either $[x_m(s), y_m(s)]$ (Case 3.1') or $[x_m(s+1), y_m(s+1)]$ (Case 3.2) satisfies (A5.26) then find l^* that solves the following problem:

$$\frac{4}{D_j(k,k+1)^2}\left[x_I'^2 + \frac{y_I'^2}{(1-e^2)}\right] = 1 \qquad \text{(A5.27)}$$

where

$$x_I^{'} = \left(x_I - xc \right) \frac{\left(x_j(k+1) - x_j(k) \right)}{D_j(k,k+1)} + \left(y_I - yc \right) \frac{\left(y_j(k+1) - y_j(k) \right)}{D_j(k,k+1)}$$

$$y_I^{'} = \left(y_I - yc \right) \frac{\left(x_j(k+1) - x_j(k) \right)}{D_j(k,k+1)} + \left(x_I - yc \right) \frac{\left(y_j(k+1) - y_j(k) \right)}{D_j(k,k+1)}$$

and $x_I = x_m(s) + \lambda \left(x_m(s+1) - x_m(s) \right)$

$y_I = y_m(s) + \lambda \left(y_m(s+1) - y_m(s) \right)$

Finally,

if Case 3.1' $\Rightarrow$ $WS_{mj}^2(s,s+1) = \lambda^{*2} \left\{ \left(x_m(s+1) - x_m(s) \right)^2 + \left(y_m(s+1) - y_m(s) \right)^2 \right\}$

else if Case 3.2' $\Rightarrow WS_{mj}^2(s,s+1) = \left(1 - \lambda^* \right)^2 \left\{ \left(x_m(s+1) - x_m(s) \right)^2 + \left(y_m(s+1) - y_m(s) \right)^2 \right\}$

$WS_{mj}(s,s+1)$ is the weight in space of segment $(s,s+1)$ of vehicle *m* route, in the calculation of $WSP_j(k,k+1)$. Note that Case 3.1' means that only stop *s* falls within the ellipse of Figure A5.1. Case 3.2' has the same meaning but for stop *s* +1.

Thus, an influence factor is defined as a Euclidean norm considering both time and space dimensions. Analytically

$$F_{mj}(s,s+1) = \begin{cases} \sqrt{WT_{mj}^2(s,s+1) + WS_{mj}^2(s,s+1)} & \text{if } WT_{mj}(s,s+1) \cdot WS_{mj}(s,s+1) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Then, the weighted available space is defined as a function of the factor computed above and the deterministic measure of the available space on vehicle m segment $(s, s+1)$ defined as

$$SP_m(s, s+1) = L_{MAX} - \left\{ L_m(s) + \left[LRES_m(s+1) \right]^+ \right\} \tag{A5.28}$$

Analytically,

$$WSP_j(k+1) = \frac{\sum_{\substack{m \in R(HO_j) \\ m \neq j}} \sum_{s=0}^{N_m - 1} F_{mj}(s, s+1) \cdot SP_m(s, s+1)}{\sum_{\substack{m \in R(HO_j) \\ m \neq j}} \sum_{s=0}^{N_m - 1} F_{mj}(s, s+1)} \tag{A5.29}$$

Weighted available space calculation: extension to stochastic case

The extension to the stochastic case is straightforward. Expression (A5.28) should be replaced by (5.23), and the clock times in all time calculations should be replaced by the corresponding expected clock time values defined in Chapter 4, equation (4.43) that includes the extra delay due to future insertions.

That is what should be done in theory, however in practice it is only possible to work with expected values associated to vehicle j. All other vehicles-segment components influencing $WSP_j(k, k+1)$ will be deterministic since that is the only available information about other vehicles when working with vehicle j.