

# UC Berkeley

## Earlier Faculty Research

**Title**

Algorithms and Strategies for Dynamic Carrier Fleet Operations: Applications to Local Trucking Operations

**Permalink**

<https://escholarship.org/uc/item/5bx091pt>

**Author**

Wang, Xiubin

**Publication Date**

2001

**UNIVERSITY OF CALIFORNIA  
IRVINE**

**Algorithms and Strategies for Dynamic Carrier Fleet Operations: Applications to  
Local Trucking Operations**

**DISSERTATION**

**submitted in partial satisfaction of the requirements for the degree of**

**DOCTOR OF PHILOSOPHY**

**in Civil Engineering**

**by**

**Xiubin Wang**

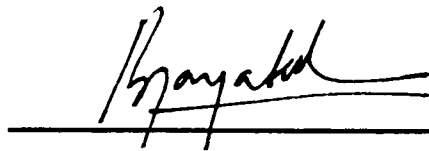
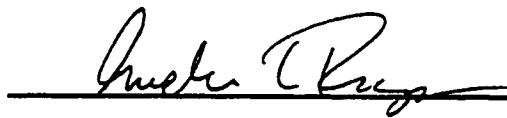
**Dissertation Committee:**

**Professor Amelia C. Regan, Chair  
Professor Wilfred Recker  
Professor R. Jayakrishnan**

**2001**

**Portion of Chapter 2 © Pergamon Press Ltd.**  
**Portion of Chapter 3 © Transportation Research Board**  
**All other chapters © Xiubin Wang, 2001**  
**All rights reserved**

The dissertation of Xiubin Wang is approved  
and is acceptable in quality and form  
for publication on microfilm:

A handwritten signature in dark ink, appearing to be "H. Yang", written above a horizontal line.A handwritten signature in dark ink, appearing to be "H. Yang", written above a horizontal line.A handwritten signature in dark ink, appearing to be "Andrew T. Rogers", written above a horizontal line.

Committee Chair

University of California, Irvine

2001

## **DEDICATION**

To

my parents, brother and sisters  
my wife Fenghuan and  
baby son Kelvin

## TABLE OF CONTENTS

|                                                                      | Page |
|----------------------------------------------------------------------|------|
| LIST OF FIGURES                                                      | vii  |
| LIST OF TABLES                                                       | ix   |
| ACKNOWLEDGEMENTS                                                     | x    |
| CURRICULUM VITAE                                                     | xi   |
| ABSTRACT OF THE DISSERTATION                                         | xiii |
| <br>CHAPTER 1 INTRODUCTION                                           |      |
| 1.1 Motivation                                                       | 1    |
| 1.2 Theoretical Implications and Contributions                       | 4    |
| 1.3 Dissertation Outline                                             | 6    |
| <br>CHAPTER 2 A MYOPIC DETERMINISTIC MODEL                           |      |
| 2.1 Introduction                                                     | 8    |
| 2.2 Formulation                                                      | 12   |
| 2.3 Over-Constrained and Under-constrained Methods                   | 14   |
| 2.4 Time Window Reduction and Partitioning                           | 17   |
| 2.5 A Discretization Scheme with Monotonically non-Increasing Costs  | 20   |
| 2.6 Convergence of the Method                                        | 21   |
| 2.7 Tests                                                            | 27   |
| 2.8 Solution Quality                                                 | 30   |
| 2.9 Tightness of Lower Bound Compared to Other Alternatives          | 32   |
| 2.10 Comparison of Solution Times for Three Lower Bound Alternatives | 34   |

|                                           |                                                                       |    |
|-------------------------------------------|-----------------------------------------------------------------------|----|
| 2.11                                      | Tests of Larger Problems                                              | 35 |
| 2.12                                      | Conclusion                                                            | 37 |
| 2.13                                      | Extensions and Further Research                                       | 38 |
| CHAPTER 3 STOCHASTIC ASSIGNMENT MODELS    |                                                                       |    |
| 3.1                                       | Introduction                                                          | 40 |
| 3.2                                       | The Effect of Stochastic Service Times                                | 42 |
| 3.3                                       | Calculation of the Expected Value for an Assignment                   | 44 |
| 3.3.1                                     | An Evaluation Method                                                  | 45 |
| 3.3.2                                     | Comments                                                              | 51 |
| 3.4                                       | Models with Stochastic Travel Times                                   | 52 |
| 3.4.1                                     | Chance Constrained Models                                             | 52 |
| 3.4.1.1                                   | A Classical Chance Constrained Method                                 | 52 |
| 3.4.1.2                                   | Formulation I and its Properties                                      | 55 |
| 3.4.1.3                                   | Formulation II                                                        | 60 |
| 3.4.2                                     | A Mixed Approximation Model of Stochastic Assignment<br>with Recourse | 62 |
| 3.5                                       | Conclusion                                                            | 64 |
| CHAPTER 4 MODELS WITH TRAILER POSITIONING |                                                                       |    |
| 4.1                                       | Introduction                                                          | 66 |
| 4.2                                       | Related Work                                                          | 70 |
| 4.3                                       | Future Research                                                       | 72 |
| CHAPTER 5 RESEARCH EXTENTIONS             |                                                                       |    |
| 5.1                                       | Introduction                                                          | 73 |

|           |                                                                                                         |     |
|-----------|---------------------------------------------------------------------------------------------------------|-----|
| 5.2       | The Traveling Salesman Problem with Separation Requirements: An Examination of Alternative Formulations |     |
| 5.2.1     | Problem Statement                                                                                       | 74  |
| 5.2.2     | Related Work                                                                                            | 75  |
| 5.2.3     | Integer Programming Formulation                                                                         | 78  |
| 5.2.4     | Some Analytical Properties of the New Formulation                                                       | 85  |
| 5.2.5     | Conclusion                                                                                              | 88  |
| 5.3       | A Cutting Plane Method for Integer Programming Problem with Binary Variables                            |     |
| 5.3.1     | Introduction                                                                                            | 90  |
| 5.3.2     | The Cutting Plane Method                                                                                | 91  |
| 5.3.3     | Analytical Properties                                                                                   | 93  |
| 5.3.4     | Obtaining Deeper Cuts                                                                                   | 95  |
| 5.3.5     | Conclusion                                                                                              | 97  |
| CHAPTER 6 | CONCLUDING REMARKS                                                                                      | 99  |
|           | REFERENCES                                                                                              | 101 |

## LIST OF FIGURES

|                                                                                                      | <b>Page</b> |
|------------------------------------------------------------------------------------------------------|-------------|
| Figure 2.1 Over-constrained network                                                                  | 15          |
| Figure 2.2 Under-constrained network                                                                 | 16          |
| Figure 2.3 Selection of the discretization point                                                     | 20          |
| Figure 2.4 Optimal assignment with unique schedule                                                   | 24          |
| Figure 2.5 Optimal assignment with a schedule that can be shifted                                    | 24          |
| Figure 2.6 A possible combination of the service times                                               | 25          |
| Figure 2.7 Infeasible assignment from under constrained method                                       | 26          |
| Figure 2.8 The ratio of the cost of lower and upper bound<br>solutions in iterations 1 and 2         | 30          |
| Figure 2.9 Comparison of three lower bounding methods                                                | 33          |
| Figure 2.10 Cumulative solution times for iterations 1-6                                             | 34          |
| Figure 2.11 Ratio of lower to upper bound at iteration one                                           | 36          |
| Figure 2.12 Solution times for the upper bound at first iteration                                    | 37          |
| Figure 3.1 A deterministic assignment in a stochastic setting                                        | 43          |
| Figure 3.2 Assignment to loads with time windows                                                     | 55          |
| Figure 3.3 Illustration of different performance of networks<br>developed under formulation I and II | 61          |
| Figure 4.1 A multi-layered work process                                                              | 67          |
| Figure 5.1 Example network                                                                           | 84          |

|            |                                                                       |    |
|------------|-----------------------------------------------------------------------|----|
| Figure 5.2 | Example to show the tightness of the right hand<br>in Equation (5.12) | 88 |
| Figure 5.3 | Example of cuts added                                                 | 94 |
| Figure 5.4 | A branching procedure to get deeper cut                               | 96 |

## LIST OF TABLES

|                                                                                         | <b>Page</b> |
|-----------------------------------------------------------------------------------------|-------------|
| Table 2.1 Probabilities associated with time windows of varying length                  | 28          |
| Table 2.2 Partitioning width                                                            | 29          |
| Table 2.3 The ratio of the cost of lower bound and upper bound<br>in iterations 1 and 2 | 31          |
| Table 2.4 Number of unsolved problems in iteration 1-6                                  | 23          |
| Table 5.1 Separation constraints                                                        | 83          |
| Table 5.2 Distance matrix                                                               | 84          |

## ACKNOWLEDGEMENTS

First of all, I'd like to thank my advisor, Dr. Amelia C. Regan, for her great support. I am grateful for her patience and her time, her consideration and her instruction. During the time of her supervision in the past three years, I have learned a lot more than ever before about how to conduct scientific research and how to express my ideas in a paper. She shared all my pleasure in my findings. She brought me to meet her former advisor. Whenever we were to meet, she never forgot to ask about my availability. She paid attention to my family more than anyone else did. When I recall this, I suddenly find I have been the luckiest student in the world.

I'd like to express my appreciation to the members in my dissertation committee: R. Jayakrishnan and Wilfred Recker for all their instruction and help, and especially for Dr. Jayakrishnan's initial effort to recruit me to this graduate program. I won't forget Dr. Wilfred Recker, a leader and gentleman at the Institute of Transportation Studies (ITS). In addition, there are so many kind people whom I will miss for a long time. These include, but are not limited to, Frank Haight, Lyn Long, Mike McNally, and Tom Golob.

My research benefited from other members of our group. Among them, Sreeram Jagannathan generated the test problems for my algorithm, and Xiangwen Lu contributed his comments on my work.

The financial support from TRB IDEA program, UCTC, JB Hunt Inc. and the National Science Foundation are gratefully acknowledged.

Finally, I would deliver my gratitude to the graduate students, both Masters and PhDs at ITS for their friendship. Among them I have to mention Elizabeth Geho, my last office mate Seri Oh and Ming-sheng Lee for their help. I really appreciate the opportunity to meet so many great people.

## **CURRICULUM VITAE**

Xiubin Wang

### **EDUCATION**

Ph.D. in Civil Engineering (Transportation)  
University of California, Irvine, 2001

M.E. in Transportation Management Engineering  
Northern Jiaotong University, Beijing, China, 1992

B.E. in Mechanical Engineering  
Northern Jiaotong University, Beijing, China, 1989

### **WORK EXPERIENCE**

Graduate Student Researcher, Institute of Transportation Studies,  
University of California, Irvine, 1997 to 2001

Assistant Research Professor, Northern Jiaotong University  
Beijing, China, 1992 to 1997

### **REFEREED PUBLICATIONS**

Regan, A.C., S. Jagannathan, and X. Wang (2001), Mixed global and local assignment algorithms for quasi dynamic local truckload trucking operations with strict time-windows, *Transportation Research Record*, in press.

Wang, X. and A.C. Regan (2001), Local truckload trucking with time window constraints, *Transportation Research*, part B, in press.

Wang, X. and A.C. Regan (2001), Assignment algorithms for local truckload trucking problems with stochastic service times and time window constraints, accepted for presentation at the 80th meeting of the Transportation Research Board and for Publication in *Transportation Research Record*.

Wang, X and A.C. Regan (2000), The traveling salesman problem with separation requirements: examination of alternative formulations, *Operations Research Letters*, under review.

## WORKING PAPERS

Wang, X and A.C. Regan (2000), A New Cutting Plane Method for Integer Programming Problems with Binary Variables, Report ITS-UCI-WP-00-12, in preparation for submission to the *European Journal of Operations Research*.

Wang, X and A.C. Regan (2000), Alternative Assignment Models for Time Constrained Local Fleet Assignment in which Service and Travel Times are Stochastic, Report ITS-UCI-WP-00-13, in preparation for submission to *Transportation Science*.

Wang, X and F. Wang (2000), A Driver's Utility Function and Its Properties, Report ITS-UCI-WP-00-14.

## **ABSTRACT OF THE DISSERTATION**

**Algorithms and Strategies for Dynamic Carrier Fleet Operations: Applications to Local Trucking Operations**

by  
**Xiubin Wang**

**Doctor of Philosophy in Civil Engineering  
University of California, Irvine, 2001  
Professor Amelia C. Regan, Chair**

We focus our research on a truckload trucking assignment problem aimed specifically at operations supporting the ground movement of intermodal freight within a compact urban area around intermodal facilities. In order to guarantee service, strict time windows are considered in this research. This assignment model has rich practical and theoretical implications.

This assignment problem is investigated in several steps. First, a myopic deterministic version is studied in which travel time, service time and the demands are fixed. A new non-decreasing partitioning scheme to deal with time window constraints for this problem is developed. A feasible option for solving the dynamic assignment problem is to repeatedly apply this deterministic algorithm in a dynamic setting in a rolling horizon framework whenever new information is available. The deterministic algorithm provides a basis for further consideration of stochastic factors including queuing times, handling times and travel times under traffic congestion. Several stochastic models are proposed and discussed. The discussion indicates that direct

adoption of stochastic models aimed at other problems involves great difficulty because of the complex nature of this problem. Therefore, approximation models are preferable. Further, by incorporating additional requirements of trailer repositioning, a more general problem of multi-layered resource allocation is defined. Multi resource allocation problems have wide practical implications in air, rail and maritime carrier fleet operations. The discussion of these models highlights a promising opportunity for future research.

All the methods and ideas motivated by this specific assignment problem can be easily extended to other routing and scheduling problems. As part of this research, we further investigated some NP-hard problems that generally underlie such applications. A special case of TSP problem, titled “the TSP with separation requirement”, is examined and a new formulation is presented. The formulation takes the TSP with precedence constraints and the time dependent TSP as special cases. Additionally, a new general cutting plane method is proposed. It applies, but is not limited to, integer programming problem with binary variables. We believe that this method has some advantages over its counterpart, Gomory’s method. However, further effort is needed to test its performance.

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Motivation**

Trucking plays a significant role in the transportation industry. Trucking moves more of the nation's freight, whether measured by value, tons, and ton-miles, than any other mode. According to the latest survey data available, in 1997 trucking constituted 69% of the total weight of all shipments (BTS, 1999). One can easily argue that making trucking more efficient leads to a more efficient economy.

There is an urgent need for efficiency in the trucking industry. The industry has undergone substantial changes since its deregulation in late 1970's and early 1980's. Deregulation enabled easy entry into the market and forced trucking companies to operate at marginal cost. The trucking industry has therefore faced tougher and tougher internal competition. The reasons are straightforward. If one company does not accept a customer, others will, provided that the revenue can cover the additional operational cost. In addition, information technologies are creating transparent on-line and real-time business activities, which further increases competition. This situation makes trucking companies struggle hard to reduce their operating costs. As a result, research on the whole operational process and on ways to reduce costs and improve operational efficiency is needed by trucking carrier fleet operators. In fact, there has been more and more attention drawn to this specific area from the research community in recent years.

We focus our research on local trucking operations. Local trucking operations represent a significant fraction of goods movements. In 1997 local shipments (less than 100 miles) constituted nearly 67 percent of the weight and 40 percent of the value of goods moved (BTS, 1999).

In addition, local trucking is often the first move in intermodal transportation. Intermodal (truck-rail) transportation continues to grow at a slow but steady pace. There are three main drivers for this growth: 1) enabling technologies; 2) labor shortages in the industry which make it much easier to hire and keep local rather than long haul drivers; and 3) potential social and economical benefits. For this reason, we place our research in the context of intermodal freight transportation.

We specifically focus on algorithms and strategies for the dynamic assignment problem. Without loss of generality, the problem can be described as follows.

Consider a set of loads to be moved within a local area. The local area will typically contain one or more intermodal facilities. Each load has a time window for service. Without loss of generality, we model these as pickup time windows only. After the start of the day, a set of vehicles are scattered throughout the service region. A vehicle can only serve one load at a time. After moving a load it turns to another directly or remains idle. In the problem considered, the size of the fleet is fixed. However, fleet size could just as easily be an endogenous rather than exogenous variable. The assumption here is that the objective is to move as many loads as possible, within their

time windows, at the least cost. Loads not feasibly served by the fixed fleet are assumed to be sub-contracted immediately to other carriers as is the case in typical intermodal operations. As new loads become known, assignments are changed to accommodate the new demands. Of course, assignments could be made in anticipation of future demands. In addition, stochastic factors affect the operation. These include dockside waiting time, queuing time at intermodal facilities, and travel time subject to reoccurring and non-reoccurring traffic congestion.

This assignment problem is very important in practice. Assignments made determine the efficiency of the operation, service quality and hence a company's competitiveness in a market where companies typically survive on profit margins as thin as 1-3 percent. Our objective is to develop implementable algorithms and operating strategies. At a time when companies are operating in an information rich environment, mathematical models provide a basis to make better use of the information at hand. Transportation is part of the larger logistics and supply chain management process. A basic assignment model could be extended to include factors related to reducing inventory cost, developing optimal pricing strategies, satisfying customer demand and improving facility utilization. In addition, a thorough investigation of this problem lends insight into related problems in rail, maritime and air cargo operations.

## **1.2 Theoretical Implications and Contributions**

Among the most intensively investigated problems related to our research are the vehicle routing problem (VRP) and the dynamic vehicle assignment problem (DVA). In the VRP vehicles are typically required to serve a set of nodes on a tour. The VRP can include time window and capacity constraints. Both types of constraints are referred to as resource constraints. An important problem imbedded in VRP is the shortest path problem with resource constraints (SPPRC), which has been proven to be NP-hard. When multiple vehicles are considered and when no capacity constraints are binding, the VRP problem is a multiple traveling salesman problem (m-TSP). The deterministic version of the problem investigated in this dissertation is an m-TSP problem with time window constraints (m-TSPTW). Recent research on vehicle routing problems investigates ways to incorporate stochastic features, including both stochastic travel time and stochastic demand. A brief review of such research is presented in Chapter 3. Typical research on dynamic vehicle allocation includes the deterministic dynamic assignment problem and its stochastic counterpart depending on whether information about current and future demand is deterministic. This problem has been addressed in detail by Powell (1988,1996, for example).

In short, the problem considered here is inherently connected with VRP and DVA problems in terms of the underlying fundamental problems and skills required to obtain solutions. Progress on one problem can shed light on the others. Basic ideas related to the deterministic and stochastic assignment models used in our research, which are

borrowed from other problems, are examples of this point. Similarly, we hope that our research on this problem can make a contribution to other classical problems.

The development of real-time solutions for the local trucking assignment problem presents serious practical challenges. The scale of problems encountered in practice are often several times that of the scale solvable optimally and efficiently. Therefore, different ways to overcome this difficulty have been examined. Instead of solving the problem directly, small-scale sub-problems are selected and solved. The impact of these methods on the long run solution quality in a stochastic setting has been examined by several researchers. Interesting results and conclusion have been drawn, as in Yang, Jaillet and Mahmassani (1999) and Mahmassani, Kim and Jaillet (2001). Related research on this problem is also seen in Regan, Mahmassani and Jaillet (1996, 1999) and Ichoua, Gendreau and Potvin (2000a, 2000b), where strategies such as en route diversion are tested. Though examining the same problem, our primary objective is to develop efficient algorithms that can be incorporated into solution strategies.

Despite the simplicity with which the truckload trucking assignment problem is described, the problem is NP-hard. Therefore it is one of the problems at the core of an area of scientific research, whose applications span computer science, telecommunications and other important scientific areas. Because the problem has an integer formulation, efficient algorithms for integer programming problems could lead to efficient solution of our problem. This justifies our effort towards the development of an efficient solution method for general integer programming problems. In addition, our

problem is a variant of the NP-hard traveling salesman problem (TSP), which underlies many important applications. Therefore, explorations have been made on the TSP problem in our study. In conclusion, our selection of the research on this topic reflects a commitment to a more general class of optimization problems.

Our contribution includes the following:

- The development of a time window discretization scheme for a deterministic assignment model;
- An examination of time window constrained stochastic assignment models;
- A discussion of the problem in which trailer positioning moves are explicitly included;
- The development of a new formulation of the TSP problem and an exploration of its properties;
- The development of a new cutting plane method for the general integer linear programming problem with binary variables.

We hope that our work will have both practical and theoretical implications.

### **1.3 Dissertation Outline**

We investigate the problem in several steps. In Chapter 2, we study the deterministic version of the dynamic assignment problem, which provides a basis for further research.

After examination of various solution techniques, we develop a time window discretization scheme. Chapter 3 examines alternative stochastic assignment models that feature the stochastic service time and time window constraints and provides an analytical discussion of their characteristics. All of these models can be considered stochastic extensions of the model presented in Chapter 2. In Chapter 4, future research on this specific problem is proposed and a multi resource allocation version of the problem is discussed. Finally related research on the TSP problem and a cutting plane method for general integer programming problem with binary variables are presented as our contribution to the more general combinatorial optimization area. Those results are presented in chapter 5. The work on the TSP provides a new formulation for this well studied problem. Analysis shows that this formulation possesses some good properties. The cutting plane method proposed reflects another endeavor. It applies to the general integer programming problem with binary variables. Compared with its counter-part, Gomory's method, as well as the popular branch and bound method, it shows some advantages. Each chapter is relatively independent of the others, and contains its own conclusions. The dissertation ends with brief concluding remarks in Chapter 6. In an effort to more effectively place our work among the published literature and facilitate clarity, rather than provide an overall literature review in a single chapter, we review literature relevant to the topic presented in each chapter within the chapter itself.

## **CHAPTER 2**

### **A MYOPIC DETERMINISTIC MODEL**

#### **2.1 Introduction**

As the first step in the investigation of the stochastic dynamic assignment problem with time window constraints, a deterministic assignment problem is examined in this chapter.

In this problem, we consider,

- Deterministic demands;
- Deterministic travel time, service time and dock-side waiting time;
- Hard time window constraints.

The problem solved here is a myopic version of the truckload vehicle assignment problem with time window constraints. We do not anticipate future demand and simply assign the vehicles to serve as many known loads as possible. Because travel takes place in a compact region, it is not necessary to consider the future locations of vehicles as is typical in the long haul version of this problem (see for example Powell, 1988). Of course, local operations are in fact somewhat dynamic in nature. A fraction of loads to be moved in a given day become known only a short time before service must take place, trailer repositioning moves are added to the system as the day progresses and loads must sometimes be reassigned due to traffic, dock-side, and intermodal facility delays. The assumption we make is that the assignment problem will be re-solved several times as the

day progresses and more information becomes known.

If we treat each loaded trip as a node, the problem may be viewed as an asymmetric multiple traveling salesman problem with time window constraints (m-TSPTW). A major difficulty to solve the m-TSPTW problem arises from the time window constraints. There are generally three classes of approximation methods used to deal with the time windows. One explicitly considers the time window constraints in the construction of routes. This class of methods includes Dantzig-Wolf decomposition, which is used to decompose the coverage constraints, Lagrangian relaxation, in which the coverage constraints are relaxed, and state space relaxation in which the feasible space of a dynamic programming algorithm is reduced (see for example Kolen Rinnooy Kan and Trienekens, 1987). Both Dantzig-Wolf decomposition and Lagrangian relaxation lead to shortest path sub-problems with time window constraints, which has been shown to be NP-hard (Dror, 1994).

The second class of methods is the relaxation of the time window constraints. Network relaxation methods solve a network problem after relaxing the time window constraints and then partition the windows according to the last infeasible solution. According to Desrosiers *et al* (1986), this method is inferior to the Dantzig-Wolf decomposition method. Lagrangian relaxation of time window constraints is reported to generate results worse than those keeping the time window in the sub-problems (Desrosiers, Sauve and Soumis, 1983).

A third class of methods to deal with the time window constraints is to discretize them. The idea is to replace continuous time constraints and individual assignment variables with a bundle of assignment variables, each corresponding to a different point in time. An early example of this approach is seen in the work by Appelgren (1969, 1971) in which a ship scheduling problem is solved. A paper by Levin (1971) uses the same strategy to generate flight assignments in which each move has a set of alternate service times. The method described in this paper is inspired by the work of Appelgren and Levin but differs in several ways. In the work by Appelgren the service time windows are not actually continuous variables. Shipments must begin on exactly one day and travel times are naturally expressed as integer multiples of days. The paper by Levin introduces the notion of "bundles" of flow variables. However, no attempt is made to address the issue of how many flow variables the bundles should contain or to examine the trade-offs between coarser and finer discretization.

A more recent application of time window discretization can be seen in Swersey and Ballard (1984), where a school bus scheduling problem is solved using a time window discretization method to minimize fleet size. Graham and Nuttle (1986) compare the performance of a time window discretization method against two heuristics for solving the school bus scheduling problem and found that it had good results. The main complaint about the method was that due to computational issues of the time an LP relaxation of the problem was solved and that when this did not have an integer solution that manually adjusting the non-integer variables could be difficult.

Discretization methods have been rarely used in recent years. The likely reason for this is that the method results in an exponentially expanded network. However, recent advances in computing have made this method more attractive than in the past. In addition, we show later that the relatively small number loads assigned to each vehicle at any given time makes this problem well suited to this method.

The contributions of this chapter are the following:

We develop an iterative method for solving m-TSPTW problems using time window discretization. At each iteration we generate and solve an over constrained version of the problem and an under constrained version. The over constrained problem provides us with a feasible solution and an upper bound on the cost of the optimal solution. The under constrained problem provides us with a lower bound on the cost of the optimal solution. As far as we know, no other researchers have provided such a bound.

We develop and implement a scheme in which the solution is guaranteed to be non-increasing in subsequent iterations.

The organization of this chapter is as follows: first we introduce the formulation, then we then present the over-constrained and under-constrained problems formulations. Next, we introduce the time window partitioning method in which non-increasing costs are guaranteed. Finally we present some empirical results.

## Notation

Let:

|            |   |                                                                                                                                                                                                                                                         |
|------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $N$        | = | the set of nodes for loads,                                                                                                                                                                                                                             |
| $K$        | = | the set of vehicles,                                                                                                                                                                                                                                    |
| $o_i$      | = | the starting node for vehicle $i$ ,                                                                                                                                                                                                                     |
| $a_i, b_i$ | = | the beginning and end of the time window for load $i$ .                                                                                                                                                                                                 |
| $T_i$      | = | the service time for load $i$ ,                                                                                                                                                                                                                         |
| $t_{ij}$   | = | the time needed to service load $i$ and then travel to the pickup location of load $j$ (the handling time at load $i$ , the loaded travel time for load $i$ and the empty travel time between the destination of load $i$ and the origin of load $j$ ), |
| $c_{ij}$   | = | the cost of travel from the destination point of load $i$ to the origin of load $j$ ,                                                                                                                                                                   |
| $M$        | = | an infinitely large constant.                                                                                                                                                                                                                           |

The flow variables are present in the problem if a feasible assignment of a vehicle from its starting location and each load and between loads is possible. The flow variable  $x_{i,j}$  is equal to 1 when there is an assignment in which load  $j$  is served by the vehicle departing node  $i$ ; it is equal to 0, otherwise.

## 2.2 Formulation

$$\text{obj min } \sum_{i \in N + \{O_k | k \in K\}} \sum_{j \in N \setminus \{i\}} (-M + c_{i,j}) x_{i,j} \quad (2.0a)$$

$$\sum_{j \in N} x_{O_i,j} \leq 1 \quad \forall i \in K \quad (2.1a)$$

$$\sum_{i \in N + \{O_k | k \in K\}} x_{i,j} \leq 1 \quad \forall j \in N \quad (2.2a)$$

$$\sum_{i \in N + \{O_k | k \in K\}} x_{i,j} - \sum_{m \in N} x_{j,m} \geq 0 \quad \forall j \in N \quad (2.3a)$$

$$x_{i,j}(T_i + t_{i,j} - T_j) \leq 0 \quad (2.4a)$$

$$a_i \leq T_i \leq b_i \quad \forall i \in N \quad (2.5a)$$

$$x_{i,j} \text{ is binary; for all } i \in N + \{O_k | k \in K\}, j \in N, i \neq j \quad (2.6a)$$

Constraints (2.1a) require the vehicles to leave each load at most once. Constraints (2.2a) dictate that each load be served at most once. Constraints (2.3a) say that a vehicle departs from a load only if it serves the load first. Constraints (2.4a) enforce the temporal relationship of consecutive loads. Constraints (2.5a) specify the time window constraints. Constraints (2.6a) are the binary constraints.

The objective function is a multi-objective one. The infinitely large value  $M$  is a sufficiently large constant that ensures that the assignment covers as many feasible loads as may be possibly served.

The problem in this chapter is slightly different from the typical m-TSPTW problem. The vehicles are not required to return to depot after each service completion. There is no a priori guarantee that each load could be served. The objective is to serve as

many loads as possible. These differences are in fact trivial. It is easy to show that by adding dummy vehicles, dummy source (sink) node and links of zero cost that this problem may be transformed to the typical m-TSPTW.

### 2.3 Over-Constrained and Under-Constrained Methods

The traditional way to deal with the non-linear time window constraints using integer programming is to linearize them. However, the linearized constraints are very loose because they are not at the facets of the polytope of the convex hull of the feasible solutions (Langevin, Soumis and Desrosiers, 1990). In the method we present here the time constraints are taken into account in a pre-processing step in which two versions of the problem are constructed. The first is over constrained and the second is under constrained.

The flow variables  $x_{i,j}$  correspond to links from vehicles to loads, and those between loads. Possible links are determined by the time window constraints associated with each load. If the time window is a single time point, then the problem is reduced to a fixed schedule problem (Desrosiers *et al*, 1995). This kind of problem has a clear and exact network representation and can be solved more efficiently.

Suppose we consider only the end points of the time window of both the first and the second load when we set up links between two loads for the network; then we obtain a network that ignores some possible links. We refer to this method as the over

constrained method. Suppose we only consider the starting point of the time window of the first load and the end point of the time window of the second load when we set up the link between two loads; then we end up with a network that includes some infeasible links. We refer to this method as the under constrained method. The over constrained method leads to a network from which we obtain a feasible solution while the under constrained method provides a lower bound for the solution.

Figures 2.1 and 2.2 show the feasible links excluded and infeasible links included in the two methods.

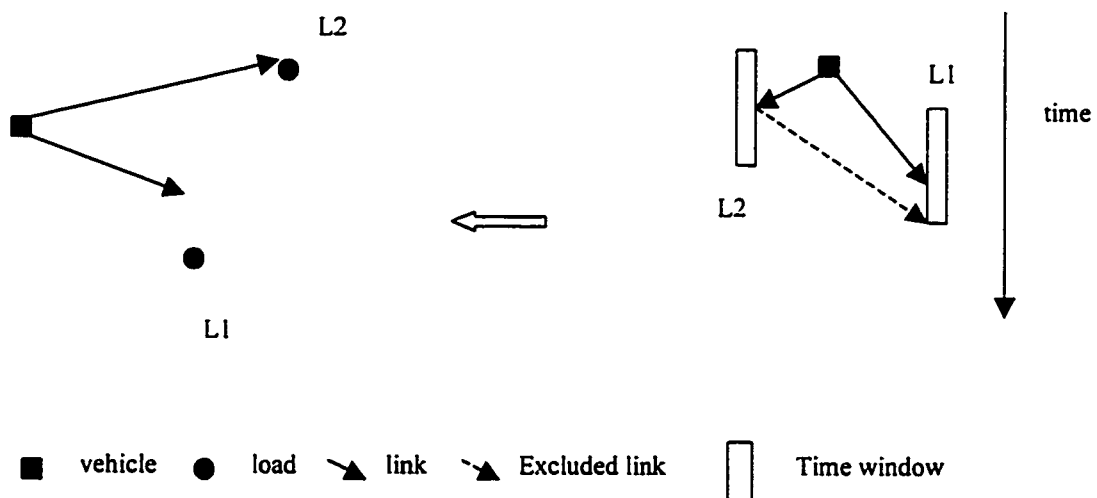


Figure 2.1) Over-constrained network

It is not possible to reach load 1 after leaving at the latest time in the time window for load 2 or to reach load 2 after leaving at the latest point in the time window for load 1. However, as is shown above, it is possible to serve load 1 after serving load 2 in the early part of the time window for load 2.

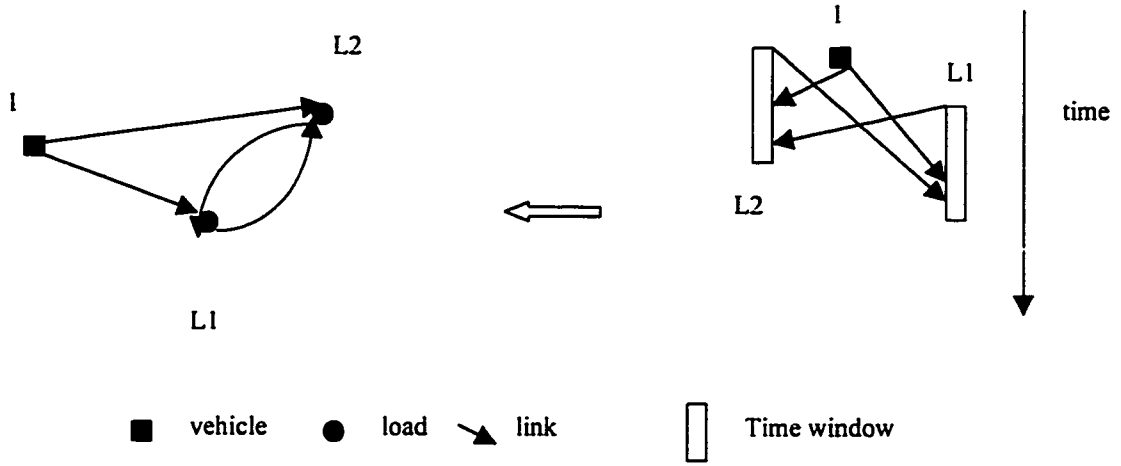


Figure 2.2) Under-constrained network

It may be observed that the under constrained network might permit infeasible links. If the vehicle leaves load 1 at the end of its time window it will have no way to reach load 2 within its time window. In fact, this network may contain cycles.

In the over constrained method we replace constraints (2.4a) and (2.5a) with the following:

$$x_{i,j}(b_i + t_{i,j} - b_j) \leq 0 \quad (2.7a)$$

In the under constrained method we replace constraints (2.4a) and (2.5a) with the following:

$$x_{i,j}(a_i + t_{i,j} - b_j) \leq 0 \quad (2.8a)$$

Constraints (2.7a) define the links for the network of over constrained method. Constraints (2.8a) define links of network from under constrained method. If the coverage constraints (2.1a) are relaxed, the formulation from the over-constrained method is a network flow problem on an a-cyclic network. There are very efficient algorithms to solve such problems. However the network generated using the under constrained method is likely to contain cycles as can be seen in figure 2.2.

If we use the formulation from the under-constrained method, there is some infeasible space included in the solution space. While for over-constrained problem, there is some feasible space excluded. As a result, the optimal solution to the linear relaxation of the formulation by the under constrained method  $Z_l^{LP}$ , the integer solution to the under constrained method  $Z_l^{IP}$ , over constrained method  $Z_m^{IP}$  and the global optimal integer solution  $Z_o^{IP}$  can be placed in the following order:

$$Z_l^{LP} \leq Z_l^{IP} \leq Z_o^{IP} \leq Z_m^{IP}$$

## 2.4 Time Window Reduction and Partitioning

In general, the bigger the time windows, the bigger the gap between  $Z_l^{IP}$  and  $Z_o^{IP}$ , as well as between  $Z_o^{IP}$  and  $Z_m^{IP}$ . Sometimes the gap between the two methods is so large that we cannot determine if we have reached an acceptable solution with respect to the optimal value.

The partitioning method is based on the observation that if the time windows are smaller, the gap between  $Z_m^{IP}$  and  $Z_l^{IP}$  is reduced. To do this, the original window is partitioned into several parts. Each part is considered as a sub-load. At most one of the sub-loads of any load may be served. The vehicle that leaves a sub-load must have entered to the same sub-load. In this way, the number of feasible links excluded by the over constrained method is reduced; similarly, the number of infeasible links included in the under constrained method is also reduced.

The problem is how to select the width to partition the time windows because smaller widths lead to much larger problems. The iterative solution method in which an upper and lower bound is obtained at each iteration allows us to begin by solving problems of reasonable size. If the ratio between lower bound and upper bound is unacceptable then the width selected is too large. In that case we select a smaller partition and solve the problem again. In the tests problems presented, a series of widths is arbitrarily selected to start with two hours and to end with 0.1 hours. The way to partition is as follows. Suppose the pre-selected width for partitioning is  $d$ , and that the load has a window  $(a_i, b_i)$ . First, determine the number of sub-loads for this iteration by taking the smallest integer that is greater than  $(b_i - a_i)$  divided by  $d$ . That is  $\text{ceil}[(b_i - a_i)/d]$  where  $\text{ceil}(x)$  is the ceiling function that finds the smallest integer greater than  $x$ . Then partition the window into this many parts evenly.

This method is further improved by employing time window reduction methods commonly used for preprocessing VRPTW problems. There may be some part of the

time window, which is of no use to the assignment since no vehicle is able to reach the load within that time. That part of the window is eliminated in order to reduce the size of the problem. This method is described in Desrochers, Desrosiers and Solomon (1992). From here on, we refer to reduced time windows when we mention them. Then the formulation after window partitioning can be modified as follows:

$$\text{obj Min } \sum_{\substack{i \in N + \{O_k | k \in K\} \\ \delta(i) = \delta(j)}} \sum_{j \in \omega \setminus \{i\}} (-M + C_{ij}) x_{i,j} \quad (2.0b)$$

$$\sum_{\substack{j \in \omega \\ \delta(j) = \delta(i)}} x_{i,j} \leq 1 \quad \forall i \in \{O_k | k \in K\} + \omega \quad (2.1b)$$

$$\sum_{\substack{i \in \omega + \{O_k | k \in K\} \\ \delta(i) = \delta(j)}} x_{i,j} \leq 1 \quad \forall j \in \omega \quad (2.2b)$$

$$\sum_{\{j | \delta(j) = \xi\}} \sum_{\substack{i \in \omega + \{O_k | k \in K\} \\ \delta(i) = \delta(j)}} x_{i,j} \leq 1 \quad \forall \xi \in N \quad (2.2'b)$$

$$\sum_{\substack{i \in \omega + \{O_k | k \in K\} \\ \delta(i) = \delta(j)}} x_{i,j} - \sum_{\substack{m \in \omega \\ \delta(m) = \delta(j)}} x_{j,m} \geq 0 \quad \forall j \in \omega \quad (2.3b)$$

$$x_{i,j} (T_i + t_{i,j} - T_j) \leq 0 \quad \forall (i,j) \in A, \delta(i) \neq \delta(j) \quad (2.4b)$$

$$a_i \leq T_i \leq b_i \quad \forall i \in \omega \quad (2.5b)$$

$$x_{i,j} \text{ is binary } \quad \forall i \in \omega + \{O_k | k \in K\}, j \in N, i \neq j \quad (2.6b)$$

$\delta(i)$  denotes the load associated with sub-load  $i$ .  $\omega$  is the set of all sub-loads.  $O_k$  is the node for vehicle  $k$ .  $K$  is the set of vehicles. Constraints (2.2'b) stipulate that at most one sub-load be served for each load.  $a_i$  and  $b_i$  represent the beginning and end of the time window of the sub-load  $i$ . In the same way, the over constrained and under constrained methods define the links between sub-loads. After constraints (2.4b) and (2.5b) are replaced with definite links, the formulation possesses a structure that has a network flow

sub-problem after decomposing or relaxing the constraints (2.1b), (2.2b) and (2.3b). In this paper, a branch and bound, method is used to solve the problems by using Cplex®.

## 2.5 A Discretization Scheme with Monotonically non-Increasing Costs

Another observation is that a smaller length to partition the window usually but not necessarily leads to a better feasible solution. An example here shows this point.



Figure 2.3) Selection of the discretization point

Figure 2.3 shows that if the time window at the left is partitioned into two parts, there could be the assignment in which the vehicle goes to two loads in a row as shown by arrows in the figure. But if a smaller width is adopted and the longer window is partitioned into three parts the over constrained method would not allow this assignment. The vehicle can only serve one load, which is worse than the solution from a larger partitioning width.

To guarantee that the over constrained method lead to no worse solutions in subsequent iterations, a special scheme is used. First partition the time window into two parts at the optimal service time from last iteration. Then use the pre-selected width to

partition both parts of the window. In addition, we include a special sub-load whose time window is limited to the first time point of the load's time window.

Here we summarize the procedure used:

1. Select a series of widths used for partitioning.
2. Partition the window into two parts at the time points where the service is delivered at last iteration from over constrained method, then select the first unused width in the pre-selected series to partition the two parts; at the first iteration, partition the whole window directly. Add the first time point of the original window as a sub-load.
3. Generate over and under constrained formulations and solve the two formulations.
4. If the ratio between lower bound and upper bound is acceptable, no smaller width can be used, or machine time is run out, stop; otherwise, select the next width to be used and return to step 2.

## **2.6 Convergence of the Method**

In this section we analyze the convergence of our method.

### **Proposition I**

Under the partitioning scheme, there is at least one partitioning series that guarantees convergence to the optimal integer solution in the over-constrained method.

## **Proof**

First, we show the solution has finite convergence. Then we show that optimal solution is obtainable through the over constrained method.

Suppose there is an optimal solution to the problem. This solution provides a bound for the feasible solutions in over constrained method. Because the solutions from the over constrained method are guaranteed to be non-decreasing, finite convergence must be guaranteed.

Second, for any problem, there must partitioning width that can guarantee that the optimal solution is included in the over constrained problem. Suppose there are  $n$  loads in the optimal assignment, out of which there are  $m$  loads served at a time point different from the starting or ending points of their windows. For the  $(n-m)$  loads for which the optimal service time is the beginning or ending of their time window, these optimal points are always included in the solution space. Therefore, the partitioning scheme need only consider how to include optimal service time for the other  $m$  loads in the over constrained method. We note each of these  $m$  loads, at any iteration, is split into at most three parts, called divisions. One of these divisions must include the optimal service time while another must include the service time from the best solution found so far. There must exist a partitioning width applicable to all the windows such that each division is in an integer multiple of this width. By partitioning with this width, the optimal service times are guaranteed to be considered for assignment. As a result, the

optimal solution is guaranteed. In fact, this width can be easily obtained. Let us suppose division  $i$  has a length of  $\frac{m_i}{n_i}$ , where  $m_i, n_i$  are positive integers with no common factors and  $i$  is the index of all the divisions. One partition that guarantees the optimal solution at next iteration is  $\frac{1}{\prod n_i}$ .

Therefore, any partitioning series that includes a term  $1/k$  times the value obtained as above, where  $k$  is an integer greater than zero, guarantees that the over constrained method contains the optimal solution. For a special case, if the optimal assignment serves all loads at the ending points of the time windows, any partition leads to optimal solution in the over constrained method. (End of proof.)

Intuitively, we observe that the over constrained method results in loss of available time by equivalently forcing drivers to wait to provide service until the last time point in each sub-load's time window. Therefore, the total sum of lost time affects the optimality of the solution. By reducing the partitioning width to be infinitesimal, the discrete problem approaches the continuous one and this sum is reduced to be nearly zero. Therefore, the feasible solution tends towards the optimal one. However, the optimal solution is not guaranteed by *any* partitioning series, even those approaching infinitesimal widths. Two cases could arise. In the first the optimal service time for each of the loads is unique as shown in figure 2.4. In this case, only a series containing a term as in the proof above guarantees optimal solution. Otherwise, any lost time resulting from under constrained method, regardless how small it is, causes the optimal solution to be

missed. For example, a partitioning series  $\{3^{-n}, n \rightarrow +\infty\}$ , though tends towards an infinitesimal width, but does not guarantee the optimal solution since at every partition in this series, the optimal service time at load 1 is lost.

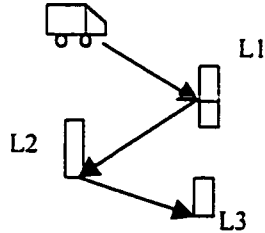


Figure 2.4) Optimal assignment with unique schedule

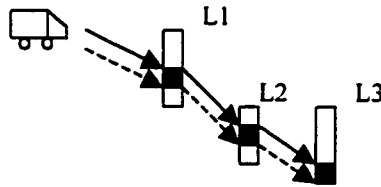


Figure 2.5) Optimal assignment with a schedule that can be shifted

The second case is shown in figure 2.5 where the schedule can be shifted. In this case, all partitioning widths smaller than some certain value lead to optimal solution. In other words, all partitioning series leading to infinitesimal partitioning widths guarantee the optimal solution. In figure 2.5, the dashed arrow lines represent the latest possible schedule that can still keep the optimal assignment. In this case, for any service time that falls in the shadow portion of an upstream load, there is a corresponding spectrum of the window at the downstream load within which every time point can be selected to guarantee an optimal solution. Figure 2.6 shows that once the optimal service time is determined at load 1, any time point in the shaded part of the window for load 2 is

optimal. For the same reason, once the optimal time points are set for both loads 1 and 2, there is also a section of window for load 3 within which all time points are optimal. In this case, if the partitioning width is small enough, the optimal solution is always obtainable in over constrained method. Though this can be shown mathematically, its intuitive proof is much clearer. Therefore we stop here.

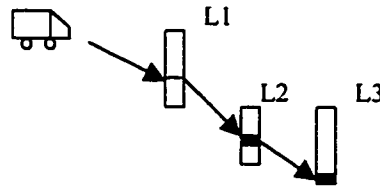


Figure 2.6) A possible combination of the service times

## Proposition II

Solutions from under-constrained method converge to the optimal integer solution if the partitioning width tends towards an infinitesimal value.

## Proof

The under constrained method leads to solutions that might be infeasible. Any infeasible solution can be excluded by adopting a smaller partitioning width. Therefore when the partitioning width tends towards an infinitesimal value, the solution space defined by the under constrained method converges to the solution space of the original problem.

This point can be shown in the example shown in figure 2.7.

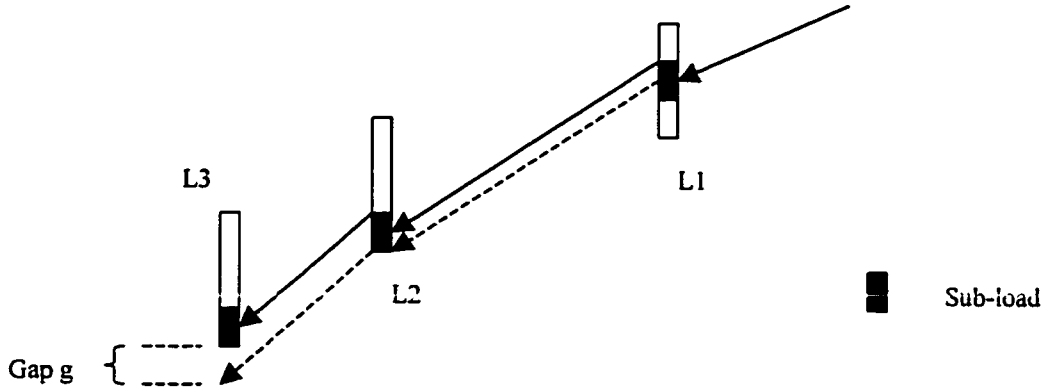


Figure 2.7) Infeasible assignment from under constrained method

In figure 2.7, assignment  $L_1L_2L_3$  is infeasible at load 3, missing the deadline by a gap  $g$ . The assignment is obtained when the three sub-loads in shadow are considered. The assignment is infeasible because when a link is added between two sub-loads in under constrained method, the driver is equivalently empowered to move backward in time dimension by a small value, say  $\theta_i$  (corresponding to load  $i$ ). The infeasible assignment is made only when the sum of the  $\theta_i$  before the infeasible load can make up the difference between the actual arrival time and the latest point of the time window for this same load. It can be observed that this small value  $\theta_i$  must be less than or equal to the width of that sub-load. By reducing the partitioning width,  $\theta_i$  can be made small enough such that  $\sum \theta_i$  would not make up the gap  $g$ . As a result, this assignment will be excluded. (End of proof.)

From the analysis above, it can be seen for a *given* partitioning series, even when the series tends infinitely small, the over constrained method does not guarantee convergence to the optimal solution while under-constrained method does. However, our intuition is that a smaller partitioning width typically contributes significantly to the improvement in optimality in the over constrained method. The reason is that most problems encountered in practice have schedules that can be shifted. In these cases, a sufficiently small partitioning width always leads to an optimal solution. Even in the case where the optimal schedule cannot be shifted, our argument is that smaller width helps to improve solution quality. This is the reason this method works well in practice as indicated by the tests that follows. In addition, most problems encountered in practice have several alternative optimal solutions.

## **2.7 Tests**

Using the GIS package TransCAD, we generated a set of representative problems based on real data. The problem generation package is part of a larger GIS based fleet management simulation model described in Jagannathan (1999) and Regan, Jagannathan and Wang (2000). The loads are generated by selecting randomly from known customers in the service area. Time windows are randomly assigned based on the distribution shown in table 2.1, which roughly corresponds to the time windows associated with loads known at the start of day.

Table 2.1 Probabilities associated with time windows of varying length

| Time window | 7:00-7:30 AM<br>(0.5 hours) | 8:00-9:30AM<br>(1.5 hours) | 8:00AM-12:00PM<br>(4 hours) | 12:00-5:00PM<br>(5 hours) |
|-------------|-----------------------------|----------------------------|-----------------------------|---------------------------|
| Probability | 0.1                         | 0.15                       | 0.35                        | 0.4                       |

For the problems in the test set, we begin with the vehicles at the depot (which in this problem is very near the rail yard) and make all vehicles available for the duration of the day. Vehicles are not required to return to depot after each service. Travel distances correspond to the shortest network travel distance. Travel time is assumed to be 35 miles per hour, reflecting congestion levels in the test region. The average loaded distance is quite short, less than twenty miles long. A handling (typically dock time) of forty minutes is assumed for each load though in an actual operation the handling time would be customer specific. We present results from 30 problems of 20 vehicles and 75 loads. Twenty was selected because it is the typical maximum size of a local sub-fleet handled by a single load manager (dispatcher). In the test, the cost is the empty travel distance. We set the parameter  $M$  to 10000 in the test since it is much larger than even the sum of the total empty travel distance. We then present results from 30 similarly generated problems of 40 vehicles and 150 loads. For these larger problems two observations are made. The first is that these larger problems reach optimality in earlier iterations, on average. The second is that for larger problems only limited iterations are possible because the size of the problems expands too quickly.

The widths to partition the time window at iterations are as follows:

Table 2.2 Partitioning width

| Iteration | 1 | 2    | 3   | 4    | 5    | 6    | 7   | 8   | 9   | 10  | 11  | 12  |
|-----------|---|------|-----|------|------|------|-----|-----|-----|-----|-----|-----|
| Width     | 2 | 1.75 | 1.5 | 1.25 | 1.00 | 0.75 | 0.6 | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| (hours)   |   |      |     |      |      |      |     |     |     |     |     |     |

We solve the problems using CPLEX<sup>®</sup> version 5.0 with no special modifications. The later versions of this software (6.0 and 6.5) are known to have faster solution times, particularly for Mixed Integer Programming (MIP) problems. Therefore, exact times presented here should be considered relative solution times. In addition, the solution times could be further improved by the addition of a more sophisticated decomposition or relaxation method. One of the goals of this research was to develop a method simple enough to be implemented by engineers with only limited training in algorithm development. This method can be implemented by anyone with limited programming skills and an understanding of the workings of commercial optimization software.

All tests are run on a desktop computer, a 400 MHZ Pentium II PC with 256 MB RAM. Three aspects of the method are tested. The optimality of the upper bound solution (feasible solution) in terms of the ratio of the cost of the lower bound to upper bound; the tightness of the lower bound as opposed to two other alternatives and their corresponding machine time.

## 2.8 Solution Quality

The ratio of the cost of the solutions associated with the under and over constrained formulations after the first two iterations are provided in table 2.3 and figure 2.8. It may be observed that even when a width as large as two hours is used to partition the windows the solutions are quite good. Here the cost does not include the term  $M$ .

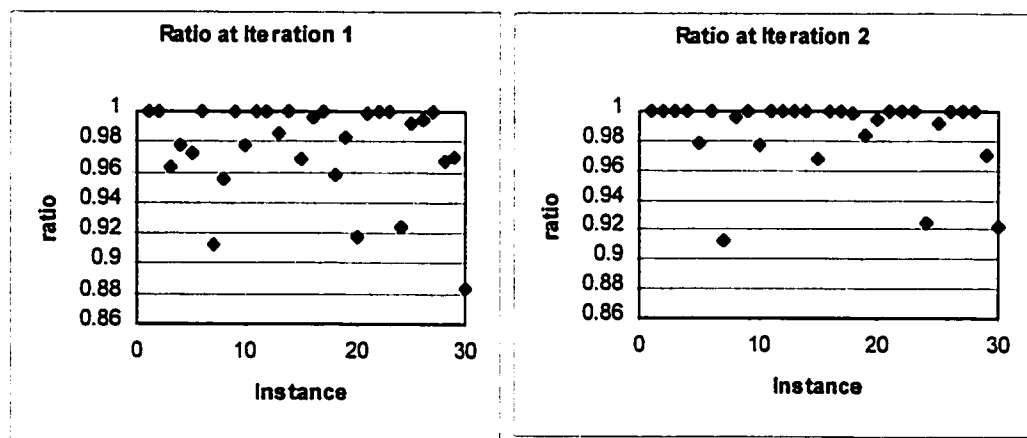


Figure 2.8) The ratio of the cost of lower and upper bound solutions in iterations 1 and 2

Table 2.3 The ratio of the cost of lower and upper bound in iterations 1 and 2

| Problem Instance | Iteration 1 | Iteration 2 |
|------------------|-------------|-------------|
| 1                | 1.0         | NA          |
| 2                | 1.0         | NA          |
| 3                | 0.96401     | 1.0         |
| 4                | 0.97838     | 1.0         |
| 5                | 0.97305     | 0.9782      |
| 6                | 1.0         | NA          |
| 7                | 0.91203     | 0.91203     |
| 8                | 0.95538     | 0.99594     |
| 9                | 1.0         | NA          |
| 10               | 0.97717     | 0.97717     |
| 11               | 1.0         | NA          |
| 12               | 1.0         | NA          |
| 13               | 0.98536     | 1.0         |
| 14               | 1.0         | NA          |
| 15               | 0.96806     | 0.96806     |
| 16               | 0.99618     | 1.0         |
| 17               | 1.0         | NA          |
| 18               | 0.95863     | 0.9993      |
| 19               | 0.98326     | 0.98358     |
| 20               | 0.91734     | 0.99524     |
| 21               | 0.99821     | 1.0         |
| 22               | 1.0         | NA          |
| 23               | 1.0         | NA          |
| 24               | 0.92423     | 0.92423     |
| 25               | 0.99227     | 0.99227     |
| 26               | 0.99518     | 1.0         |
| 27               | 1.0         | NA          |
| 28               | 0.96672     | 0.96996     |
| 29               | 1.0         | 0.96996     |
| 30               | 0.88405     | 0.92211     |

## 2.9 Tightness of Lower Bound Compared to Other Alternatives

Because the time required to solve for the lower bound at each iteration may be long, two alternative methods are considered for the generation of a lower bound. The first is to solve the LP relaxation of the original formulation (2.0a)-(2.6a). We call this solution L1. The other alternative is to solve the LP relaxation of the under constrained formulation (2.0b)-(2.6b). We call L2. In this formulation the time window constraints are linearized as in Desrosiers, *et al*, (1986). Equations (2.4b) are replaced with equations (2.4'b).

$$x_{i,j}(T_i + t_{i,j} - T_j) \leq 0 \quad \forall (i,j) \in A, \delta(i) \neq \delta(j), \quad (2.4b)$$

$$T_i + t_{i,j} - T_j \leq (1 - x_{i,j})M \quad \forall (i,j) \in A, \delta(i) \neq \delta(j), \quad (2.4'b)$$

Figure 2.9 compares the tightness of each of these bounds for iterations 1-6, presenting only problems that remain unsolved in each iteration. We refer to the under constrained method in this figure as UC. It may be observed that the improvement obtained by using the lower bound from the under constrained, and more computationally expensive method is limited, on average. However this does not exclude the possibility that in some instances the under constrained method provides a much tighter lower bound than its LP relaxation. In one of the thirty problems examined the gap between the two lower bounds after two iterations was more than 5%. This suggests that if this method is applied in off-line situation where the solution times are allowed to be fairly long, that implementing the tighter lower bound has considerable benefit. However, if the method

is used in an on-line situation then the bound associated with the LP relaxation of the under constrained formulation should be used because the limited improvement in the lower bound is not worth the corresponding increase in solution time. Three of the thirty problem instances are not included in the table because we were unable to obtain the optimal solution for these problems despite obtaining tight bounds for these solutions.

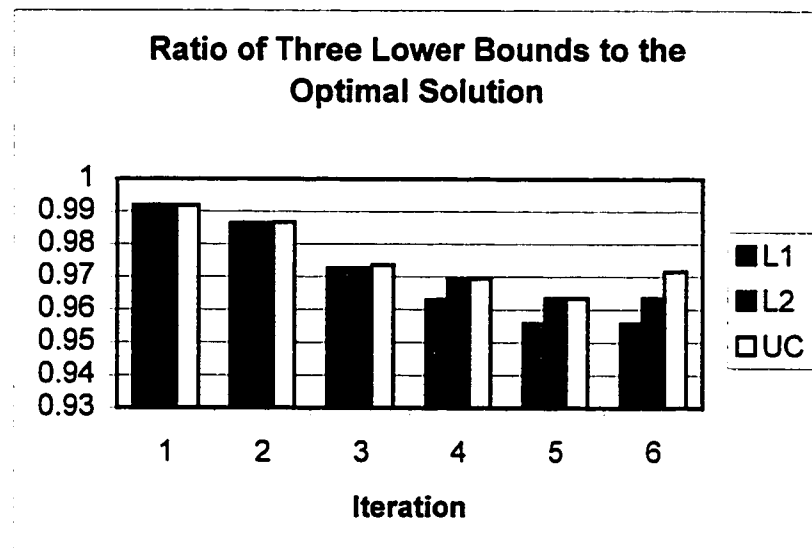


Figure 2.9) Comparison of three lower bounding methods

Table 2.4 Number of unsolved problems in iterations 1-6

|                 |    |      |     |      |   |      |
|-----------------|----|------|-----|------|---|------|
| Iteration       | 1  | 2    | 3   | 4    | 5 | 6    |
| Width           | 2  | 1.75 | 1.5 | 1.25 | 1 | 0.75 |
| Number Unsolved | 30 | 19   | 10  | 7    | 5 | 5    |

## 2.10 Comparison of Solution Times for Three Lower Bound Alternatives

In the operation of interest here, solutions must be obtained rapidly. Figure 2.10 presents the cumulative solution times for iterations 1-6 for the three methods, L1, L2 and UC (under constrained IP). The cumulative time includes the solution time for the under constrained problems and over constrained problems.

It is worth mentioning here that it is very difficult to solve the original integer programming formulation of this problem directly. For the 30 problems examined, none could be solved using a standard branch and bound method directly. We contrast this with the fact that all of them can be solved in under constrained formulation by branch and bound method within a couple of minutes.

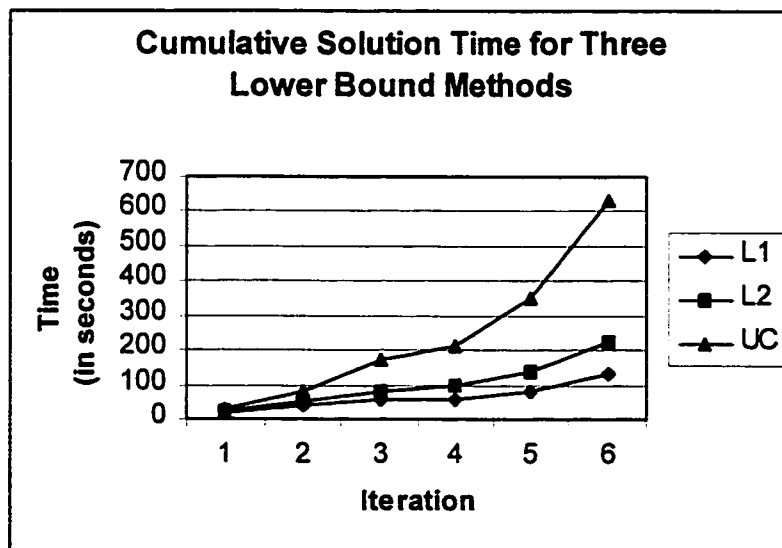


Figure 2.10) Cumulative solution times for iterations 1-6

## 2.11 Tests of Larger Problems

A natural question is how this method performs on larger problems. According to the same distribution as in the 30 problems just tested, we generated thirty more instances of 40 vehicles and 150 loads. The solution times for these problems were relatively long. In the system this method is intended to support solution times must be no more than 30 minutes. Less than five minutes is preferable because dispatchers will sometimes disagree with a solution found (due to information available to them but not available to the decision support system) and will need to remove some loads for assignment outside the optimization framework. For that reason we examine only the solution that may be obtained for the larger problems within this time constraint. We present the results obtained after the first iteration.

Figure 2.11 shows the ratio between the lower bound and upper bounds for the larger problems. It may be observed that the discretization method appears to work better on larger problems than on small ones. Even with a width of two hours, many of the problems are solved to optimality and most are solved to within a gap of 2%.

The average time required to obtain the first feasible solution (the solution of the over constrained formulation) for the thirty problems was 438.8 seconds. The average time required to generate the lower bound using the under constrained method was 255.8 seconds. The under constrained method takes less time because the solution obtained

from the over constrained formulation is used as a cut off point for the branch and bound algorithm. The time required to solve the LP relaxation L1, in order to find a corresponding lower bound is less than one minute. For three of the thirty problems we were unable to solve for the lower bound from the under constrained formulation. However, for these problems the cost of the LP relaxation L1 was exactly equal to the cost of the feasible solution (the optimal solution was found). For the 27 problems for which we were able to obtain the bound associated with the first iteration of the under constrained problem, the value of these lower bounds were exactly the same. Figures 2.11 and 2.12 show the gap between the upper and lower bounds and also the solution times required to find the first feasible solution for each of the 30 larger problems. Figure 2.12 shows the solution times for over constrained formulation (feasible solution) only.

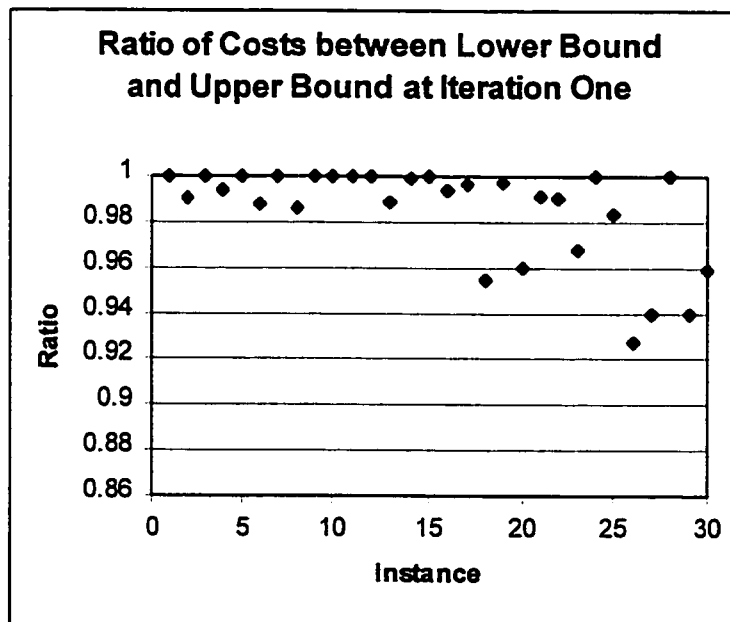


Figure 2.11) Ratio of lower to upper bound at iteration one

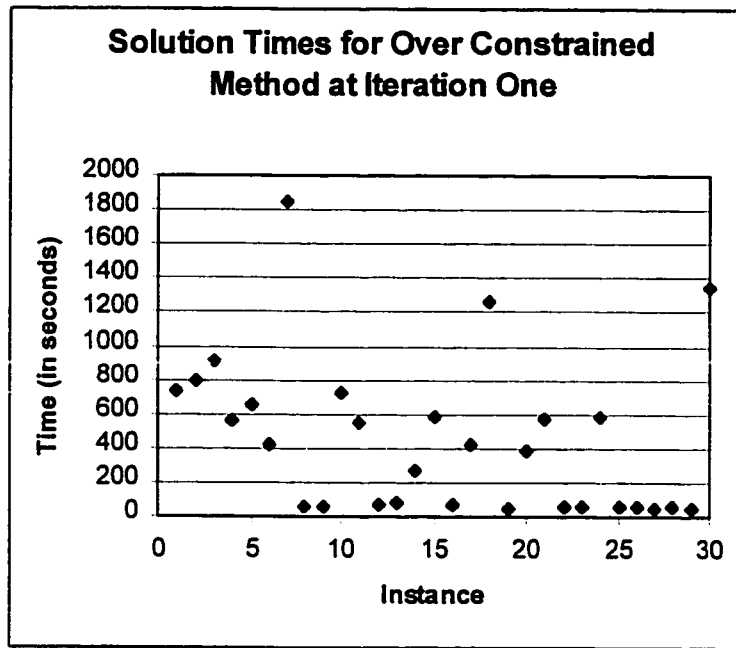


Figure 2.12) Solution times for the upper bound at first iteration

## 2.12 Conclusion

In this chapter, a method to develop vehicle assignments for local truckload operations is presented. Under constrained and over constrained method are presented. The under constrained method is used to evaluate the optimality of the feasible solution obtained using the over constrained method. Over constrained and under constrained method provides a mean to compromise between the optimality and machine time. A cost non-increasing partitioning scheme is developed for implementation in the iterative solution process. The convergence of the over and constrained methods is analyzed. A set of problems based on real data has been examined.

Test problems presented in this paper focus on relatively small, but operationally

realistic problems. Test result shows that smaller partitioning width and larger problems lead to higher level of optimality, but longer machine time. Test results suggest that using what is now a relatively slow commercial optimization package that operational problems of reasonable size can be solved quickly. In addition, the method examined results in sub-problems that are simply network flow problems. Therefore, decomposition techniques could be applied along with time window partitioning method discussed in order to solve larger problems more quickly. Empirical analysis suggests that for these problems, partitioning longer time windows into two hour windows from which a solution will be selected results in good solutions. For the problems of this size, the loss of optimality introduced by discretization is limited as is indicated by the ratio between upper bound and lower bound. For real-time operations, solving the LP relaxation of the under constrained or original formulation seems the most efficient method for obtaining a lower bound. On the other hand, if machine time is not a big concern, under constrained method instead of its LP relaxation is preferable.

### **2.13 Extensions and Further Research**

As is always a possible option, the myopic deterministic model could be applied repeatedly to dynamically adjust the assignment as new loads are called in and new information is available. The myopic optimality does help improve the performance in a long run compared with myopic heuristic methods. In a series of tests where travel time and service time are set to be deterministic, the strategy to repeatedly apply the myopic deterministic model outperforms its heuristic counter part significantly in terms of

average empty travel distance, number of loads served, and like. For a detailed description of the tests, interested readers please refer to Regan, Jagannathan and Wang(2001).

Though anticipation of future demand is a significant feature of dynamic assignment problem, it is not considered in this dissertation. The reason is that demands within a compact region are always accessible instantly since travel time is very short compared with its counterpart in long haul trucking, where anticipation of future demand is necessary for vehicle balance between regions. However, in a region around intermodal facilities where there are traffic congestion and queues at customer locations, the stochasticity of travel time and service time is necessary to address. This is discussed in the Chapter 3.

Finally, while this method was developed with time constrained local truckload trucking operations in mind, it applies to any optimization problem in which the number of time constrained tasks to be assigned to each server at any time is relatively small (as opposed to the typical vehicle routing problem with time windows). Such problems arise in job shop scheduling, for example.

## **CHAPTER 3**

### **STOCHASTIC ASSIGNMENT MODELS**

#### **3.1 Introduction**

In this chapter, a stochastic assignment problem is examined. In this problem, we consider:

- Deterministic demands;
- Stochastic travel time, service time and dock-side waiting time;
- Time window constraints.

The dock side waiting time, the time required for loading and unloading, and the duration of travel times are all stochastic because of reoccurring and non-reoccurring congestion and queues at intermodal facilities and customer locations. Deterministic assignment models, as discussed in Chapter 2, provide a way for companies that have traditionally relied on dispatchers alone to generate assignments to reduce costs and improve service. However, in a highly stochastic operating environment, deterministic assignment models may produce sub-optimal or infeasible assignments. Models which explicitly incorporate stochastic characteristics should provide more robust solutions.

There has been some previous research in the area of vehicle routing and scheduling in which stochastic travel times are considered explicitly. However, little has been done concerning the local truckload trucking problem. In long haul truckload

trucking assignment problems, travel times are relatively long compared to the local truckload trucking assignment in which the service area is confined to a compact region. The variability of the travel time in long haul trucking operations is less of a problem than in local operations because in those operations, once the driver perceives that he or she is behind schedule, there are opportunities to make up for the delay later in the trip. Perhaps this is one of the reasons that typical work on long haul trucking problems considers only the stochasticity associated with the *demand*. Full consideration of stochastic travel times is important for the local truckload trucking problems in which drivers are subject to the urban traffic congestion and long queues at customer locations and intermodal facilities.

To date there has been only one paper dealing with vehicle routing problems with stochastic travel time. In Laporte, Louveaux and Mercure (1992), several models are proposed including a chance constrained model and a stochastic programming model with recourse. Related research addresses vehicle routing problems with stochastic demand (VRPSD). Some examples are Dror, Laporte and Trudeau (1989), Dror, Laporte and Louveaux (1993), and Gendreau, Laporte and Seguin (1995). Good heuristic methods have been developed for VRPSD problems. These are described in Gendreau, Laporte and Seguin (1996). More recently, Yang, Mathur and Ballou (2000) provide a new solution method for the vehicle routing problem with stochastic demand. Two heuristic methods to identify cost effective anticipatory restocking points are developed. There is a rich literature examining truckload trucking problems with stochastic demand including Powell (1988, 1996), and more recently Powell, Snow and Cheung (2000). In

addition, Powell, Towns and Marar (2000) examine the effect of optimal myopic solutions in a stochastic setting in a long haul truckload trucking problem. That research demonstrates that stochasticity in travel time could significantly impact the overall profitability of long haul motor carriers. The same should be true in local operations.

In this chapter, we examine a special case of the local truckload trucking problem in which both stochastic service times (travel, loading, unloading and on-dock waiting times) and time window constraints are considered. We start by examining the effect of the stochasticity on the assignment, then present a method for evaluating the quality of an assignment. Then, several alternative assignment models are proposed. We end with some conclusions and discussion about extensions to this research.

### **3.2 The Effect of Stochastic Service Times**

We can see by example that an assignment generated under deterministic assumptions can be sub-optimal in a stochastic setting. It is assumed that the assignments here are deterministic a priori solutions and that the objective is to serve as many loads as possible while minimizing the cost of the service.

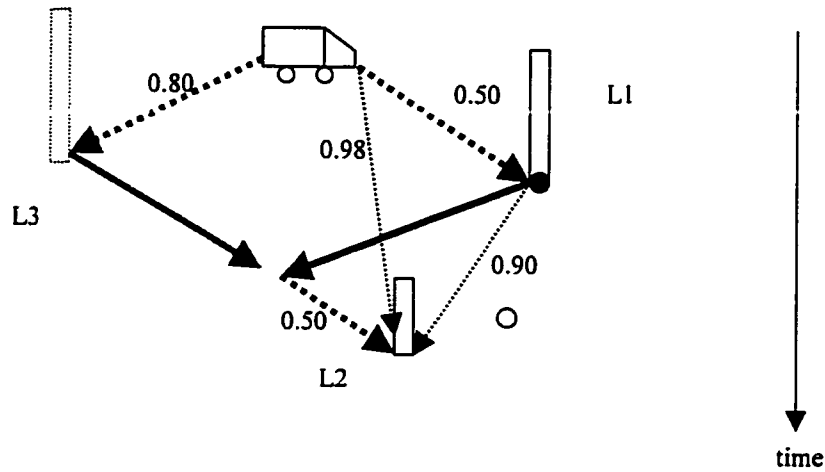


Figure 3.1) A deterministic assignment in a stochastic setting

In the example shown in figure 3.1, the solid lines represent loaded movements while dashed lines represent empty movements. The numbers next to the lines represent the probability that the vehicle will arrive before the end points of the pickup time windows. In the diagram, the distance from the vehicle to load 3 is longer by a very small value than that from the vehicle to load 1. The probability that the vehicle can successfully serve these loads is given in the figure and based on the stochasticity of the travel time in the network. All other characteristics of load 3 are the same as those of load 1. We further assume that the vehicle can either serve load 1 followed by load 2 or load 3 followed by load 2, but not all three loads. The deterministic model that includes all the links in the figure leads to an optimal assignment  $L_1L_2$ , which means the vehicle serves

load 1 followed by load 2. We now calculate the expected number of loads that will be served before their deadlines under this assignment.

Let,

A= event that load 1 is served,

B= event that load 2 is served

Then the expected number of loads served can be obtained in the following way.

$$\begin{aligned} E[L] &= P(A) + P(B) \\ &= P(A) + P(B | A)P(A) + P(B | \bar{A})P(\bar{A}) \\ &= 0.5 + 0.5 * 0.5 + 0.5 * 0.90 = 1.2 \end{aligned}$$

We can see that the expected value differs significantly from the objective value of the deterministic assignment (1.2 verses 2.0). Let us further consider an alternative assignment  $L_3L_2$  that in a deterministic setting is inferior to the one above. The assignment  $L_3L_2$  has an expected value of 1.44, much higher than the deterministic optimal assignment.

From the example, we can conclude that stochastic factors significantly affect the optimality of the deterministic assignment.

### **3.3 The Calculation of the Expected Value for an Assignment**

In the proceeding section, we described a method to calculate the expected number of loads served for a small problem. In this section, a formal expression of the calculation method is given. First, it is important to reiterate the preconditions imposed on the problem. Given an assignment, a vehicle tries to serve the next load only if the

possibility for it to serve the load is greater than zero. If, due to delays earlier in the schedule, a load cannot be served, the vehicle simply skips that load and goes directly to the next load in the assignment. The operational assumption here is that the load will be picked up by another vehicle.

### 3.3.1 An Evaluation Method

Given an a priori assignment  $L_1 L_2 \dots L_n$ , the vehicle goes to load 1 through  $n$  in numerical order. We let  $O$  represent the set of all possible outcomes and let  $o \in O$  represent an individual outcome. In addition,  $s_o(i) \in \{1, 0\}$  represents the status of a load  $i$ , one for being served, zero otherwise. So, each outcome can be represented by the following:  $s_o(1)s_o(2)s_o(3)\dots s_o(n)$ . For example, 1 0 1 represents an outcome in which load one is served, load 2 is missed, and starting from the origin of load 2, the vehicle goes and serves load 3.

Let,

$p_o$  = the probability associated with a given outcome  $o \in O$ ,

$T_j$  = the arrival time of vehicle at load  $j$ .

Then, the expected number of loads served is given by:

$$E[L] = \sum_{o \in O} \left[ \sum_{j=1}^n s_o(j) \right] p_o \quad (3.1)$$

Where

$$p_o = P\left(\bigcap_{j=1}^n (T_j \leq b_j)^{s_o(j)} (T_j \geq b_j)^{1-s_o(j)}\right) \quad (3.2)$$

and in the expression above,  $(T_j \leq b_j)^{s_o(j)} (T_j \geq b_j)^{1-s_o(j)}$  represent whether load  $j$  is served or not. For simplicity,  $\bigcap_{j=1}^n (T_j \leq b_j)^{s_o(j)} (T_j \geq b_j)^{1-s_o(j)}$  is used for the expression of the following,

$$(T_1 \leq b_1)^{s_o(1)} (T_1 \geq b_1)^{1-s_o(1)} \text{ and } (T_2 \leq b_2)^{s_o(2)} (T_2 \geq b_2)^{1-s_o(2)} \dots \text{ and } (T_n \leq b_n)^{s_o(n)} (T_n \geq b_n)^{1-s_o(n)}$$

Further, let,

$c_o(j)$  = the cost of moving empty to the pickup location of load  $j$  in an outcome  $o \in O$ , which varies according to the previous load in the chain and whether that load was served.

$a_i$  and  $b_i$  = the starting and ending points for the time window of load  $i$ .

$r$  = the contribution to revenue associated with each load served.

$t_{j-1,j}^d$  = service time between load  $j-1$  to load  $j$  (total travel time from origin of load  $j-1$  to origin of load  $j$ ) if load  $j-1$  is served.

$t_{j-1,j}^o$  = service time between load  $j-1$  to load  $j$  (total travel time from the origin of load  $j-1$  to the origin of load  $j$ ) if load  $j-1$  is missed.

The expected cost associated with an assignment is given by

$$E[R] = \sum_{o \in O} \left\{ \left( \sum_{j \in I} c_o(j) - r \left( \sum_{j \in I} s_o(j) \right) \right) \left( P \left( \bigcap_{j=1}^n (T_j \leq b_j)^{s_o(j)} (T_j \geq b_j)^{1-s_o(j)} \right) \right) \right\} \quad (3.3)$$

Note that the probability that a load will be served is given by the sum of three mutually exclusive probabilities: i) the probability that the load will be served if the load

immediately preceding it was served at the beginning of its time window; ii) the probability that the load will be served if the load immediately preceding it was served within its time window; and, iii) the probability that the load will be served if the preceding load was missed. This fact will be used in the following equations.

For convenience of expression, let

$$E_o(j) = \bigcap_{i=1}^{j-1} (T_i \leq b_i)^{s_o(i)} (T_i \geq b_i)^{1-s_o(i)}$$

represent the service status for the loads preceding the  $j^{th}$  load in an outcome  $o$ .

This leads to equation (3.4).

$$\begin{aligned} P(T_j \leq b_j, E_o(j)) &= P(T_{j-1} + t_{j-1,j}^d \leq b_j, a_{j-1} \leq T_{j-1} \leq b_{j-1}, E_o(j-1))s_o(j-1) \\ &+ P(a_{j-1} + t_{j-1,j}^d \leq b_j, T_{j-1} \leq a_{j-1}, E_o(j-1))s_o(j-1) \\ &+ P(T_{j-1} + t_{j-1,j}^o \leq b_j, T_{j-1} \geq b_{j-1}, E_o(j-1))(1-s_o(j-1)) \end{aligned} \quad (3.4)$$

For example, if both loads  $j$  and  $(j-1)$  are served in this outcome, the joint probability can be expressed in the following way,

$$\begin{aligned} P(T_j \leq b_j, T_{j-1} \leq b_{j-1}, E_o(j-1)) &= P(T_{j-1} + t_{j-1,j}^d \leq b_j, a_{j-1} \leq T_{j-1} \leq b_{j-1}, E_o(j-1)) \\ &+ P(a_{j-1} + t_{j-1,j}^d \leq b_j, T_{j-1} \leq a_{j-1}, E_o(j-1)) \end{aligned}$$

Now we can explain how to obtain the joint probability associated with a given outcome. The calculation of the joint probability for event  $\bigcap_{j=1}^n (T_j \leq b_j)^{s_o(j)} (T_j \geq b_j)^{1-s_o(j)}$

can be made inductively from the beginning of the assignment. First we explain how to derive the joint probability density function when the distribution of arrival time at the prior load is known. This distribution depends on service status of the proceeding loads in the assignment. Here each load is indexed according to the order the load is served in the assignment. Note that  $T_j = T_{j-1} + t_{j-1,j}$ , while  $t_{j-1,j}$  is the time from origin of load  $j-1$  to origin of load  $j$  (this could be  $t_{j-1,j}^d$  or  $t_{j-1,j}^o$  depending on whether or not load  $j-1$  is served). If it is assumed that  $t_{j-1,j}$  is independent of  $T_{j-1}$ , then joint density function of the two random variables is simply the product of the two. We can easily get the probability for the joint event. Therefore, if we have the probability distribution function for  $T_{j-1}$ , and its joint distribution with  $t_{j-1,j}$ , we can obtain the probability for the event,  $(T_j \leq t) \cap E_o(j)$ . Using this result, we can in turn get the probability for the event,  $(T_{j+1} \leq t) \cap E_o(j+1)$ . So, beginning with the first load in an outcome of the assignment, we can gradually get the probability for the event of the first two loads, then the first three loads, and so on, until all the loads are included.

More specifically, given the distribution of the arrival time of the vehicle at load one, we can obtain the probability distribution of the arrival time at the second load as follows:

a) if load one is served

$$\begin{aligned}
 P(T_2 < t, E_o(2)) &= P(t_{1,2}^d + a_1 < t, T_1 < a_1) + P(t_{1,2}^d + T_1 < t, a_1 < T_1 < b_1) \\
 &= \int_{-\infty}^{a_1} \int_{-\infty}^{-a_1} f_{t_{1,2}^d, T_1} dt_{1,2}^d dT_1 + \int_{a_1}^{b_1} \int_{-\infty}^{t-T_1} f_{t_{1,2}^d, T_1} dt_{1,2}^d dT_1
 \end{aligned} \tag{3.5a}$$

b) if load one is missed

$$P(T_2 < t, E_o(2)) = P(t_{1,2}^o + T_1 < t, T_1 > b_1) = \int_{b_1}^{+\infty} \int_{-\infty}^{t-T_1} f_{t_{1,2}^o, T_1} dt_{1,2}^o dT_1 \quad (3.5b)$$

where  $f_{t_{1,2}^o, T_1}$ ,  $f_{t_{1,2}^d, T_1}$  are the joint distribution of  $t_{1,2}^o$  and  $T_1$  and that of  $t_{1,2}^d$  and  $T_1$ .

Let us see how to recursively use the above equation to derive the probability for the event  $(T_j \leq t) \cap E_o(j)$ .

From equation (3.5), we have

$$\begin{aligned} f_{(T_2, E_o(2))} &= \partial P(T_2 < t, E_o(2)) / \partial t \\ &= \partial [P(T_2 < t | E_o(2)) * P_o(E_o(2))] / \partial t \\ &= f_{(T_2 | E_o(2))} P_o(E_o(2)) \end{aligned}$$

Further, if we assume that  $t_{2,3}$  is independent of  $T_2$ , the joint probability density function is just the product of the two density functions as shown below,

$$f_{t_{2,3}, T_2 | E_o(2)} = f_{t_{2,3} | E_o(2)} * f_{T_2 | E_o(2)} = f_{t_{2,3}} * f_{T_2 | E_o(2)}$$

We can easily obtain the probability  $P(T_3 < t, E_o(3))$ , in the same way as in Equation (3.5), where  $T_3 = t_{2,3} + T_2$ . This can be shown below.

For example, suppose Load 2 is served.

According to Equation (3.5), we have

$$\begin{aligned}
P(E_o(3)) &= P(T_2 < b_2, E_o(2)) \\
P(T_3 < t, E_o(3)) &= P(t_{2,3}^d + a_2 < t, \underline{T_2 < a_1}, E_o(2)) + P(t_{2,3}^d + T_2 < t, \underline{a_2 < T_2 < b_2}, E_o(2)) \\
&= [\int_{-\infty}^{a_1} \int_{-\infty}^{-a_1} f_{t_{2,3}^d, T_2 | E_o(2)} dt_{2,3}^d dT_2 + \int_{a_1}^{b_1} \int_{-\infty}^{-T_1} f_{t_{2,3}^d, T_2 | E_o(2)} dt_{2,3}^d dT_2] * P_o(E_o(2)) \\
&= \int_{-\infty}^{a_1} \int_{-\infty}^{-a_1} (f_{t_{2,3}^d | E_o(2)} f_{T_2 | E_o(2)} P_o(E_o(2))) dt_{2,3}^d dT_2 + \int_{a_1}^{b_1} \int_{-\infty}^{-T_1} (f_{t_{2,3}^d | E_o(2)} f_{T_2 | E_o(2)} P_o(E_o(2))) dt_{2,3}^d dT_2 \\
&= \int_{-\infty}^{a_1} \int_{-\infty}^{-a_1} f_{t_{2,3}^d} f_{T_2, E_o(2)} dt_{2,3}^d dT_2 + \int_{a_1}^{b_1} \int_{-\infty}^{-T_1} f_{t_{2,3}^d} f_{T_2, E_o(2)} dt_{2,3}^d dT_2
\end{aligned}$$

By repeating this process, we can get the probability  $P_o$  associated with a given outcome of the assignment. It is not necessary to obtain the density function  $f_{t_{2,3}, T_2 | E_o(2)}$  explicitly. The function  $f_{t_{2,3}, T_2, E_o(2)}$  can be used instead, as it is the same to all  $j$  in the assignment.

### Observation

The complexity of calculation of the expectation for an a priori assignment is in the order of  $O(2^n)$ .

### Argument:

Given an a priori assignment, the total number of outcomes sum to  $2^n$ , where  $n$  is the number of loads assigned. This number is simply equal to  $\sum_{i=0}^n \binom{n}{i}$ , where  $i$  is the number of loads served in an outcome and  $\binom{n}{i}$  is the number of outcomes in which the same

number of loads are served. Each outcome has an associated probability and specific contribution to the expectation. Therefore, each individual instance must be calculated. (End of informal argument)

### 3.3.2 Comments

In the case of soft time windows which permit service earlier or later than the time window specifies (we call this a *fully soft window*), the evaluation method is much easier.

We only need to obtain the probability distribution of  $\sum_{j=0}^m t_{j-1,j}$  for each of the loads.

Then the probability for each load to be served within its time window can be calculated by integration. In this case, the complexity of evaluation is in the order of  $O(n)$ . However, if earlier service is forbidden while late service is allowed (we call this a *semi-soft window*), the complexity of evaluation should be still in the order of  $2^n$ .

This chapter examines only the case with strict time windows and presents assignment models for this case. If time windows are semi-soft or fully soft, as is often the case in practice, different assignment models are needed. Those models are beyond the discussion of this chapter in which we simply present preliminary analysis of this challenging problem.

### **3.4 Models with Stochastic Travel Times**

In Laporte, Louveaux and Mercure (1992), a chance constrained formulation for the vehicle routing problem with stochastic travel times is presented. In that research, the primary objective is to keep the working time of the vehicle within a preset time limit. In ours, the primary objective is to serve as many loads as possible. In the sections to follow, we present several alternative models for the truckload trucking problem with stochastic travel times and analyze their properties.

#### **3.4.1 Chance Constrained Models**

In this section, three models are presented and some of their properties are analyzed.

##### **3.4.1.1 A Classical Chance Constrained Method**

As a counter part to the formulation of Laporte et al (1992), we provide a chance constrained formulation for the truckload trucking assignment problem with stochastic travel time as follows.

$$Obj \min \sum_{i \in N + \{o_k | k \in K\}} \sum_{j \in N \setminus \{i\}} (-M + c_{ij}) x_{i,j} \quad (3.6)$$

*subject to*

$$\begin{aligned} X &\in S \\ E(L(i)) &\geq |L(i)| (1 - \alpha) \quad \forall i \in K \end{aligned} \quad (3.7)$$

Where  $M$  represents a large value associated with serving a load.  $S$  is the set of routes that satisfy time window constraints as defined by (2.0a)~(2.6a).  $L(i)$  is an assignment for vehicle  $i$ .  $|L(i)|$  is the number of loads in the assignment for vehicle  $i$ .  $E(L(i))$  is the expected number of loads served and  $1 - \alpha$  is the preset service ratio.  $E(L)$  is calculated as in (3.1).  $X = \{x_{i,j}\}$ ,  $x_{i,j}$  is a binary variable.

A solution scheme associated with this formulation involves a relaxation of the chance constraints (3.7), followed by implementation of a branch and bound method for finding an integer solution. The branch and bound method can be coupled with special techniques to deal with time window constraints, as seen, for example in Wang and Regan (2001a). In fact, any method for solving the multiple traveling salesman problem with time window constraints (m-TSPTW) may be used to solve the formulation with relaxed chance constraints. With each integer solution, there is a follow-up check of the chance constraint violation. If the integer solution violates the chance constraints, then the node is fathomed.

The difficulty with this method is as follows. In order to obtain an optimal integer solution, all the integer points in the branch and bound tree must be examined. The comparison of integer solutions involves unavoidably the calculation of the fairly complex expectations of the number of loads served. This calculation makes this method impractical at best. For the same reason, stochastic assignment models with recourse cannot be applied directly without simplification. The evaluation method is too complicated.

In order to avoid such difficulty, new methods are introduced. Before each formulation, we discretize the time windows in order to transform a continuous problem in to a discrete one. We review the discretization scheme used for deterministic assignment problem first. In the discretization scheme, we replace each time window with a set of uniform discrete time points (including the starting and ending points of the window) with each point representing a subload.  $x_{i,j}$  is included in the problem if (subload)  $j$  can be feasibly served by a vehicle leaving sub-load  $i$ . In a deterministic model, this implies the following.

$$x_{i,j}(b_i + t_{i,j} - b_j) \leq 0 \quad (3.8)$$

Here  $b_i$  is the time point for the subload  $i$ .

Based on the discretization method, alternative models are presented in the following section.

### 3.4.1.2 Formulation I and its Properties

The goal of this method is to develop a network incorporating stochastic service time. As described above, we replace the time windows with a set of discrete points and then generate a network connecting feasible sub-loads. In this case, links are only included if their service probability is greater than  $1-\alpha$ , where  $1-\alpha$  should be fairly high, say, 0.95. Then we formulate this problem as in (2.0b)~(2.3b), (2.6b) and the following,

$$x_{i,j}[P(t_{i,j} < b_j - b_i) - (1-\alpha)] \geq 0 \quad (3.9)$$

Where  $t_{i,j}$  is assumed to have a known probability distribution and  $1-\alpha$  is the preset confidence level.

Next we examine the expected number of loads served in the assignment generated under formulation I. Again we assume that assignments made here are the result of a priori solutions. Vehicles follow the assignment without re-optimization.

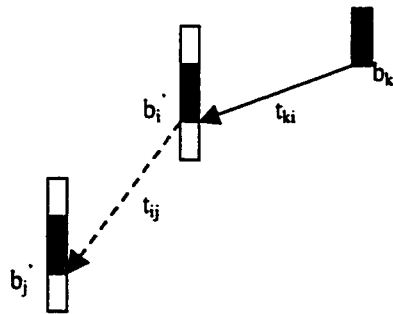


Figure 3.2) Assignment to loads with time windows

### Proposition I

Given any assumption about the probability distribution of the service time, and given a strict time window for pickup, the load at the  $n^{\text{th}}$  position in an assignment chain made by the assignment model given in formulation I, with a confidence level  $1-\alpha$ , can be served with a probability no less than  $(1-\alpha)^n$ .

### Proof

We start with an example. Suppose that the assignment is  $L_1L_2L_3$ . The windows for loads 1 through 3 are  $[a_1, b_1], [a_2, b_2], [a_3, b_3]$ , respectively, while  $b_1^*, b_2^*, b_3^*$  are the time points in which the assignment is made. Travel times from  $L_1$  to  $L_2$  and from  $L_2$  to  $L_3$  are  $t_{1,2}, t_{2,3}$  respectively. The probability for load  $L_1$  to be served before time point  $b_1$  is clearly greater than  $1-\alpha$ . The probability that  $L_2$  will be served is as follows,

$$\begin{aligned} p(T_2 \leq b_2) &\geq p(T_2 \leq b_2^*) \\ &= p(t_{1,2}^d + a_1 \leq b_2^* | T_1 \leq a_1) p(T_1 \leq a_1) \\ &\quad + p(t_{1,2}^d + T_1 \leq b_2^* | a_1 \leq T_1 \leq b_1^*) p(a_1 \leq T_1 \leq b_1^*) \\ &\quad + p(t_{1,2}^o + T_1 \leq b_2^* | T_1 \geq b_1^*) p(T_1 \geq b_1^*) \\ &> (1-\alpha) p(T_1 \leq a_1) + (1-\alpha) p(a_1 \leq T_1 \leq b_1^*) \\ &= (1-\alpha) p(T_1 \leq b_1^*) \\ &\geq (1-\alpha)^2 \end{aligned}$$

This conclusion can be proved inductively. Suppose the  $n^{\text{th}}$  load is served with a probability bigger than  $(1-\alpha)^n$  before time point  $b_n^*$ , then we can show that the  $(n+1)^{\text{th}}$  load can be served with the probability bigger than  $(1-\alpha)^{n+1}$ . (End of proof.)

According to Proposition I, we can estimate the probability that the load will be served under a given assignment. Suppose a load is fourth in an assignment, and suppose that the confidence level is set to 0.95. Then we can easily obtain a bound for the probability that the load will be served. This bound is  $0.95^4$ . For local truckload trucking problems, a typical assignment chain is three loads and chains of more than five loads are highly unusual. Note that this is only a lower bound, it does not represent the actual likelihood that the load will be served. The actual service probability is expected to be higher than this.

With help of proposition I, a bound on the expected number of loads served can be estimated. We show this in Proposition II.

### **Proposition II**

For a given assignment of  $n$  loads based on the network implied by Formulation I with each link having a confidence level above  $1-\alpha$ , the expected number of loads served is more than  $[1-(1-\alpha)^{n+1}]/\alpha - 1$ .

## Proof

This result follows immediately from Proposition I by applying the equation for the sum of a series with equal ratio between two consecutive terms. The first term is  $(1-\alpha)$ , the last term is  $(1-\alpha)^n$ , and the ratio between any consecutive terms is  $(1-\alpha)$ . (End of proof.)

The lower bound for the number of loads served indicated in Proposition I can be further improved under moderate assumptions. The reason for this is that missing of a load leads to higher service rate of the loads following it. Remember that we assume that missed loads are lost. In fact, in an actual operation missed loads would typically be served by another driver or subcontracted to a local owner operator or dray service. Under that assumption, the loaded move and handling time associated with the missed load is saved. In the problem of local truckload trucking assignment, travel time is fairly small relative to the handling time (including dock-side waiting time). Therefore, saving the handling time associated with a single load can contribute significantly to the increase of the service probability of later loads in the chain. After adding a very moderate assumption, we develop a better estimate of the lower bound.

The assumption is that the probability of serving a load that immediately follows a service failure is greater than  $1-\alpha$  (equation 3.10).

$$p(t_{j-1,j}^o + T_{j-1} \leq b_j' | T_{j-1} \geq b_{j-1}') > 1-\alpha \quad \forall j \quad (3.10)$$

where  $b_j^*$  is the time point used for assignment.

Then the following conclusion applies.

### **Proposition III**

With condition (3.10) imposed, each load can be served with a probability no less than  $1-\alpha$ .

#### **Proof**

We create a surrogate set of windows in which the last time points are  $b_i^*$ ,  $i \leq n$  with  $b_i^*$  being the time point used for making the assignment and the starting points are the same as original windows. Because these time windows are shorter than the true time windows, the expected number of loads served will be bounded from above by that of the set of loads with original windows. Therefore, if we show that the service probability with the surrogate windows is greater than  $1-\alpha$ , then the true service probability must also be greater than  $1-\alpha$ . Let's start with the 2<sup>nd</sup> load.

$$\begin{aligned}
p(t_{1,2} + T_1 \leq b_2) &= p(t_{1,2}^d + a_1 \leq b_2 | T_1 \leq a_1) p(T_1 \leq a_1) \\
&\quad + p(t_{1,2}^d + T_1 \leq b_2 | a_1 \leq T_1 \leq b_1) p(a_1 \leq T_1 \leq b_1) \\
&\quad + p(t_{1,2}^o + T_1 \leq b_2 | T_1 \geq b_1) p(T_1 \geq b_1) \\
&> (1 - \alpha) p(T_1 \leq a_1) + (1 - \alpha) p(a_1 \leq T_1 \leq b_1) + (1 - \alpha) p(T_1 \geq b_1) \\
&= (1 - \alpha) p(T_1 \leq b_1) + (1 - \alpha) p(T_1 \geq b_1) \\
&= (1 - \alpha)
\end{aligned}$$

From equation (3.5a) and (3.5b), we observe that the variation of the arrival time at a load in a given assignment must be smaller in the case with strict time windows than in the case without. The main difficulty with the assignment model is that the effect of the stochasticity of a service time on the service of subsequent loads can not be easily evaluated. If we ignore this effect, a simpler model can be developed. In this model the objective is to maximize the sum of expectations of individual links.

### 3.4.1.3 Formulation II

In this formulation, discretization of time windows is performed as before. However, instead of only selecting those links with a confidence level above a certain value and treating them equally, we select all the links with a probability over certain value  $\alpha$  and differentiate them by their associated probabilities. Each probability is considered differently in the assignment model by associating with it an expected cost or revenue. Then we formulate the problem and solve it.

More specifically, the objective function is set as follows,

$$obj \ Min \sum_{\substack{i \in N + \{o_k | k \in K\} \\ \delta(i) \neq \delta(j)}} \sum_{j \in \omega \setminus \{i\}} [-M(1 - \alpha_{i,j}) + c_{i,j}] x_{i,j} \quad (3.11)$$

Subject to:

Constraints (2.1b)–(2.3b), (2.6b) and the following,

$$x_{i,j} [P(t_{i,j} < b_j - b_i) - (1 - \alpha_{i,j})] = 0 \quad (3.12)$$

$$\alpha_{i,j} \leq \alpha \quad (3.13)$$

Here  $\alpha$  can be much lower than that used in formulation I. For example, it could be 0.4 or 0.5 though smaller values lead to larger networks, which may be computationally intractable.

The model provides an improvement to formulation I in some cases. An argument for this model is that formulation I is too conservative. figure 3.3 provides an example to illustrate this point.

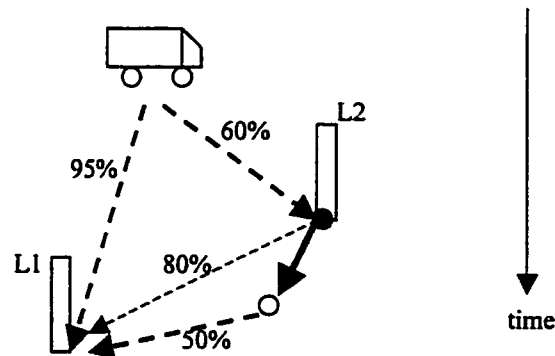


Figure 3.3) Illustration of different performance of networks developed under formulations I and II

In the example in figure 3.3, the network implied by formulation I leads to assignment  $L_1$  only if the probability level is set to be above 0.90. The expected number of loads served is 0.95 while formulation II could lead to an assignment  $L_2L_1$  which has an expected number of loads served 1.22, higher than that of the preceding assignment.

### 3.4.2 A Mixed Approximation Model of Stochastic Assignment with Recourse

We call the model mixed in that links with too small a probability are eliminated. Therefore it is not a fully stochastic assignment model. In addition, in this model, the evaluation of the expectation term is only an approximation.

An approximation of the expectation term is used for simplicity. First we introduce the approximation method to evaluate a given assignment. A simpler way to evaluate a given assignment is obtainable if we make the condition imposed by Equation (3.10) stronger. In place of Equation (3.10), we assume

$$p(t_{j-1,j} + T_{j-1} \leq b_j' | T_{j-1} \geq b_1') = 1 \quad (3.10')$$

which says that when the load directly preceding a given load is missed, then the given load is guaranteed to be served on time (because most of the service time associated with the missed load is saved). Then we can simply approximate the expected number served in a given assignment  $L_1L_2...L_n$  for vehicle  $i$  in the following way.

$$E_i' = \sum_{j=1}^n \alpha'_{j-1,j} \quad (3.14)$$

While  $\alpha'_{j-1,j}$  is obtained recursively through the following procedure:

$$\alpha'_{j-1,j} = \alpha'_{j-2,j-1} \alpha_{j-1,j} + (1 - \alpha'_{j-2,j-1}) \quad (3.15)$$

Where  $\alpha_{j-1,j}$  represents the probability associated with the link (connecting two sub-loads) from sub-load  $j-1$  to  $j$ .

The travel cost associated with the  $j^{th}$  load is  $c_{j-1,j}^d \alpha'_{j-1,j} \alpha_j + c_{j-1,j}^o (1 - \alpha'_{j-1,j})$ , where  $c_{j-1,j}^d$  and  $c_{j-1,j}^o$  are the cost of travel to origin of load  $j$  from the origin of load  $j-1$  when load  $j-1$  is served and missed respectively. So, the total travel cost of a given assignment of a vehicle  $i$  is as follows:

$$E_i(c) = \sum_{j=1}^n [c_{j-1,j}^d \alpha'_{j-1,j} \alpha_j + c_{j-1,j}^o (1 - \alpha'_{j-1,j})] \quad (3.16)$$

And, the total expectation including the number of loads served and the travel cost should be the following:

$$Exp = \sum_{i \in K} (-M * E_i' + E_i(c)) \quad (3.17)$$

The difference between this approximate expectation and its true value comes from assumption in Equation (3.10'). If assumption (3.10') is very close to reality, then the expectation obtained in this way should be very close to that of the optimal assignment. Because the fraction of the window after the time points considered for the assignment are not included, the expected number of loads served obtained here is

smaller than it should be. However, under assumption (3.10'), the expected number would be greater than it should be. Therefore, the expectation should be fairly close to the real value.

The formulation of this model is given by:

*obj min Exp*

Subject to:

Constraints (2.1b)~(2.4b), (2.6b) and (3.12), (3.13).

Even though the evaluation process is approximated in a more efficient way, one which is at the order of  $O(n)$ , the method still has room for improvement. So far, it has to go through all the integer points, which is inefficient. Further research is still open for the estimation of the objective value at the fractional point in the branch and bound tree. In such problems as these where stochastic factors complicate the already complicated models, approximation methods are usually needed.

### 3.5 Conclusion

The truckload trucking assignment problems with stochastic service time and time windows is important in practice. In this chapter, this problem is examined. A method to evaluate the a priori assignment is developed. The complexity of this method makes typical ones in the stochastic vehicle routing problem inefficient if applied in the problem considered in this chapter. Alternatives to overcome this difficulty are provided. A bound

is given to estimate the expected number of loads served in formulation I, a conservative formulation. This bound is strong in the sense that it does not depend on any assumption about the distribution of the service time. Based on observation, a less conservative chance constrained method, formulation II is provided. By adopting an approximation to the evaluation method, a mixed approximation model of stochastic assignment with recourse is introduced. Approximation models are preferable in very complicated situations where accurate calculation is impractical or even infeasible.

Results presented in this chapter are preliminary. They represent a starting point for research on a challenging class of problems. Continuing research includes the development of improved formulations and an examination of the effectiveness of the proposed models when compared with deterministic dynamic assignment models applied in a rolling horizon framework.

## **CHAPTER 4**

### **MODELS WITH TRAILER POSITIONING**

#### **4.1 Introduction**

One of the most important features for the dynamic vehicle assignment problem is vehicle positioning. Repositioning is performed in anticipation of future demand. However, in local truckload trucking, demands occur within a relatively compact region and hence are within close reach of the fleet of vehicles. Repositioning of vehicles is not as significant as it is in long haul truckload trucking in which loads could be missed if vehicles are not pre-positioned. Nonetheless, the problem considered here does have a similar feature. Trailer repositioning is required in anticipation of future demand. In local truckload operations travel time is relatively short compared to loading time. Therefore, trailers need to be positioned early to allow for loading time. This creates a problem in which the assignment manages several resources: empty trailers, drivers and loaded trailers. Assignment of one resource depends heavily on that of another. The objective of the research on this problem is to develop methods to minimize the total cost to provide service including trailer positioning moves (deadhead), empty moves (bobtail) and loaded moves.

Positioning of trailers depends on future demands. For regular customers there should always be trailers positioned whenever possible. For occasional customers, trailers are usually positioned as requested. In the case of regular customers, the number of

parking positions for trailers is limited and variable, so positioning moves may have other constraints. A vehicle can leave for another assignment in a move called “bobtail move” after each positioning. In this process, the vehicle does not need to be coupled with the trailer. figure 4.1 illustrates the work process.

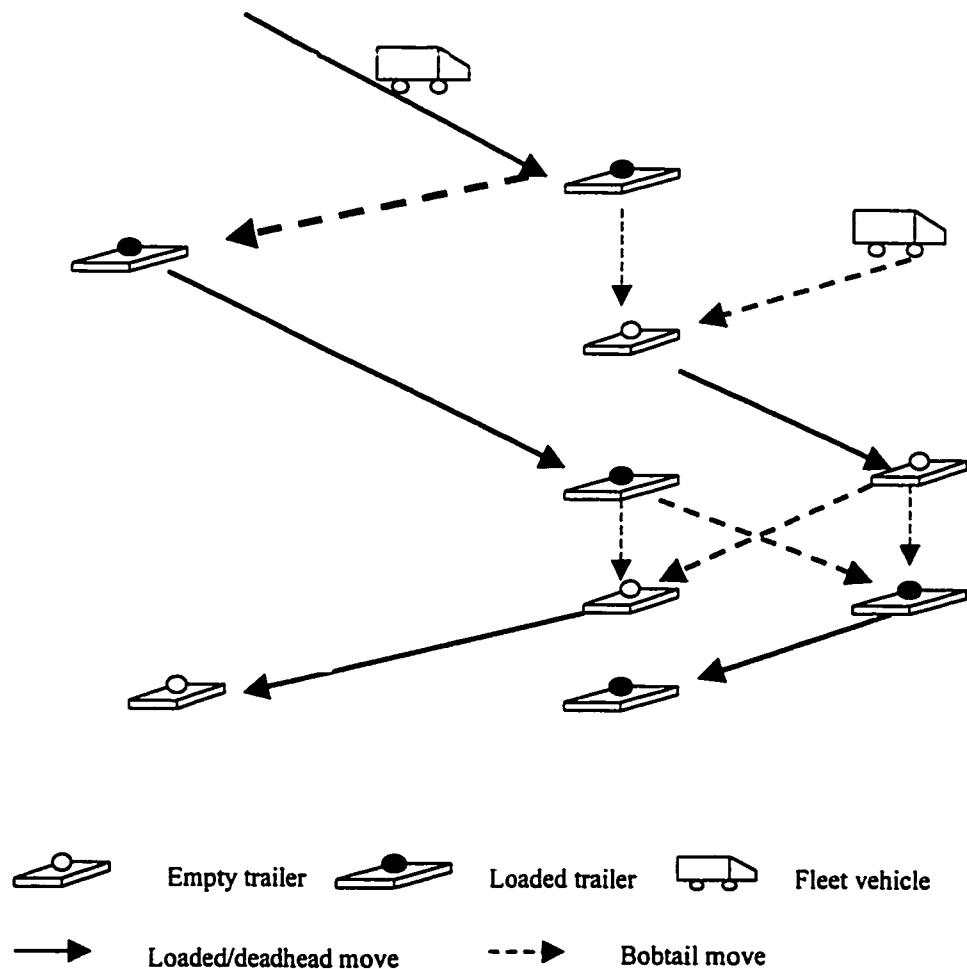


Figure 4.1) A multi-layered work process

In figure 4.1, the vehicle on the left moves loads to a location while vehicle on the right moves the empty trailers to needy customers. A loaded movement generates two resources: an available driver and an empty trailer and requires two resources: a driver and a trailer. The problem is formulated as follows:

## Notation

$x_{ij}^{E,(t1,t2)}$  is the decision variable for the bobtail move from node  $i$  to node  $j$ , which starts at time  $t1$  and arrives at time  $t2$ .

$x_{ij}^{H,(t1,t2)}$  is an integer variable for number of loads in the loaded move from node  $i$  to node  $j$ , which starts at time  $t1$  and arrive at time  $t2$ .

$x_{ij}^{P,(t1,t2)}$  is an integer variable for the number of trailers in the positioning moves from node  $i$  to node  $j$ , which starts at time  $t1$  and arrive at time  $t2$ .

$D_i^0$  is the number of drivers available at time of the assignment

$R_i^0$  is the number of trailers available at the time of assignment

$T^h$  is the handling time (loading/unloading),  $T$  is the planning horizon

$\omega_{ij}^{t1}$  is the set of loads with the origin  $i$  and destination  $j$  that can be served at time  $t1$

$N$  is the set of nodes;  $A$  is the set of links for loaded move

$$\text{Max } \sum_{t1} \sum_{(i,j) \in A} (M + r_{ij}) x_{ij}^{H,(t1,t2)} - \sum_{t1} \sum_i \sum_j c_{ij} x_{ij}^{E,(t1,t2)} - \sum_{t1} \sum_i \sum_j c_{ij} x_{ij}^{P,(t1,t2)} \quad (4.0)$$

subject to

$$\sum_{t1} x_{ij}^{H,(t1,t2),\omega} \leq |\omega_{ij}| \quad \forall t1 \leq T, (i,j) \in A \quad (4.1)$$

$$\begin{aligned} \sum_j x_{ij}^{H,(t1,t2)} &\leq \sum_{t2' \in T^h < t1} \sum_k x_{ki}^{H,(t1',t2')} + \sum_{t2' < t1} \sum_k x_{ki}^{P,(t1',t2')} \\ &\quad - \sum_{t1' < t1} \sum_k x_{ik}^{H,(t1',t2')} - \sum_{t1' < t1} \sum_k x_{ik}^{P,(t1',t2')} + R_i^0 \end{aligned} \quad \forall t1 \leq T, i \in N \quad (4.2)$$

$$\begin{aligned} \sum_j x_{ij}^{H,(t1,t2)} &\leq \sum_{t2' \in T^h < t1} \sum_k x_{ki}^{H,(t1',t2')} + \sum_{t2' < t1} \sum_k x_{ki}^{P,(t1',t2')} + \sum_{t2' < t1} \sum_k x_{ki}^{E,(t1',t2')} \\ &\quad - \sum_{t1' < t1} \sum_k x_{ik}^{H,(t1',t2')} - \sum_{t1' < t1} \sum_k x_{ik}^{P,(t1',t2')} - \sum_{t1' < t1} \sum_k x_{ik}^{E,(t1',t2')} + D_i^0 \end{aligned} \quad \forall t1 \leq T, i \in N \quad (4.3)$$

$$\begin{aligned} \sum_j x_{ij}^{P,(t1,t2)} &\leq \sum_{t2' \in T^h < t1} \sum_k x_{ki}^{H,(t1',t2')} + \sum_{t2' < t1} \sum_k x_{ki}^{P,(t1',t2')} + \sum_{t2' < t1} \sum_k x_{ki}^{E,(t1',t2')} \\ &\quad - \sum_{t2' \in T^h < t1} \sum_k x_{ik}^{H,(t1',t2')} - \sum_{t2' < t1} \sum_k x_{ik}^{P,(t1',t2')} - \sum_{t2' < t1} \sum_k x_{ik}^{E,(t1',t2')} + D_i^0 \end{aligned} \quad \forall t1 \leq T, i \in N \quad (4.4)$$

$$\begin{aligned} \sum_j x_{ij}^{P,(t1,t2)} &\leq \sum_{t2' \in T^h < t1} \sum_k x_{ki}^{H,(t1',t2')} + \sum_{t2' < t1} \sum_k x_{ki}^{P,(t1',t2')} \\ &\quad - \sum_{t2' \in T^h < t1} \sum_k x_{ik}^{H,(t1',t2')} - \sum_{t2' < t1} \sum_k x_{ik}^{P,(t1',t2')} + R_i^0 \end{aligned} \quad \forall t1 \leq T, i \in N \quad (4.5)$$

$$\begin{aligned} \sum_j x_{ij}^{E,(t1,t2)} &\leq \sum_{t2' \in T^h < t1} \sum_{k \in N} x_{ki}^{H,(t1',t2')} + \sum_{t2' < t1} \sum_{k \in N} x_{ki}^{P,(t1',t2')} + \sum_{t2' < t1} \sum_{k \in N} x_{ki}^{E,(t1',t2')} \\ &\quad - \sum_{t2' \in T^h < t1} \sum_{k \in N} x_{ik}^{H,(t1',t2')} - \sum_{t2' < t1} \sum_{k \in N} x_{ik}^{P,(t1',t2')} - \sum_{t2' < t1} \sum_{k \in N} x_{ik}^{E,(t1',t2')} + D_i^0 \end{aligned} \quad \forall t1 \leq T, i \in N \quad (4.6)$$

$$\text{if } x_{ij}^{H,(t1,t2)} > 0, \quad t2 = t1 + t_{ij}, \quad (4.7)$$

$$\text{if } x_{ij}^{E,(t1,t2)} > 0, \quad t2 = t1 + t_{ij}, \quad (4.8)$$

$$\text{if } x_{ij}^{P,(t1,t2)} > 0, \quad t2 = t1 + t_{ij}, \quad (4.9)$$

$$\text{Binary } x_{ij}^{E,(t1,t2),\omega}, \text{ integer } x_{ij}^{P,(t1,t2)}, x_{ij}^{H,(t1,t2)} \quad (4.10)$$

Constraints (4.1) stipulate that loaded move sums up to no more than the loads available at any time. Constraints (4.2) require conservation of trailers in loaded move. Constraints (4.3) indicates drivers conservation in the loaded move. Constraints (4.4) prescribe drivers conservation in trailer positioning move. Constraints (4.5) say trailer conservation

in trailer positioning move. Constraints (4.6) state the driver conservation in bobtail move.

From the formulation, we see that this is a very complicated problem. Although no effort towards the final solution of this problem has been made in this research, we try to define and analyze this problem because of its rich practical implications. The analysis here is intended to highlight a promising topic for future research.

## **4.2 Related Work**

This is a problem that manages multi-layered resources. It is an example from a class of problems known as dynamic resource allocation problems. An important dimension of resource management is the principle of layering, where primitive resources (such as drivers, tractors and trailers) are joined to form composite resources. The management of each layered resource can itself be an NP-hard problem. Solution of each individual problem alone does not provide sufficient benefits in most real operations. The reason is that good management of each single layer often does not contribute much to the efficiency of the whole system. Simultaneous consideration of all layers, which makes the problem even harder to solve, is required to obtain significant benefits. This is especially true when management of one layer is not independent of management of another. Layers may possess inherent temporal or causal relationships. The specific problem considered in this chapter is concerned with management of three layers: drivers, empty trailers and loads. Research on the multi-layered problem is rewarding in

another sense that it has wide practical applications in rail and air in addition to trucking operations.

While the scale of such problems is prohibitive and prevents a direct application of basic skills of general integer programming problems, analysis of the temporal and spatial structure of the problem begs careful consideration of a method known as “Logistics Queuing Network”. The Logistics Queuing Network (LQN) method, which was first proposed by Powell *et al* (1995) and then revisited by Powell and Carvalho(1998) and again by Carvalho and Powell(2000), aims to take advantage of a special space-time structure of the problem. It decomposes the problem into a set of local terminal problems that can be solved easily. The potential of locating a resource at each terminal is used in the solution of each terminal problem. By adjusting this potential repeatedly, the problem can be solved iteratively until fairly close to optimum. This method is very similar to the Lagrangian relaxation method except that the LQN method is applied to integer rather than continuous optimization problems. Adjusting the potential is similar to adjusting the multipliers in the objective function. While experiments as reported in the literature show that it works well to provide fairly good solutions very quickly, this method can only be applied to our specific problem with great difficulty since it requires discretization of the continuous time horizon. As discussed in Chapter 2 for a related problem, narrow discretization can lead to exponentially expanding problems while large intervals to discretize typically result in loss of optimality.

### **4.3 Future Research**

Even if direct application of the LQN method can be successful for this problem, the amount of work required is prohibitive. Therefore, in this dissertation we define and analyze, but do not attempt to solve the problem. We mention the problem because its solution would be meaningful not just for this specific dynamic resource allocation problem alone, but for the solution of a related problem which explicitly incorporates in the problems stochastic factors in the manner discussed in Chapter 3. Incorporation of stochastic factors would further make this problem distinct from its counterparts in the published literature. Further still, incorporation of more stochastic factors could result in a more realistic and robust solutions in practice. This could also possibly lead to development in algorithms and theory.

## **CHAPTER 5**

### **RESEARCH EXTENSIONS**

#### **5.1 Introduction**

As discussed in Chapter 2, the deterministic myopic version of the truckload trucking assignment problem can be shown to be an m-TSP problem. The optimization version of the m-TSP problem has been shown to be NP-hard in the strong sense. As a member of the NP-hard family, the m-TSP is connected to all other NP-hard problems, including the best known of these, the general TSP. The TSP provides the foundation for most NP-hard vehicle routing problems. Many of these can be decomposed into TSP subproblems. Further, work on the TSP applies directly to problems in many other areas including transportation, computer networking, manufacturing and telecommunications. The rich theoretical and practical implications of the TSP problem have attracted much attention from the research community. Another NP-hard problem is general integer programming. Its efficient solution leads to efficient solution of the TSP problem as well as other NP-hard problems. While most of our research is devoted to variants of the TSP problem, our attention was also drawn to solution methods for integer programming problems. In this chapter, we describe investigations that are complementary but separate from the main topic of this dissertation. The first investigation resulted in a new formulation for a special case of the TSP, namely the TSP with separation requirements. This formulation resulted from work on the probabilistic TSP, which attracted our interest as a method for generating a priori solutions in a dynamic operating environment.

Such methods provide an alternative to deterministic dynamic routing and scheduling methods, applied in a rolling horizon framework. The second investigation resulted in the development of a new cutting plane method which applies to general integer programming problems with binary variables. Our findings are explained in the following two sections.

## **5.2 The Traveling Salesman Problem with Separation Requirements: An Examination of Alternative Formulations**

While there is a rich literature on the TSP problem including exact methods, heuristic methods, asymptotic optimal heuristic algorithms and many other topics as well, it is not our intention in this dissertation to provide a full overview of such research on either the TSP or integer programming methods. We simply present our findings and discuss the most relevant literature.

### **5.2.1 Problem Statement**

Consider a set of nodes  $N$ , each of which requires a visit. The goal is to construct the least cost feasible tour visiting all of these nodes. There is a set of costs associated with any pair of nodes, each of which corresponds to a situation with a different number of nodes lying between this node pair. We refer to this as the separation cost later. In addition, for any pair of nodes  $(i, j)$ , there may be a separation requirement, which can be explained in the following way.

- a) Node  $i$  must be visited  $m$  nodes before node  $j$ , where  $m$  could be any reasonable value bigger than zero.
- b) Node  $i$  must be visited exactly  $m$  nodes apart from node  $j$ .
- c) Node  $i$  must not be visited exactly  $m$  nodes apart from  $j$

This is a traveling salesman problem with separation requirements (TSPS). If there are no separation requirements, this problem reduces to the traveling salesman problem (TSP). A special case of this problem is the TSP with precedence constraints. Further, by explaining the separation cost in a special way, this problem is equivalent to the time dependent TSP problem.

### 5.2.2 Related Work

To our knowledge, this problem has not been explicitly addressed in the literature. However, there do exist formulations for related problems that can be applied to the TSPS. For example, Jaillet (1985) provides three mathematical programming formulations for the Probabilistic TSP. One is non-linear integer programming formulation, another is a mixed integer linear programming formulation, while the third is a pure integer linear programming formulation. All these formulations, though arising from PTSP, apply to TSPS. However, they are all extensions of the classical formulation of Dantzig, Fulkerson and Johnson (1954), the so-called DFJ formulation. As a result, their solutions unavoidably involve a follow up check of the sub-tour violation during the course of branch and bound procedure. Further still, it can be shown that though they

include a large number of additional variables, these integer linear formulations are equivalent to DFJ formulation in terms of their polytopes (as in Lawler, 1963).

Another class of problem closely related to the TSPS is the time dependent traveling salesman problem (TDTSP). In the TDTSP, the cost of a link leading out of a node is a function of the position of this node in the tour. By explicitly defining the position of each node relative to the reference node by decision variables, the relative position of any node pair is implicitly defined. As a result, the integer programming formulation in quadratic form (referred to as QAP) in Picard and Queranne (1978) can be applied to the TSPS problem. In this formulation, separation requirements could be considered by associating a sufficiently large cost with each term corresponding to a separation situation in the objective function. Although there are some other formulations for the TDTSP, these do not apply to the TSPS problem because of the way the decision variables are defined. On the other hand, the formulation for TSPS problems presented in this paper readily applies to TDTSP problems.

There has been rich research on the traveling salesman problem with precedence constraints (TSPPC). Some of the representative work can be found in Balas, Fischetti and Pulleyblank (1995), Balas, Lenstra and Vazacopoulos (1995) and Fiala Timlin and Pulleyblank (1992). Precedence constraints do not consider a specific number of nodes lying between a node pair as in TSPS problem. Inversely, however, we show by example that formulations for TSPS problem could provide an alternative method to accommodate precedence constraints. Therefore, the TSPPC is a special case of TSPS.

In this section, we provide an alternative integer programming formulation of the problem that can be solved with general branch and bound algorithm. Unlike the formulations for the PTSP (Jaillet, 1985), it does not require follow up check of sub-tour violation in the algorithm. In addition our formulation does not have explicit sub-tour elimination constraints and has fewer variables than those formulations. Further, it is better than QAP for this problem in that it has an objective function in linear form and a stronger constraint set. These properties will be explored in the following section.

Our contributions can be stated as follows.

1. We provide an integer programming formulation for the traveling salesman problem with separation requirement. This formulation applies to the traveling salesman problem with precedence constraints (SPPC) and time dependent traveling salesman problem (TDTSP) as special cases.
2. This formulation leads to a new formulation for TSP. Analytical properties are explored. Analysis shows that this formulation provides a tight subtour polytope for small subtours.

Here we introduce and discuss some analytical properties of the formulation.

### 5.2.3 Integer Programming Formulation

Before going directly to the new formulation, we first present QAP by Picard and Queranne (1978) for later reference.

$$\begin{aligned} \min & \sum_i \sum_j \sum_{k>t} \sum_t c_{ij}^{t,k} x_{it} x_{jk} \\ \text{s.t.} & \\ & \sum_{t=1}^n x_{it} = 1, \quad \forall i \end{aligned} \tag{5.1}$$

$$\sum_{i=1}^n x_{it} = 1, \quad \forall t \tag{5.2}$$

$$x_{it} \in \{0,1\} \quad \forall i,t \tag{5.3}$$

Where  $x_{it}$  equals to one when node  $i$  is at the  $t^{\text{th}}$  position on the given tour; zero, otherwise.  $c_{ij}^{t,k}$  is the cost associated with the node pair  $(i,j)$  when node  $i$  is in the  $t^{\text{th}}$  position while node  $j$  is positioned  $k^{\text{th}}$  in the tour. In the TDTSP problem, all the terms with  $k$  greater than  $t$  plus one are set to be zero. In addition, for all  $i$ ,  $c_{ii}^{t,t}$  are zero except  $c_{ii}^{1,1}$  and  $c_{ii}^{n,n}$ , which are the cost from depot to node  $i$  (when  $i$  is positioned first), and the cost from node  $i$  back to the depot (when node  $i$  is positioned the last), respectively. Constraints (5.1) prescribe that each node can only be placed at one position. Constraints (5.2) require that each position can only be occupied by one node.

For the new formulation, new variables are needed.

Let,

$x_{i,j}^k$  = One when there are  $k$  nodes between node  $i$  and  $j$  in the tour along the designated direction; zero, otherwise.

We refer to  $x_{i,j}^k$  as the adjacency variable. Then the problem can be formulated using adjacency variables in the following way (referred to as LOF for Linear Objective Formulation):

$$obj \min \sum_{i \in N} \sum_{j \in N \setminus \{i\}} \sum_{k=0}^{|N|-2} c_{i,j}^k x_{i,j}^k \quad (5.4)$$

s.t.

$$\sum_{j \in N \setminus \{i\}} x_{i,j}^k = 1 \quad \forall i \in N, 0 \leq k \leq |N|-2 \quad (5.5)$$

$$\sum_{j \in N \setminus \{i\}} x_{j,i}^k = 1 \quad \forall i \in N, 0 \leq k \leq |N|-2 \quad (5.6)$$

$$\sum_k x_{i,j}^k = 1 \quad \forall (i,j), i \neq j \quad (5.7)$$

$$x_{i,j}^k - x_{j,i}^{|N|-k-2} = 0 \quad 0 \leq k \leq |N|-2, \forall (i,j), i \neq j \quad (5.8)$$

$$\sum_k k x_{i,m}^k + \sum_k k x_{j,i}^k + 1 \geq \sum_k k x_{j,m}^k \quad \forall (i,j,m) \quad i \neq j \neq m \quad (5.9)$$

$$binary \quad x_{i,j}^k \quad \forall i,j,k \in N \quad (5.10)$$

Constraints (5.5) and (5.6) define the uniqueness of each position before and after each node on the tour respectively. Constraints (5.7) define the uniqueness of the relative position of any two nodes. Constraints (5.8) dictate the complementarity of relative positions of any pair of nodes. The fifth set of constraints (5.9) enforce a modified triangle inequality concerned with the number of nodes separating any two nodes in the tour rather than the distance between nodes.

The formulation contains  $2n^2(n-1)$  constraints and  $n(n-1)^2$  variables, where  $n$  is the number of nodes. The tour has a certain direction. Let us arbitrarily assume the tour direction is clockwise.

**Lemma 1** Constraints (5.9) hold in any tour.

**Proof**

For any three nodes  $\{i, j, \ell\}$ , there are two possible sequences along the given direction.

These are  $(i, j, \ell)$  and  $(i, \ell, j)$ .

In the case of sequence  $(i, j, \ell)$ , the following holds:

$$\sum_k kx_{i,j}^k + \sum_k kx_{j,\ell}^k + 1 = \sum_k kx_{\ell,j}^k$$

While in the case of sequence  $(i, \ell, j)$  the following holds:

$$\sum_k kx_{i,j}^k + \sum_k kx_{j,\ell}^k + 1 > \sum_k kx_{\ell,j}^k$$

Hence, the constraints (5.9) always hold.

**Proposition 1** Constraints (5.5) through (5.10) define a tour.

## Proof

Suppose there are  $n$  nodes. Let us begin with one node, and suppose it has index 1. Consider all the constraints when  $i$  is equal to 1 in the constraints (5.4) through (5.8). These constraints and (5.10) form a set  $S_1$ .  $S_1$  includes the same constraints as in QAP, so  $S_1$  defines a tour. Similarly, we have set  $S_i$  for node  $i$ . For any  $i \in N$ ,  $S_i$  also defines a tour. Hence we have  $n$  tours, each starting with a different node. Now we will show all the  $n$  tours are identical by further considering the constraints (5.8) and (5.9).

For simplicity of expression, we denote by  $L_i$  a tour defined by  $S_i$ , which is a series of  $n$  nodes beginning with node  $i$ . We assume  $L_1$  is  $(1, 2, \dots, n-1, n)$ , where  $2, 3, \dots, n-1, n$  are index numbers for the nodes at the corresponding positions in the tour  $L_1$ . Then we start with the second node indexed 2, which is immediately after node 1 in  $L_1$ . To begin, we show that  $L_2$  is identical to  $L_1$ . Considering node 2 is immediately after node 1 in  $L_1$ , we get  $x_{1,2}^0 = 1$ . By the complimentary constraints (5.8), we have  $x_{2,1}^{n-2} = 1$  from the fact  $x_{1,2}^0 = 1$ , which means that node 1 must be at the end of  $L_2$ . Then let us consider the position for node  $n$  in the tour  $L_2$ . Applying constraints (5.9), we have

$$\sum_k x_{1,2}^k + \sum_k x_{2,n}^k + 1 \geq \sum_k x_{1,n}^k. \text{ We then get } \sum_k x_{2,n}^k \geq \sum_k x_{1,n}^k - \sum_k x_{1,2}^k - 1.$$

Note here  $x_{1,n}^{n-2} = 1$ ,  $x_{1,2}^0 = 1$ . Hence we get  $\sum_k x_{2,n}^k \geq n-3$  from the inequality above. From this result, we can see that node  $n$  can only be put at the second to last position in  $L_2$  since the last one in  $L_2$  is occupied by node 1. Similarly, we can show that

node  $n-1$  can only be put at the third last position. By repeating this process,  $L_2$  can be shown to be  $(2, 3, \dots, n-1, n, 1)$ , which is identical to  $L_1$ . In the same reason, every  $S_i$  defines the same tour. Then we can say, all the constraints (5.5) through (5.10) define a unique tour. (End of proof)

We now examine the LP relaxations of our formulation LOF and QAP. Given any formulation  $P$ , let  $F(P)$  denote the feasible solutions of  $P$ . We denote the LP relaxation of QAP by  $QAP_L$  and LOF by  $LOF_L$ .

**Proposition 2.**  $F(LOF_L) \subset F(QAP_L)$

**Proof**

The variables in  $QAP_L$   $x_{i,t}$  corresponds to the variables  $x_{o,i}^k$  in  $LOF_L$ . The constraints (5.1) through (5.3) is only a subset of those in (5.5) and (5.7). So, every solution to  $LOF_L$  always satisfies the constraints (5.1) through (5.3). Inversely, any solution to  $QAP_L$  must satisfy those corresponding constraints in (5.5) and (5.7), but might not satisfy the remaining constraints in (5.5) through (5.7). It is sufficient to give an example here. For a network as simple as one with three nodes with node 1 being the depot, we have a feasible solution to  $QAP_L$  as follows:

$$x_{2,1} = 0.2, x_{2,2} = 0.8, x_{3,1} = 0.8, x_{3,2} = 0.2$$

Which correspond to variables in  $LOF_L$ :

$$x_{1,2}^0 = 0.2, x_{1,2}^1 = 0.8, x_{1,3}^0 = 0.8, x_{1,3}^1 = 0.2$$

However, According to constraints in (5.9), we must have

$$\sum_k kx_{2,3}^k \leq \sum_k kx_{1,3}^k - \sum_k kx_{1,2}^k - 1 = -1$$

We can easily see this is infeasible. So, the solution to  $QAP_L$  is not a feasible solution to  $LOF_L$ . (End of proof.)

Significant here, but beyond the discussion of this paper is that our formulation suggests a way to strengthen the QAP formulation.

## Examples

In table 5.1 we show how separation requirements can be accommodated in the formulation.

Table 5.1: Separation Constraints

|          |                                  |                                                        |
|----------|----------------------------------|--------------------------------------------------------|
| Case I   | $x_{i,j}^m = 1$                  | $i$ must be visited $m$ nodes before $j$               |
| Case II  | $\sum_{k < m+1} k x_{i,j}^k = 0$ | $i$ must be visited more than $m$ nodes ahead of $j$ . |
| Case III | $\sum_{k > m-1} k x_{i,j}^k = 0$ | $i$ must be visited less than $m$ nodes ahead of $j$ . |

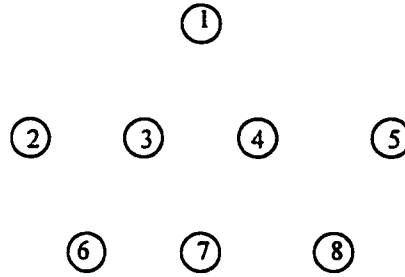


Figure 5.1) Example network

The network shown in figure 1 is fully connected. Its distance matrix is shown in table 2.

Table 5.2 Distance matrix

| Node | 1     | 2    | 3     | 4    | 5     | 6    | 7     | 8     |
|------|-------|------|-------|------|-------|------|-------|-------|
| 1    | 0     | 1.73 | 1     | 1    | 1.73  | 2    | 1.732 | 2     |
| 2    | 1.73  | 0    | 1     | 2    | 3     | 1    | 1.732 | 2.65  |
| 3    | 1     | 1    | 0     | 1    | 2     | 1    | 1     | 1.732 |
| 4    | 1     | 2    | 1     | 0    | 1     | 1.73 | 1     | 1     |
| 5    | 1.73  | 3    | 2     | 1    | 0     | 2.65 | 1.732 | 1     |
| 6    | 2     | 1    | 1     | 1.73 | 2.65  | 0    | 1     | 2     |
| 7    | 1.732 | 1.73 | 1     | 1    | 1.732 | 1    | 0     | 1     |
| 8    | 2     | 2.65 | 1.732 | 1    | 1     | 2    | 1     | 0     |

If there are no separation requirements in the example above, the problem can be formulated as described in (5.4) through (5.10). In this case, the optimal TSP solution is 1-3-2-6-7-8-5-4-1, with an associated cost of 8.0. If node 1 and node 7 must be separated by one node in the tour, then a constraint can be added as follows:  $x_{1,7}^1 + x_{7,1}^1 = 1$

The resulting solution is 1-3-7-8-5-4-6-2-1, with an associated length of 9.42.

### 5.2.4 Some Analytical Properties of the New Formulation

In the following section, we explore some properties of  $LOF_L$  with respect to its subtour polytope. Typical work in examination of this property as in Padberg and Sung (1991) also indicates this is a very important property.

**Lemma 2.** For any sub-tour  $s$ ,  $s \in E$ , where  $E$  is the set of links, assume  $s = \{\ell_{ij}, \ell_{jk}, \dots, \ell_{li}\}$ .

Let  $\Phi = \{(i,j) | \ell_{ij} \in s\}$ . The following inequality always holds in the space defined by the constraints (5.5)~(5.9)

$$\sum_{(i,j) \in \Phi} \sum_{k=1}^{|N|-2} kx_{i,j}^k \geq |N| - |s| \quad (5.11)$$

Where  $|s|$  denotes the number of nodes in the subtour while  $\ell_{ij}$  is used to denote the link from node  $i$  to node  $j$ .

**Proof**

First, we have the following result.

$$\sum_{k=0}^{|N|-2} kx_{i,m}^k + \sum_{k=0}^{|N|-2} kx_{m,i}^k = \sum_{k=0}^{|N|-2} kx_{i,m}^k + \sum_{k=0}^{|N|-2} (|N| - 2 - k)x_{i,m}^{|N|-2-k} = \sum_{k=0}^{|N|-2} (N - 2)x_{i,m}^k = |N| - 2$$

Then for any three nodes  $(i, g, m)$ , constraints (5.9) and the result above lead to the following inequality.

$$\sum_{k=0}^{|N|-2} kx_{i,g}^k + \sum_{k=0}^{|N|-2} kx_{g,m}^k + 1 \geq \sum_{k=0}^{|N|-2} kx_{i,m}^k = |N| - 2 - \sum_{k=0}^{|N|-2} kx_{m,i}^k \quad (1)$$

The result (1) will be used immediately in the following inference.

Suppose the order of the nodes along a given tour  $s$  is  $(i,j,l \dots g,m,i)$ , by the modified triangle inequality constraints (5.9), we get the following.

$$\begin{aligned} \sum_{k=0}^{|N|-2} kx_{i,j}^k + \sum_{k=0}^{|N|-2} kx_{j,l}^k + 1 &\geq \sum_{k=0}^{|N|-2} kx_{i,l}^k \\ \sum_{k=0}^{|N|-2} kx_{i,l}^k + \sum_{k=0}^{|N|-2} kx_{l,p}^k + 1 &\geq \sum_{k=0}^{|N|-2} kx_{i,p}^k \\ &\dots\dots\dots \\ \sum_{k=0}^{|N|-2} kx_{i,g}^k + \sum_{k=0}^{|N|-2} kx_{g,m}^k + 1 &\geq \sum_{k=0}^{|N|-2} kx_{i,m}^k = |N| - 2 - \sum_{k=0}^{|N|-2} kx_{m,i}^k \end{aligned} \quad (1)$$

Summing the constraints at both sides of the inequalities above, we get the following:

$$\sum_{(i,m) \in s} \sum_{k=0}^{|N|-2} kx_{i,m}^k \geq |N| - |s| \quad (m)$$

(End of proof.)

**Proposition 3.** In the space defined by the constraints (5.5)~(5.9), the following inequality always holds along any sub-tour  $s$ .

$$\sum_{(i,j) \in \Phi} x_{i,j}^0 \leq |s| - \frac{|N| - |s|}{|N| - 2} \quad (5.12)$$

## Proof

We have the following,

$$\sum_{(i,j) \in \Phi} \sum_{k=1}^{|N|-2} k x_{i,j}^k \leq \sum_{(i,j) \in \Phi} (|N|-2) \sum_{k=1}^{|N|-2} x_{i,j}^k = (|N|-2) \left[ |s| - \sum_{(i,j) \in \Phi} x_{i,j}^0 \right]$$

Combined with the following result in (5.11),  $\sum_{(i,m) \in \Phi} \sum_{k=1}^{|N|-2} k x_{i,m}^k \geq |N| - |s|$

we obtain  $\sum_{(i,j) \in \Phi} x_{i,j}^0 \leq |s| - \frac{|N| - |s|}{|N| - 2}$  (End of proof)

As in Padberg and Sung (1991), the subtour polytope in DFJ formulation is often used as a reference for comparison, which is defined in the following constraints.

$$\sum_{(i,j) \in \Phi} x_{i,j} \leq |s| - 1, \quad \forall s \subset N \mid |s| \geq 2$$

Note,  $x_{i,j}^0$  corresponds to the decision variable  $x_{ij}$  in the DFJ formulation. The constraints in our new formulation which involve only variables  $x_{i,j}^0$  correspond to the traditional flow based formulations while the remaining can be understood as the subtour elimination constraints. Compared with the traditional DFJ formulation, the sub-tour elimination effect here is weak when the subtour is of large cardinality. However, it is stronger with small subtours than with large ones.

The following is an example of an eight-node network showing that the bound in the right hand side we get by (5.12) is tight. In the subtour 5->6->7->8->5, the sum of all

the flow variables  $x_{i,j}^o$  is 3.333, equal to the bound at the right hand side,

$$|s| - \frac{|N| - |s|}{|N| - 2} = 4 - (8 - 4)/(8 - 2) = 10/3$$

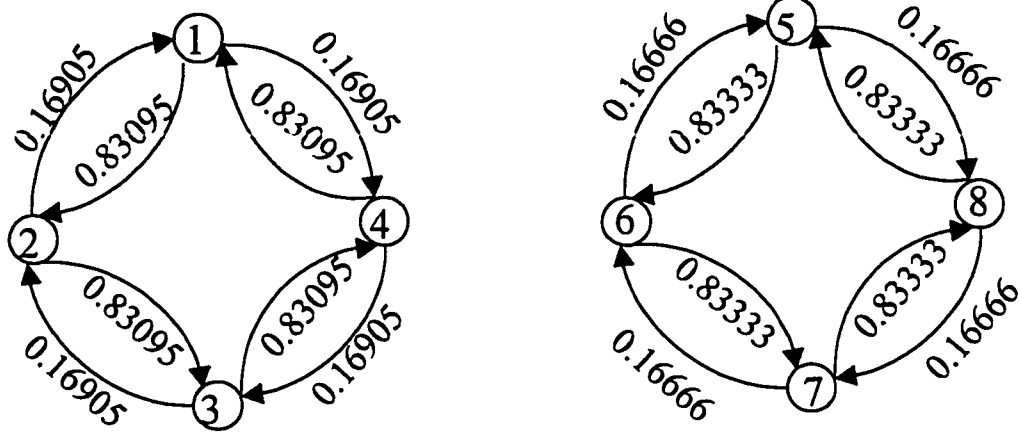


Figure 5.2) Example to show the tightness of the right hand in Equation (5.12)

### 5.2.5 Conclusion

In this section we compare alternative formulations and provide a new integer programming formulation with linear objective function for the TSP problem with separation requirements. This formulation also applies to the TDTSP problem and the TSP problem with precedence constraints. Since traveling salesman problem is a special case of this problem, it is also a new formulation for the traveling salesman problem. This formulation has more constraints than the quadratic assignment formulation QAP in Picard and Queranne (1978), and hence, its LP relaxation has a tighter polytope than that of QAP. Further analysis shows this formulation forms tight sub-tour polytope for small sub-tours compared with traditional DFJ formulation. An advantage of this formulation is

that it can accommodate separation requirements explicitly, and can also incorporate separation costs into the objective function.

### 5.3 A Cutting Plane Method for Integer Programming Problems with Binary Variables

In this section, we discuss a new general cutting plane method and provide some preliminary analysis of its characteristics.

#### 5.3.1 Introduction

Consider an integer programming problem (IP) in which the integer variables are restricted to binary values. The problem can be stated as follows.

$$\begin{array}{ll} \text{Min} & CX \\ \text{s.t.} & AX = B \\ & X \text{ binary} \end{array}$$

This is the typical IP formulation. Two types of algorithms have been developed to solve these problems. They are cutting plane methods, of which the most well known and often used is Gomory's method, and the branch and bound method. In this section of the dissertation we present a new cutting plane method. Simple analysis of it is performed. Its advantage can be stated as follows.

1. The number of constraints added is no more than the number of binary variables;
2. Can be combined with branching method to obtain a deeper cut.

### 5.3.2 The Cutting Plane Method

The new cutting plane method follows the steps listed below

- i) Select a variable  $x_i$  that is fractional in the solution to the LP relaxation of the original IP problem. Solve two formulations with this variable being set to be 1 and 0 respectively. Suppose the two formulations are represented by  $F_i(1)$  and  $F_i(0)$  with minimums  $M_i^1$  and  $M_i^0$  correspondingly. Then add a constraint,

$$CX \geq M_i^0 + (M_i^1 - M_i^0)x_i$$

This provides us with a new formulation with this additional constraint. For convenience, we call this constraint a Twin-Min cut. If either  $F_i(1)$  or  $F_i(0)$  has no solution, then fix  $x_i$  to be 0 or 1 at all the subsequent iterations. This implies that no Twin-Min cut from  $x_i$  will be added later.

- ii) Select another variable that is fractional in the solution of either  $F_i(1)$  or  $F_i(0)$ . Repeat the same process to add constraints. If there are no fractional variables in the LP solution, stop.

It should be clear that in the case of maximization, the added constraints are in the following form:  $CX \leq M_i^0 + (M_i^1 - M_i^0)x_i$ . Without explicit explanation, we refer to the case of minimization in the following sections.

## Observation

A new Twin-Min cut makes old ones of the same variable in the formulation redundant.

An argument follows. With iterations, both  $M_i^1$  and  $M_i^0$  associated with the variable  $x_i$  are monotonically non-decreasing. As a result, for each value of the variable  $x_i$ , its associated objective value is also monotonically non-decreasing after its Twin-Min cut is added. Therefore, the new cut replaces the old one. (End of argument.)

As a result, the maximum number of added constraints is equal to the number of variables. On the other hand, in Gomory's cutting plane method, there is no such a tight limit on the number of constraints added. In that method, the number of added constraints explodes with iterations. As the algorithm progresses, many constraints become redundant. However, there is no efficient way to identify redundant constraints. The identification of the redundant constraints is itself an NP-hard problem (as indicated in Nemhauser and Wolsey, 1988). Therefore an advantage of our new method is that the number of added constraints does not explode with iterations. In addition, it could be combined with a typical branch and bound method to obtain deeper cuts. In the next section we examine some of the analytical properties of the method.

### 5.3.3 Analytical Properties

#### Proposition I

For an IP formulation with two binary variables, at most two Twin-Min cuts are added.

#### Proof

Let us start with the case of maximization for graphical clarity. Suppose the two variables are  $x_1$  and  $x_2$ . We first consider adding the Twin-Min cut of  $x_1$ . For the formulations with  $x_1$  being binary, there are two possibilities when the LP relaxation of the original formulation does not have an integer solution. In one case there is an optimal integer solution and a fractional solution to  $F_1(1)$  and  $F_1(0)$  respectively. In the other  $x_2$  is fractional in both  $F_1(1)$  and  $F_1(0)$ . In the first case, we will obtain a Twin-Min cut passing through an integer point and a point with  $x_2$  being fractional while in the second case, we will end with a Twin-Min cut passing through two points with  $x_2$  being fractional. In the second case, the variable  $x_2$  would be set to a fixed integer (1 or 0) finally in the formulation when additional Twin-Min cut of  $x_2$  is further considered. Then one more branch over  $x_1$  will solve the problem ultimately. In the first case, the two cuts can be shown in the diagram as Figure 5.3. (End of proof.)

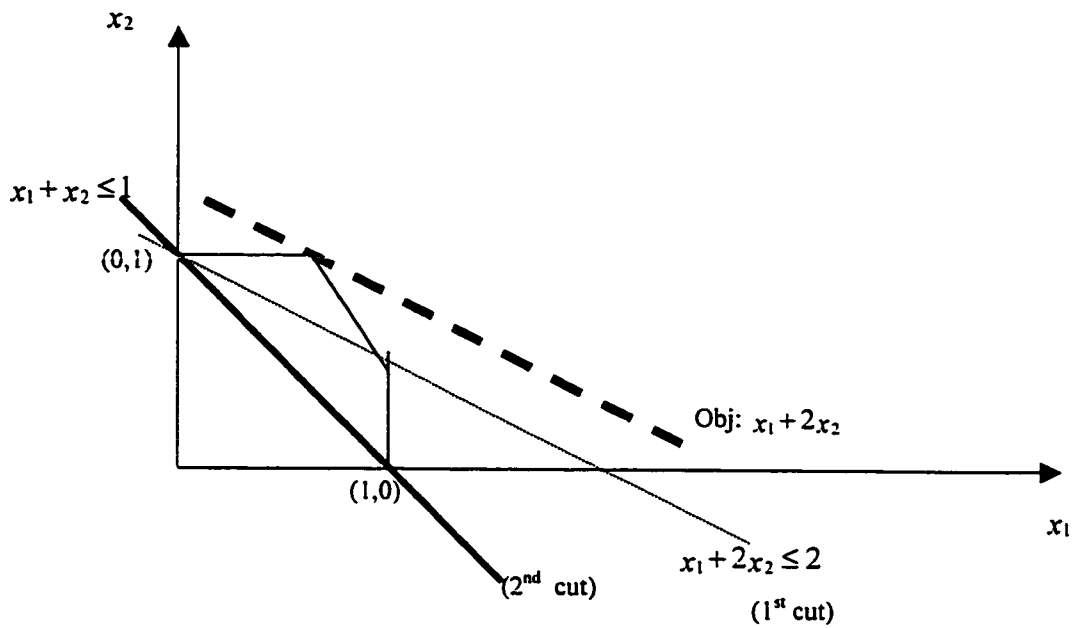


Figure 5.3) Example of cuts added

In the example of Figure 5.3, the first cut corresponds to the variable  $x_1$ , which was set to be 1 and 0 respectively while the second cut corresponds to  $x_2$  in the formulation with the first cut added.

Gomory's cutting plane method does not possess so small a limit on the number of constraints added even in this simple case.

### Proposition II

The new cutting plane method indicated in step i) and ii) guarantees finite convergence.

## Argument

A formal proof of this proposition would be lengthy. Here a brief argument based on intuition is given instead. First, the solution space is bounded at 1 and 0 at every dimension. So the optimal integer solution for a feasible problem is also bounded by certain values. The optimal fractional solution must be a limited value in the same reason. As a result, the difference between the optimal integer solution and the optimal fractional solution is limited and could be bounded by a limited value. We only need to show that the optimal fractional solution at each iteration monotonically increases by a value, which is no less than zero. This can be shown to be true. At each iteration, if the value of a variable  $x_i$  is fractional in the optimal solution, then to increase or decrease the value of this variable leads to either a bigger or equal objective value. Therefore finite convergence is guaranteed. (End of argument.)

Though we have shown that this algorithm has finite convergence, methods to raise its convergence speed are needed. A possible way to improve it is to obtain a deeper Twin-Min cut with an improved  $M_i^1$  and  $M_i^0$ . The following section explains this idea.

### 5.3.4 Obtaining Deeper Cuts

As an example, for the formulation  $F_i(1)$ , if we further split it into two new formulations with one *more* fractional variable, assume it is  $x_j$ , being 1 and 0 respectively we could end with two improved (or at least not worse) objective values for  $M_i^1$ . Select the smaller

one as  $M_i^1$ . This  $M_i^1$  must be a better bound than the one obtained without fixing  $x_i$  to be integer. In the same way, we can obtain a better bound  $M_i^0$  for  $F_i(0)$ . With both  $M_i^1$  and  $M_i^0$  being improved, we have a deeper cut corresponding to the variable  $x_i$ . Better bounds are available by further branching over more variables. In the extreme, if branching over all the variables could be performed, only one cut is needed to generate an optimal integer solution. An advantage of this cutting plane method in getting a better cut is that its branching depth (the number of variables over which branching is made) is controllable.

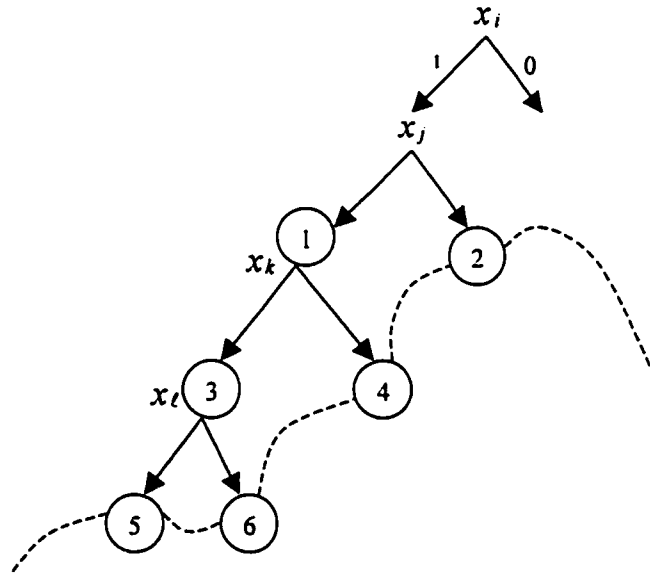


Figure 5.4) A branching procedure to get deeper cut

The method to get an improved  $M_i^1$  or  $M_i^0$  for  $x_i$  is shown in Figure 5.4. The left side, which contains the nodes from 1 to 4, is for the value  $M_i^1$ . Here let  $\phi_i$  represent the objective value at node  $i$ .  $M_i^1$  is evaluated in the following way  $M_i^1 = \min\{\phi_i\}$ , where  $i$  is

the index of the node on a cut across the tree at the side with  $x_i$  being equal to 1. A cut is shown in Figure 5.4. For clarification we provide the following definition.

### **Definition**

A set of nodes in the branching tree is defined as a *cut* if the complimentary set of the nodes is split into two sets without linkage and if there are no paths between the nodes of the cut.

If the objective value of a node on a cut is not the minimum of all the nodes on this cut, it is not necessary to branch further from this node. In short, it is only meaningful to branch over the node on a cut with the minimum value in order to get a deeper cut. In this way, a bigger  $M_i^1$  is available, which is critical for a deeper cut. Therefore the process of looking for deeper cut associated with a variable involves finding a cut of largest minimum value for the tree under a certain variable.

### **5.3.5 Conclusion**

In this section, a new cutting plane method is provided. Some simple properties are analyzed. Preliminary analysis shows it possesses some advantage over Gomory's method. One of them is that it guarantees the number of constraints added in the formulation is bounded by the number of binary variables in the problem. We show that for the simple case in which only two binary variables are present, our method guarantees

that the number of iterations required is less than Gomory's method guarantees. Though this algorithm is discussed in the context of IP problems, it applies to the mixed IP problems (MIP) too. In addition, it is possible to extend this result to the general IP/MIP problems that do not require the integer variables to be binary. An immediate task, however, is to test the performance of the proposed algorithm on typical IP formulations.

## CHAPTER 6

### CONCLUDING REMARKS

The specific problems and the contributions are summarized in Chapter 1 and clarified in the chapters that follow. In this chapter we offer some concluding remarks.

The versions of the truckload assignment problem addressed early in this dissertation form the basis for the research presented in later chapters which draws our work closer to the real world. Although we make the case for the contribution made by each part of this dissertation, we feel that the more important significance stems from the fact that our work examines a significant practical problem and then breaks this problem down into a series of challenging problems, each of which has theoretical implications. While we show the steps needed to solve this complex practical problem, each sub-problem identified remains to be solved more effectively in the future.

The assignment problem is by no means easy to solve. Our research aims to find some algorithms and strategies that can be *implemented* in practice. In fact, every algorithm or method proposed in this dissertation leads to a solution. The algorithm for the myopic deterministic version of the problem can be implemented using commercial optimization software quite easily. Stochastic models provide a compromise between mathematical complexity and practical maneuverability. The new formulation of the TSP could solve some special problems that other formulations find difficult. The cutting

plane method applies directly to the general integer programming problem with binary variables.

While we have identified the multi resource allocation problem defined for this specific problem context, our work towards its solution has been limited by time constraints. This problem is very significant in practice and has drawn much attention from researchers in recent years. Further research along these lines could have significant practical benefits and could also be theoretically rewarding.

The NP completeness theory is classical. Though the hard problems have not been solved, it does make clear the inherent connections regarding the complexity of solution between many problems that look, on the surface, substantially different. The problem studied in this dissertation is also a member of the NP-hard family. Research on the TSP problem and the IP problem, two basic NP-hard problems which are related to our practical problem, has been included in this dissertation research. The tentative results are included in Chapter 5. We hope to investigate the cutting plane method more completely in the near future.

Though the results included in this dissertation are significant, they represent a small fraction of what can be done with these problems. It seems the deeper we delve into these problems the more there is to be done. We intend to continue this line of research for several years and to extend each section of the work considerably.

## REFERENCES

- Appelgren, L.H. (1969), "A column generation approach for a ship scheduling problem", *Transportation Science*, 3 (1), pp. 53-68.
- Appelgren, L.H. (1971), "Integer programming methods for a vessel scheduling problem", *Transportation Science*, 5 (1), pp. 62-74.
- Balas, E., M. Fischetti and W.R. Pulleyblank (1995), "The Precedence-Constrained Asymmetric Traveling Salesman Problem Polytope", *Mathematical Programming* 68, 241-265.
- Balas, E., J.K. Lenstra and A. Vazacopoulos (1995), "The One-Machine Problem with Delayed Precedence Constraints and Its Use in Job Shop Scheduling", *Management Science* 41(1), 94-109.
- BTS (Bureau of Transportation Statistics) (1999), <http://www.bts.gov/ntda/nts/NTS99/data/NTS99.complete.pdf>.
- Carvalho T.A. and W.B. Powell (2000), "A Multiplier Adjustment Method for Dynamic Resource Allocation Problems", *Transportation Science*, 34(2), 150-164.
- Dantzig, G.B., D.R. Fulkerson and S.M. Johnson (1954), "Solution of a Large-Scale Traveling Salesman Problem", *Operations Research* 2, 393-410.
- Desrochers, M., J. Desrosiers and M.M. Solomon (1992), "A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows", *Operations Research* 40, pp. 342-254.
- Desrosiers, J., Y. Dumas, M.M. Solomon and F. Soumis (1995), "Time Constrained Vehicle Routing and Scheduling", in Ball, M.O., T.L. Magnanti, C.L. Monma and F.L. Nemhauser (eds), *Handbooks in Operations Research and Management Science*, Vol 8., Network Routing, Elsevier (North-Holland), Amsterdam, pp. 35-140.
- Desrosiers, J., F. Soumis and M. Sauve (1983), "Lagrangian Relaxation Methods Vehicle Routing and Scheduling with Time Windows", *Les Cahiers du GERAD*, 0709-986X, Paper 83-26.
- Desrosiers, J., F. Soumis, M. Desrochers and M. Sauve (1986), "Methods for Routing with Time Windows", *European Journal of Operational Research*, 23, pp. 236-245.

- Desrosiers, J., F. Soumis and M. Sauve (1988), "Lagrangian Relaxation Methods for Solving Minimum Fleet Size Multiple Traveling Salesman Problems with Time Windows", *Management Science*, 34, pp. 1005-1022.
- Dror, M. (1994), "Note on the Complexity of the Shortest Path Models for Column Generations in the VRPTW", *Operations Research*, 42, pp. 977-978.
- Dror, M., G. Laporte and P. Trudeau (1989), "Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks", *Transportation Science*, 23 (2), 166-176.
- Dror, M., G. Laporte and F. Louveaux (1993), "Vehicle Routing with Stochastic Demands and Restricted Failures", *ZOR, Methods and Models of Operations Research*, 37 (3), 273-283.
- Fiala Timlin, M.T. and W.R. Pulleyblank (1992), "Precedence Constrained Routing and Helicopter Scheduling: Heuristic Design", *Interfaces*, 22(3), 100-111.
- Gendreau, M., G. Laporte and R. Seguin (1995), "An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Customers", *Transportation Science*, 29 (2), 143-155.
- Gendreau, M. G. Laporte and R. Seguin (1996), "A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers", *Operations Research*, 44, 469-477.
- Graham, D. and H.L.W. Nuttle (1986), "A Comparison of Heuristics for a School Bus Scheduling Problem", *Transportation Research*, Part B, 20 (2), pp. 175-182.
- Ichoua, S., M. Gendreau, and J. Y. Potvin (2000a), "Diversion Issues in Real-Time Vehicle Dispatching", *Transportation Science*, in press.
- Ichoua, S., M. Gendreau, and J.Y. Potvin (2000b), "Vehicle Dispatching with Time-Dependent Travel Times", University of Montreal, CRT working paper.
- Jagannathan, S. (1999), "Analysis of Dynamic Dispatching Heuristics for Carrier Fleet Operations in a GIS Framework", MS Thesis, University of California, Irvine.
- Jaillet, P. (1985), "Probabilistic Traveling Salesman Problem", Ph.D. thesis, Massachusetts Institute of Technology.
- Kolen, A.W.J., A.H.G. Rinnooy Kan and H.W.J.M. Trienekens (1987), "Vehicle Routing with Time Windows", *Operations Research*, 35, pp. 266-273.
- Langevin, A., F. Soumis, and J. Desrosiers (1990), "Classification of Traveling Salesman Problem Formulations", *Operations Research Letters*, 9, p. 127-132.

- Laporte, G., F. Louveaux and H. Mercure (1992), "The Vehicle Routing Problem with Stochastic Travel Time", *Transportation Science*, 26 (3), 161-171.
- Lawler, E.L. (1963), "The Quadratic Assignment Problem", *Management Science* 19, 586-599
- Levin, A. (1971), "Scheduling and Fleet Routing Models for Transportation Systems", *Transportation Science*, 5 (3), pp. 232-255.
- Mahmassani, H.S., Kim, Y. and Jaillet, P. (2001), "Local Optimization Approaches to Solve Dynamic Commercial Fleet Management Problems", *Transportation Research Record*, forthcoming.
- Padberg, M. and T. Sung (1991), "An Analytical Comparison of Different Formulations of the Traveling Salesman Problem", *Mathematical Programming* 52, 315-357.
- Picard, J.C. and M. Queyranne (1978), "The Time Dependent Traveling Salesman Problem and its Application to the Tardiness Problem in One Machine Scheduling", *Operations Research* 26, 86-110.
- Powell, W.B. (1988), "A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem", in *Vehicle Routing: Methods and Studies* (B.L. Golden and A.A. Assad, eds.), Elsevier Science Publishers, Amsterdam, The Netherlands, pp. 249-291.
- Powell, W.B. (1996), "A Stochastic Formulation of the Dynamic Assignment Problem, with an Application to Truckload Motor Carriers", *Transportation Science*, 30 (3), 195-219.
- Powell, W.B., W. Snow and R.K. Cheung (2000), "Adaptive Labeling Algorithms for the Dynamic Assignment Problem", *Transportation Science*. 34 (1), 50-66.
- Powell, W.B., M. T. Towns, and A. Marar (2000), "On the Value of Optimal Myopic Solutions for Dynamic Routing and Scheduling Problems in the Presence of User Noncompliance", *Transportation Science*, 34 (1), 67-85.
- Powell, W.B. and T.A. Carvalho (1998), "Dynamic Control of Logistics Queuing Network for Large-Scale Fleet Management", *Transportation Science*, 32(2), 90-109
- Powell, W.B., T. Carvalho, G. Godfrey and H. Simao (1995), "Dynamic Fleet Management as a Logistic Queuing Network", *Annals of Operations Research*, 61, 165-188

- Regan, A.C., S. Jagannathan, and X. Wang (2001), "Mixed Global and Locally Optimal Heuristics for Local Truckload Trucking Operations with Strict Time-Windows", *Transportation Research Record*, forthcoming.
- Regan, A.C., H.S. Mahmassani and P. Jaillet (1996), Dynamic Dispatching Strategies Under Real-Time Information for Carrier Fleet Management, in J.B. Lesort (ed), *Transportation and Traffic Theory*, Pergamon, pp. 737-756.
- Regan, A.C., Mahmassani, H.S., and Jaillet, P. (1999), "Dynamic Fleet Management for Trucking Operations." World Market Series: Business Briefing. Global Truck and Commercial Vehicle Technology: An Analysis of the Truck and Commercial Vehicle Industry and Perspectives o the Future, pp. 31-35, World Markets Research Center.
- Swersey, A. and W.Ballard (1984), "Scheduling School Buses", *Management Science*, 30, pp. 844-853.
- Wang, X and A.C. Regan (2000a), "An Integer Programming Formulation for the Probabilistic Traveling Salesman Problem in Which Each Node Has the Same Probability of Requiring a Visit", *Operations Research Letters*, under review.
- Wang, X and A.C. Regan (2000b) A New Cutting Plane Method for Integer Programming Problems with Binary Variables, Report ITS-UCI-WP-00-12, in preparation for submission to the *European Journal of Operations Research*.
- Wang, X and A.C. Regan (2000c) Alternative Assignment Models for Time Constrained Local Fleet Assignment in which Service and Travel Times are Stochastic, Report ITS-UCI-WP-00-13, in preparation for submission to *Transportation Science*.
- Wang,X and A.C. Regan (2001a), "Local Truckload Pickup and Delivery with Hard Time Window Constraints", *Transportation Research, Part B*, in press.
- Wang, X. and A.C. Regan (2001b), "Assignment Algorithms for Local Truckload Trucking Problems with Stochastic Service Times and Time Window Constraints", Accepted for presentation at the 80th meeting of the Transportation Research Board and for Publication in *Transportation Research Record*.
- Yang, J., Jaillet, P., and Mahmassani, H.S. (1999), "On-Line Algorithms for Truck Fleet Assignment and Scheduling under Real-time Information," *Transportation Research Record* 1667, pp.107-113.
- Yang, W., K. Marthur and R. Ballou (2000), "Stochastic Vehicle Routing Problem with Restocking", *Transportation Science*, 34 (1), 99-112.