Manpower and Transportation Planning

LAI, Shu Wo

A Thesis Submitted in Partial Fulfilment

of the Requirements for the Degree of

Doctor of Philosophy

in

Systems Engineering and Engineering Management

The Chinese University of Hong Kong

September 2014

Thesis Assessment Committee

Professor Chun Hung Cheng (Chair)
Professor Janny May Yee Leung (Thesis Supervisor)
Professor Anthony Man Cho So (Committee Member)
Professor Gilbert Laporte (External Examiner)

Abstract

This thesis studies three routing and scheduling problems arising in manpower and transportation planning. These problems are rooted in real applications, and carry interesting characteristics. By exploiting the structures of the problems, this thesis provides effective mathematical models and algorithms for solving the problems practically. Managerial insights are developed via extensive computational tests and sensitivity analyses.

Effective scheduling of staff can generate considerable saving where unnecessary costs due to misallocation of staff to the demand are reduced. The second chapter of this thesis studies the Shift Rostering Problem—the assignment of staff to shifts over a planning horizon such that work rules are observed. In view of the special structure of the work-rule constraints, we model work rules in terms of prohibited meta-sequences and resource constraints. A novel graph-based formulation is proposed where the formulation size depends on the structure of the work-rule constraints and is independent of the number of staff. This is particularly beneficial when the work rules possess sufficient structure that results in a small formulation. For some cases when the canonical formulation could not solve in a reasonable time, our approach can find optimal solutions in a few minutes.

Vehicle routing problems occur frequently in the delivery and collection of items between a central depot and a number of customer locations. Motivated from the distribution of beer and malt beverages in China, the third chapter of this thesis studies a time-constrained heterogeneous vehicle routing problem on a multigraph. Parallel arcs represent Pareto-optimal paths between two locations, with various travel times and costs. We provide a mixed-integer linear programming formulation of the problem and propose a tabu search heuristic for its solution. The tabu search is designed to address the parallel arc structure on the network, which necessitates modifications of the basic search operations such as insertion. Our numerical experiments demonstrate the effectiveness of the proposed tabu search heuristic and provide further managerial insights through sensitivity analysis. The numerical experiments suggest considerable transportation cost savings attributable to

the utilization of alternative route structure and provide some insights to aid distributors on their vehicle dispatch policies.

Most transportation planning models are deterministic and do not consider uncertainties in operations. Therefore, disruptions on the planned daily schedule often occurs in the daily operations due to unexpected traffic conditions, vehicle breakdowns, accidents, special events, etc. When delays due to these uncertainties accumulate and propagate in the execution and operation of the planned schedule, poor service and high operational cost result. The fourth chapter of this thesis addresses a real-time tram scheduling problem arising in a public transit company in Hong Kong. Motormen and trams are scheduled simultaneously to provide passenger transportation service in some commercial routes. To mitigate unexpected overtime and meal-break delays due to the uncertainties in operations, planned schedules are revised dynamically using real-time information under a rolling-horizon framework. Furthermore, route frequencies are maximized simultaneously for improved passenger transportation service. We provide a number of mathematical models for revising the schedules in real-time. A general event-driven simulation model is developed to evaluate the efficiency and the effectiveness of the models with real-world data.

摘要

本論文研究員工排程及車輛調度問題，優化供應鏈物流和公共交通系統。論文的第一部分研究一個員工排班問題 —— 編排員工每天的工作崗位，避免需求錯配，同時符合複雜的工作編排規定。論文的第二部分研究一個多車種車輛調度問題。有別於傳統的車輛調度問題，兩個不同地點之間有多條路徑連結，每一條路徑有特定的運送成本和需時。不但要考慮車輛到達顧客地點的次序，而且要決定連結顧客地點之間的路徑。以最少的運送成本在限時內滿足所有顧客的需求。由於在實際的運作中，隨機的行車時間會經常擾亂預定的排程。因此，論文的第三部分研究利用實時資訊即時調整員工以及車輛排程。每當重新編排的時候，車輛須調度至合適位置，確保員工有足夠車輛能夠按照編排的時間出發。本論文為上述的三個員工排程及車輛調度問題建構了不同的數學模型，並且針對其特定的結構提出了有效的算法以及解決方案。

Acknowledgements

I would like to express my deepest gratitude to my supervisor Professor Janny Leung who guided my study and gave me insightful suggestions on my research. Thanks for her support which led me to realize my goal. Without her guidance and persistent help, this thesis would not have been possible.

Besides, I would like to thank the other thesis committee members, Professor Gilbert Laporte, Professor Chun Hung Cheng and Professor Anthony So, for their support and insightful comments. Thanks to Professor Ozgun Demirag for many generous suggestions and collaborative work in the third chapter of this thesis. Furthermore, I would like to thank Hong Kong Tramways for letting us visit the company so that we were able to understand the operations in detail. My appreciation also goes to David Wong and Victor for sharing their valuable working experience in the company, and Dorbin Ng for his support in the project. In addition, thanks to Yong Hong Kuo and Terence Tou for their thoughtful comments throughout the four years of my studies in our research meetings.

I would like to thank Professor Jesper Larsen who introduced me to operations research when I was an exchange student at the Technical University of Denmark. His supervision and guidance on a scheduling project motivated my research interests and my studies as a postgraduate student in the research area. I would like to thank Professor Stefan Røpke and Professor David Pisinger who taught me the very first courses in operations research. Without their guidance and teaching, I wouldn't be able to know or appreciate operations research.

Last but not least, I would like to thank my family, Huan-Yau Lai, Mun-Yau Au and Shu-Ming Lai, for their love and support over the years. They have been a source of encouragement and happiness to me. This thesis is dedicated to them all.

This work is dedicated to my family.

Contents

Abstract **i**

Acknowledgement **iv**

1 Introduction **1**
1.1 Manpower and Transportation Planning 1
1.2 Novel Characteristics 3

2 A Graph-Based Formulation for the Shift Rostering Problem **5**
2.1 Introduction 6
2.1.1 Problem Description 8
2.1.2 Problem Definition 10
2.2 Graph-Based Formulation 13
2.2.1 Graph Representation 14
2.2.2 A Network-flow Model with Side Constraints 21
2.3 Computational Result 26
2.3.1 Work Rules 27
2.3.2 Small Instances 30
2.3.3 Large Instances 30
2.3.4 Random Prohibited Meta-sequences 34
2.4 Conclusion 35

3 A Tabu Search Heuristic for the Heterogeneous Vehicle Routing Problem on a Multigraph **37**
3.1 Introduction and Practical Motivation 37
3.2 Literature Review 41
3.3 Problem Formulation 44
3.4 Solution Methodology 46

3.4.1 Penalized Objective Function 47
3.4.2 Initial Solution 48
3.4.3 Neighborhood Structure 48
3.4.4 Vertex Sequencing and Arc Selection 49
3.5 Numerical Analysis 52
3.5.1 Experimental Setting 52
3.5.2 Performance of the Solution Approaches 53
3.5.3 Sensitivity Analysis and Managerial Insights 55
3.6 Concluding Remarks 63

4 Real-Time Tram Scheduling 65
4.1 Introduction 65
4.1.1 Literature Review 67
4.1.2 Description of the Motivating Problem 69
4.2 Mathematical Models 75
4.2.1 Parameters 76
4.2.2 Formulation 77
4.2.3 Tram Availability Considerations 78
4.2.4 Multiple-Period Demand 79
4.3 Preliminary Results 79
4.3.1 A Simulation Model for Evaluation 80
4.3.2 Efficiency 82
4.3.3 Effectiveness 83
4.4 Conclusion 84

5 Summary 86

Bibliography 89

List of Figures

2.1 The underlying graph for the example problem 14
2.2 The graph for the example problem that considers 6 days and 4 shifts per day. 15
2.3 The optimal flow for the example problem 22
2.4 The graph before simplifications 24
2.5 An example *merge* operation 24
2.6 The graph after merging 25
2.7 An example contract operation 25
2.8 The graph after mergings and contractions 26
2.9 The size of the underlying graph 33
2.10 The time used in solving (SRP2) (left-hand-side) and its LP-relaxation (right-hand-side) 33
2.11 The size of the underlying graph 35

3.1 Arc Selection for a Fixed Sequence of Vertices 50
3.2 Insertion operation in a simple graph (left-hand-side) and in a multigraph (right-hand-side). 51
3.3 Optimality gap on small-sized instances 54
3.4 Performance of TS and IH under different computational time limits 56
3.5 Distribution of customers 56
3.6 The impact of the route duration limit, vehicle capacities and customer distribution on transportation costs 57
3.7 Effects of considering alternative routes in Example 1. (Left) Simple graph with only less costly arcs, (Right) Multigraph . . 58
3.8 Transportation cost savings due to parallel arcs 59
3.9 The percentage of more costly (and shorter duration) arcs traversed by vehicle type 60

3.10 The percentage of remote customers served by the small capacity vehicles 60
3.11 Time utilization by vehicle type 61
3.12 Percentage of remote customers and more costly arcs in the routes of small capacity vehicles under different customer distributions 62
3.13 Capacity utilization by vehicle type under different customer distributions 62
3.14 Time utilization by vehicle type under different customer distributions 63

4.1 Standard commercial routes of the company 66
4.2 A real-time decision support system 67
4.3 Travel-Time Distribution 73
4.4 The coverage of a particular route on a day 75

List of Tables

2.1 An example cover table . . . 8
2.2 An example roster . . . 10
2.3 Example work rules . . . 12
2.4 An optimal solution to the example problem . . . 22
2.5 Solving (SRP1) and (SRP2) on small instances . . . 31
2.6 Solving (SRP2) on large instances . . . 32
2.7 The time(s) used in solving (SRP2) with different staffing ratios . . . 34
2.8 Solving (SRP2) on instances with random meta-sequences . . . 35
2.9 The solution value to (SRP2) and the time(s) used in solving (SRP2) . . . 36

3.1 Parameter values . . . 53
3.2 Parameter values for the tabu search heuristic . . . 54
3.3 Performance of MIP, IH and TS on medium-sized instances . . . 55
3.4 HVRP solutions in Example 1 . . . 59

4.1 Commercial routes and deadheads . . . 70
4.2 The number of trams required for all groups and time periods . . . 71
4.3 Motorman Duties . . . 72
4.4 The mean travel times in each period . . . 73
4.5 The standard deviation of the travel times in each period . . . 74
4.6 Realized schedule for one motorman . . . 75
4.7 Summary of the edge attributes . . . 77
4.8 Summary of the methods . . . 80
4.9 User-defined parameters for the formulations . . . 82
4.10 CPU times . . . 83
4.11 Performance about motorman schedules . . . 84
4.12 The worst-case for all motormen . . . 84

4.13 Passenger transportation service 84

Chapter 1

Introduction

This thesis investigates the scheduling of vehicles and drivers to provide reliable and cost-effective transportation services. We focus on three routing and scheduling problems arising in manpower and transportation planning. These problems are rooted in real applications, and encompass interesting characteristics. By exploiting the structures of the problems, this thesis provides effective mathematical models and algorithms for solving the problems practically. Managerial insights are developed via extensive computational tests and sensitivity analyses.

This chapter gives a brief introduction to the three problems covered in this thesis. Section 1.1 describes the typical planning processes of manpower and transportation planning. Section 1.2 describes the characteristics that motivate the specific problems.

1.1 Manpower and Transportation Planning

In this section, we briefly describe the planning processes of a public transit company in Hong Kong as an example to illustrate some typical considerations. A review on existing literature about public transportation planning can be found in Desaulniers and Hickman (2007).

Public transportation planning processes are typically divided into four phases: strategy planning, tactical planning, operational planning and real-time control. This thesis focuses the scheduling processes in operational planning and real-time control. As a tradition, we describe the planning processes we encountered as inter-related problems in a sequential way: timetabling, vehicle scheduling, shift scheduling, shift rostering and real-time control.

Timetabling

The *timetabling* process determines the required transportation services during the planning horizon. Based on the estimated origin-destination passengers demand, the objective is to maximize passenger service levels and minimize operational costs, with limited number of vehicles and drivers. The scheduling is constrained by the available fleet size and some operational requirements.

Vehicle Scheduling

The *vehicle scheduling* process focuses on the scheduling of vehicles to meet the demand requirements specified in the timetable. Deadheads are also introduced to connect the commercial routes. The goal of vehicle scheduling is to minimize the operational costs including the number of vehicles and the total distance travelled. There are some operational requirements governing the vehicle routing and scheduling. For example, the maximum capacity, a fixed depot, the maximum travel distance, etc. This operational requirements can usually be formulated as resource constraints.

Shift Scheduling

With the predetermined vehicle schedules, drivers are then assigned to operate these vehicles for the trips and deadheads under some work rules. Moreover, meal breaks and rests have to be assigned to drivers appropriately. The problem is usually known as *shift scheduling*, where the goal is to determine the daily duties (sequence of tasks to be performed) that minimize the number of drivers. Duties of similar working hours and rest periods are categorised into shift types, like Morning Shifts, Night Shifts, etc. The solution shows how many drivers are needed for each shift type in a day.

Shift Rostering

A roster is constructed to cover the shift-based demand as much as possible. When drivers are assigned to shifts, work rules are handled simultaneously for workload balancing, staff regulations, day-offs assignments, etc. The goal of this *rostering* process is to cover the demand as much as possible, minimizing the number of staff needed, and maximizing staff preferences.

Real-Time Control

Disruptions on planned schedules occur occasionally due to unexpected traffic conditions, bad weather, accidents, etc. Delays due to these uncertainties in travel time may accumulate and propagate in the execution, resulting in poor services and high operational costs. In real-time control, disrupted schedules are repaired. To hedge and mitigate these risks, schedules are modified continuously to avoid serious delays and to improve service.

1.2 Novel Characteristics

This thesis focuses on routing and scheduling problems that have some novel characteristics. Considerable savings and improved services can be realised when these characteristics are incorporated into the transportation planning and control processes. This section describes the characteristics that motivate the study of the specific problems covered in this thesis.

Modeling Staff Regulations

Human resource is a scarce resource, difficult to manage and often involves expensive training costs. Effective scheduling of staff can generate considerable saving where unnecessary costs due to mis-allocation of staff to the demand are reduced. Personnel schedules have to be not only cost-effective, but also applicable in the human context. Thus, numerous work rules must be followed when assigning shifts to employees. Since the introduction of personnel scheduling problems in 1950s, it remains an interesting question on how these complicated work rules could be formulated generally and handled efficiently. The second chapter of this thesis presents a novel way of modeling work rules. The effectiveness of the proposed approach is demonstrated on solving a shift rostering problem. Handling work rules successfully for the problem would give insights for solving more complicated versions.

Alternative Route Considerations

The road network underlying a distribution system presents multiple travel options for vehicles. For example, a vehicle going from one customer location to another may consider different paths of travel based on criteria such as travel time, cost and distance. These alternative routes are typically not considered in the analysis of vehicle routing problems which are often studied on a simple graph. A multigraph structure, however, would enable the operators to build vehicle routes by utilizing the parallel arcs between each pair of customer locations which can help them address realistic trade-offs such as transportation costs and delivery time. In the third chapter of this thesis, we study a time-constrained heterogeneous vehicle routing problem on a multigraph. The problem is motivated from the distribution of beer and malt beverages in China, with some characteristics including the possibility of alternative paths of travel under the prevalence of road toll charges, fleet heterogeneity, and time-restricted delivery.

Uncertainties

Most existing formulations for transportation planning consider deterministic travel times. However, travel times may not be realized as what are expected, due to the weather conditions, special events, traffic conditions, etc. Delays due to these uncertainties in operations may accumulate and propagate in the execution of the planned schedule, resulting in poor services and high operational costs. With the advance in RFID technologies and database management systems, the locations of trams can be determined instantaneously at a relatively low cost nowadays. The fourth chapter of this thesis investigates how real-time information can be utilized in combination with historical data to improve the controllers' real-time routing and scheduling decisions practically.

Chapter 2

A Graph-Based Formulation for the Shift Rostering Problem

This chapter investigates a shift-rostering problem (SRP)—the assignment of staff to shifts over a planning horizon such that work rules are observed. We formulate the work rules in terms of *prohibited meta-sequences* and *resource constraints*. This framework provides much flexibility in modeling work rules in a continually changing environment. Canonical formulations could not solve the problem effectively when there are a large number of staff and there are a large number of feasible shift patterns. We proposed a graph-based formulation where the set of feasible shift patterns are represented by the paths of a graph. The formulation size depends on the structure of the work-rule constraints and is independent of the number of staff. This is particularly beneficial when the work rules possess sufficient structure that results in a small graph-based formulation. Moreover, the formulation often yields multiple optimal solutions which are beneficial for managerial decisions in practice. We have conducted computational tests on randomly-generated instances with work rules drawn from practice. Our results indicate that for some cases when the canonical formulation could not solve the problem in a reasonable time, our approach can find optimal solutions in a few minutes.

2.1 Introduction

Personnel scheduling has been useful in hospital nurse-scheduling, call center operations, airlines, urban transportation, supply chain industries, etc. Human resource is a scarce resource, difficult to manage and often involves expensive training cost. Effective scheduling of staff can generate considerable saving where unnecessary costs due to mis-allocation of staff to the demand are reduced. In addition to cost concerns, a well-planned roster can also improve staff satisfaction of a company, which is important but often overlooked. Other potential benefits include easier recruitment, higher level of services, better morale, improved productivity and decreases in turnover, absenteeism, overtime, requests for days off and tardiness.

Personnel schedules have to be not only cost-effective, but also applicable in the human context. Thus, numerous work rules must be followed when assigning shifts to employees. Since the introduction of personnel scheduling problems in 1950s, it remains an interesting question on how these complicated work rules could be formulated generally and handled efficiently. This chapter presents a novel modeling approach addressing this issue. The effectiveness of the proposed approach is demonstrated on solving a shift rostering problem. Handling work rules successfully for the problem would give insights for solving more complicated versions.

As early as 1954, Dantzig (1954) studied staff scheduling at toll-booths. Recent literature surveys on staff scheduling are given by Cheang et al. (2003), Burke et al. (2004) and Ernst et al. (2004). Mathematical modeling is useful for exact methods based on mathematical programming techniques as well as heuristics that generate good feasible solutions efficiently. In the following paragraphs, we present a brief review of mathematical models that have been useful for solving personnel scheduling problems.

Dantzig (1954) formulated a shift-pattern model for staff scheduling at toll-booths in which feasible shift patterns are decision variables. For small instances or when the work rules are extremely restrictive, it is possible to explicitly enumerate all the feasible shift patterns that can be assigned to the staff. The LP-relaxation of this model can be solved via a column generation technique (see Dantzig and Wolfe (1960)) in which a subset of feasible shift patterns is obtained by solving pricing subproblems. Since then, a number of column-generation-based exact and heuristic approaches for shift scheduling have been developed; see, for example, Gamache et al. (1999), Sarin and Aggarwal (2001) and Caprara et al. (2003). Since there are many feasible patterns in most operational settings, various methods were developed for generating a subset of patterns, in order to reduce the number of variables to a manageable level. In many cases, these approaches were able to produce near-optimal solutions in a reasonable time.

Branch-and-cut approaches have been effective for solving large integer programs. Decision variables can be used to represent assignments of shifts to individual staff. The effectiveness relies on the choice of valid inequalities as well as the efficiency of the separation algorithm. Felici and Gentile (2004) derived a number of facet-defining valid inequalities for a staff scheduling problem. Special branching rules are applied to break symmetries to improve the efficiency of the branch-and-bound process. Cappanera and Gallo (2004) proposed a 0-1 multicommodity flow problem where the work rules are handled as side constraints. Valid inequalities are derived to tighten the LP-relaxation of the formulation. When compared with the pattern formulation, these integer programs provide more flexibility in modeling work rules for individual employee, but less applicable when there are a large number of homogeneous employees.

Network flow models are widely used in formulating and solving many practical problems, see Ahuja et al. (1993). Balakrishnan and Wong (1990) formulated a network flow model for cyclic staff scheduling. The network can be constructed easily when shift-type changes (*on* to *off* or vice versa) within a *stretch* (period of consecutive days) are not allowed. Millar and Kiragu (1998) extended the work of Balakrishnan and Wong (1990) to the acyclic case when a stretch is specified in terms of a *stint* (shift sequence with specified start date, length and cost). Other network flow models of related problems can be found in Segal (1974), Nicoletti (1975), Caprara et al. (1997), Caprara et al. (1998), Moz and Pato (2004) and Steinzen et al. (2010). Underlying graphs are often used to handle work rules of very special structures. Other work rules that cannot be incorporated were usually left as side constraints. Although network flow models have been used for solving staff scheduling problems for decades, it remains an interesting question on how complicated work rules could be handled using the underlying graph effectively.

The purpose of this chapter is to present a novel modeling approach for a shift rostering problem. The proposed approach consists of four steps. First of all, work rules are written in terms of prohibited meta-sequences and resource constraints. Work rules such as workload balancing and day-off assignments can be naturally formulated in this framework. An efficient algorithm is then used to construct an underlying graph representing all the feasible patterns. By solving a network flow model with side constraints, an optimal flow is obtained. Since all the work rules are handled using the underlying graph, the optimal flow represents many optimal solutions. By disaggregating the optimal flow into paths and transforming the paths into patterns, a preferred optimal solution is selected. The proposed approach produces a large number of optimal solutions, which affords flexibility in considering other managerial concerns when deciding on the roster.

We investigate a *shift rostering problem* (SRP) where the work rules can be formulated in terms of prohibited meta-sequences and resource constraints. In the following subsections, we describes the problem using a small example and then present a mixed-integer programming formulation.

2.1.1 Problem Description

Within the manpower-planning process, the Shift Rostering Problem (SRP) focusses on the stage where staff are assigned to shifts over a planning horizon. We assume that the demand requirements for the planning horizon has been determined, the work-day has been divided into a fixed number of shifts and the workload for each shift in the planning horizon has been set.

Table 2.1: An example cover table

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Shift 1 (morning)	39	83	30	33	36	66
Shift 2 (afternoon)	21	11	34	17	31	20
Shift 3 (night)	40	6	36	50	33	14
Shift 4 (day-off)	0	0	0	0	0	0

Consider a small example with a planning horizon of 6 days and each day has 4 shifts. The demand (in number of staff required) of the shifts in the planning horizon is given in a *cover table* as illustrated in Table 2.1. Each staff is assigned to exactly one shift per day. The objective is to schedule 150 staff to cover the demand as much as possible such that all work rules are observed. Work rules vary in practice by industries and by organisations. They may include industrial regulations, staff contract conditions, etc. We consider a work rule as conditions that disallow the assignment of some shift patterns to the staff. In particular, all work rules are handled as hard constraints. Below are some example work rules.

At most 5 night shifts can be assigned in the planning horizon.

At least 2 night shifts should be assigned in the planning horizon.

At most 6 working shifts can be assigned in consecutive days.

No more than one day off in every 5 days, i.e.,
working shifts should be assigned in at least 4 consecutive days.

A day-off should be assigned after two consecutive night shifts.

No night-dayoff-night shift sequence.

Repeating shift pattern of 5 days on and then 2 days off.

Minimum rest time between working shifts (e.g. no morning shift after a night shift).

The resulting solution is a *roster* showing how many staff will be working on each shift during the planning horizon. A sequence of shift-assignments for the planning horizon is a *shift pattern*. The *roster* to *cover* the demand can be represented at an aggregate level by a set of *feasible shift patterns* (that satisfy the work rules) together with the number of staff assigned. Table 2.2 shows a roster for the example problem. The first row of the roster shows that 36 staff is scheduled to work shift 4 in day 1, shift 1 in day 2, shift 3 in day 3, shift 4 in day 4, shift 1 in days 5 and 6; that is, they are assigned the shift pattern (4, 1, 3, 4, 1, 1). This roster uses 8 shift patterns, each is represented by a row of the table, with the leftmost column indicating the number of staff that is assigned to the shift pattern. A roster is said to be feasible when every staff is assigned a feasible shift pattern.

The roster should provide sufficient staff to cover as much of the demand as possible. In our approach, we do not require the staffing level for a shift to match demand exactly. If the staff available do not meet the required level of cover, *under-cover* occurs and over-time or temporary staff may be an option. If the staff level available exceeds the required cover, *over-cover* occurs and extra off-line activities, like training or project work, may be an option. The penalty cost of an over-cover and an under-cover can be interpreted as the

Table 2.2: An example roster

	Feasible shift pattern					
No. of staff	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
36	4	1	3	4	1	1
33	1	1	4	1	3	4
30	3	4	1	3	4	1
11	2	2	4	2	2	4
14	4	1	2	4	2	3
10	2	4	2	3	4	2
10	3	4	2	3	4	2
6	1	3	4	2	2	4

per-shift average cost of a permanent staff and a temporary staff respectively. In this chapter, we schedule a given number of *homogeneous* staff (all staff are subject to the same set of work rules). The objective of SRP is to find a feasible roster that minimizes the total cost of under-cover and over-cover.

2.1.2 Problem Definition

Let $\mathcal{I} = \{1, 2, ..., I\}$ index the set of shifts, $\mathcal{J} = \{1, 2, ..., J\}$ index the set of days and $\mathcal{K} = \{1, 2, ..., K\}$ index the staff available. For each shift $i \in \mathcal{I}$ and each day $j \in \mathcal{J}$, we define *over-cover* o_{ij} as the number of excessive staff assigned to cover the demand and *under-cover* u_{ij} as the number of extra staff required to cover the demand. Under-cover or over-cover is allowed at a penalty cost $\alpha_{ij} \in \mathbb{R}^+$ and $\beta_{ij} \in \mathbb{R}^+$ respectively for each shift $i \in \mathcal{I}$ in each day $j \in \mathcal{J}$. Let

$$x_{ij}^k = \begin{cases} 1, & \text{if staff } k \in \mathcal{K} \text{ is assigned to cover shift } i \in \mathcal{I} \text{ in day } j \in \mathcal{J}; \\ 0, & \text{otherwise.} \end{cases}$$

Given the demand (in number of staff required) $d_{ij} \in \mathbb{Z}^+, \forall i \in \mathcal{I}, j \in \mathcal{J}$, the objective is to minimize the sum of under-cover and over-cover penalties.

$$(SRP1): \; min \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} (\alpha_{ij} \; u_{ij} + \beta_{ij} \; o_{ij}) \tag{2.1}$$

$$s.t. \sum_{k \in \mathcal{K}} x_{ij}^k + u_{ij} - o_{ij} = d_{ij}, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{2.2}$$

$$u_{ij}, o_{ij} \geq 0, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{2.3}$$

$$\sum_{i \in \mathcal{I}} x_{ij}^k = 1, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \tag{2.4}$$

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} w_{ij}^{rk} x_{ij}^k \leq W_r^k, \qquad \forall r \in \mathcal{R}^k, k \in \mathcal{K}, \tag{2.5}$$

$$\sum_{l=1}^{L_m} \sum_{i \in S_m(l)} x_{i,j+l}^k \leq L_m - 1, \; \forall j \in \{0, ..., J - L_m\}, m \in \mathcal{S}^k, k \in \mathcal{K}, \tag{2.6}$$

$$x_{ij}^k \in \{0, 1\}, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}. \tag{2.7}$$

As the demand (in number of staff d_{ij}) is an integer, the under-cover u_{ij} and over-cover o_{ij} are guaranteed to be integer-valued as well, so only non-negativity restrictions (2.3) are needed. A shift can either be a working shift or a day-off shift. Exactly one shift in a day should be assigned to each staff according to (2.4). We consider work rules that can be represented by resource constraints and prohibited meta-sequences. The work rule requirements are formulated by (2.5) and (2.6), and are described below.

Some work rules can be easily formulated as resource constraints. Let $\mathcal{R}^k$ be the set of resources for staff $k \in \mathcal{K}$. For each resource $r \in \mathcal{R}^k$ of staff $k \in \mathcal{K}$, let $W_r^k \in \mathbb{Z}^+$ be the resource capacity and let $w_{ij}^{rk} \in \mathbb{Z}, 0 \leq w_{ij}^{rk} \leq W_r^k$, be the amount of resource r consumed when staff k is assigned to cover shift $i \in \mathcal{I}$ in day $j \in \mathcal{J}$. The resource usage for each staff must not exceed the resource capacity. The resource constraints are formulated by (2.5). For example, the work rule *"at most 2 night shifts can be assigned in total"* can be expressed by a resource constraint with capacity 2 with each night shift consuming one unit of resource and other shift types consuming none. Due to the presence of Constraints (2.4), the work rule "*at least 2 night shifts can be assigned in total*" can be viewed as "*at most 4 non-night shifts can be assigned in total*" and hence can be expressed by a resource with capacity 4 (= J-2), with each non-night shift consuming one unit of resource. Combining the two resource constraints, the work rule "*exactly two night shifts must be assigned in total*" can be implemented as well. In this way, the number of night-shift assignments can be distributed among the staff. Similarly, different workload balancing requirements can be easily achieved under this setting. For example, the work rule *"no staff should work more than 33 working hours"* can be formulated naturally using a resource constraint, even if working hours differ by shift and by day.

Although resource constraints can model many work rules, other work rules are more naturally formulated using prohibited meta-sequences. We define a *prohibited meta-sequence* m as a sequence of subsets of shifts ($S_m(1)$, $S_m(2)$, ..., $S_m(L_m)$) where $S_m(l) \subseteq \mathcal{I}$ for all $l \in \{1, 2, ..., L_m\}$. The number of components in a meta-sequence m, denoted by L_m, is referred as the length of the meta-sequence. Let $\mathcal{S}^k$ index the set of prohibited meta-sequences for staff $k \in \mathcal{K}$. A prohibited meta-sequence $m \in \mathcal{S}^k$ disallows the assignments of the shift sequences in $S_m(1) \times S_m(2) \times ... \times S_m(L_m)$ to staff k. The constraints enforcing the prohibited meta-sequences are formulated by (2.6). For example, the work rules listed in Table 2.3 can be naturally expressed by the prohibited meta-sequence as shown. The work rule *"at least 2 consecutive working shifts"* is formulated by prohibited meta-sequence $(\{4\}, \{1, 2, 3\}, \{4\})$ that disallows the shift sequences (4,1,4), (4,2,4) and (4,3,4).

Table 2.3: Example work rules

Work rules	Prohibited meta-sequences
a day-off should be assigned after a night shift.	$(\{3\}, \{1, 2, 3\})$
no night-dayoff-night shift sequence.	$(\{3\}, \{4\}, \{3\})$
at most 2 consecutive working shifts.	$(\{1, 2, 3\}, \{1, 2, 3\}, \{1, 2, 3\})$
at least 2 consecutive working shifts.	$(\{4\}, \{1, 2, 3\}, \{4\})$
at most 1 consecutive day-off.	$(\{4\}, \{4\})$

In this chapter, we study a shift rostering problem with homogeneous staff(all staff are subject to the same set of work rules). Hence, we can set

$$\mathcal{S} = \mathcal{S}^k,\ \mathcal{R} = \mathcal{R}^k,\ w_{ij}^r = w_{ij}^{rk},\ W_r = W_r^k \qquad \forall k \in \mathcal{K}.$$

The planning horizon for shift assignment and rostering is often quite long, typically horizons from several weeks up to 6 months are considered with several shifts in a day. As the number of variables and constraints of (SRP1) also increase with the number of staff, the resulting shift rostering problems are large-scale mixed-integer programs. Solving such a large-scale problem is very time consuming. When the staff are homogeneous, the resulting symmetries in assignment often further increase the solution time in a branch-and-bound framework; see Margot (2010). While recognizing that heterogeneity in staff skills is an important consideration in many shift and tour scheduling problems, we focus in this chapter on the setting where staff can be considered identical. In the next section, we propose a graph-based formulation that takes advantage of this homogeneity.

Consider a small example problem that consists of 6 days and 4 shifts per day, with the prohibited meta-sequences described in Table 2.3. Formulation (SRP1) consists of 4374 constraints, 3600 binary variables and 48 non-negative real variables. The problem has 120 feasible shift patterns. With the graph-based formulation, the example problem can be formulated using only 33 constraints, 28 non-negative integer variables and 48 non-negative real variables.

2.2 Graph-Based Formulation

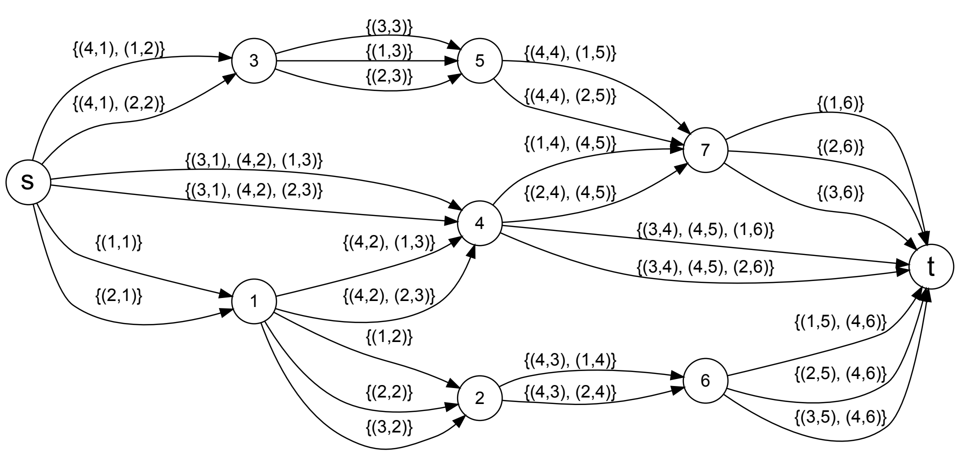
In this section, we propose a graph-based formulation where the size of the formulation depends on the structure of the work-rule constraints and is independent of the number of staff. Given a set of resource constraints and a set of prohibited meta-sequences, we construct a directed graph with a source s and a sink t so that the set of (s,t)-paths correspond to the set of feasible shift patterns. Using the graph representation of the work rules, we can reformulate the shift rostering problem as a network flow model where the demand requirements are handled as side constraints. The resulting optimal flow can be decomposed into (s,t)-paths and then transformed to an optimal solution of the shift rostering problem. If the work-rule constraints possess sufficient structures, a graph of small size may be used to represent a large set of feasible shift patterns. We have identified some of these structures and they appear in work rules that are commonly found in practical scenarios. Furthermore, as the same optimal flow can be decomposed into different sets of (s,t)-paths, multiple optimal solutions to the shift rostering problem could be obtained. This is beneficial for managerial decisions in practice.

2.2.1 Graph Representation

We construct a directed graph $G(V, E)$ with a source $s \in V$ and a sink $t \in V$ so that the set of (s, t)-paths correspond to the set of feasible shift patterns. Each edge in E is associated with a set of assignments. If assignment $(i, j) \in \mathcal{I} \times \mathcal{J}$ is included in an edge, staff assigned according to the edge will be assigned to cover shift $i \in \mathcal{I}$ in day $j \in \mathcal{J}$. The assignments along an (s, t)-path gives a feasible shift pattern that can be assigned to any of the K staff. A K-flow in the graph therefore corresponds to a feasible solution to the SRP.

Figure 2.1 illustrates the type of graph we look for and it represents the feasible shift patterns of the small example problem introduced in Section 2.1. The edge incident from the source s to vertex 3 associated with $\{(4, 1), (1, 2)\}$ corresponds to the assignment of shift 4 in day 1 and shift 1 in day 2. Consider a path that passes through vertices $(s, 3, 5, 7, t)$ where the edges are associated with $\{(4, 1), (1, 2)\}$, $\{(3, 3)\}$, $\{(4, 4), (1, 5)\}$ and $\{(1, 6)\}$ respectively, the path corresponds to the shift pattern (4, 1, 3, 4, 1, 1). The shift pattern is feasible because it satisfies all the six work rules in the example problem. Note that any (s, t)-path in the graph in Figure 2.1 corresponds to a feasible shift pattern for our example. All the 120 feasible shift patterns for our example are represented by the graph that consists of only 9 vertices and 28 edges. The example demonstrates the potential of using a graph of small size to represent a large number of feasible shift patterns.

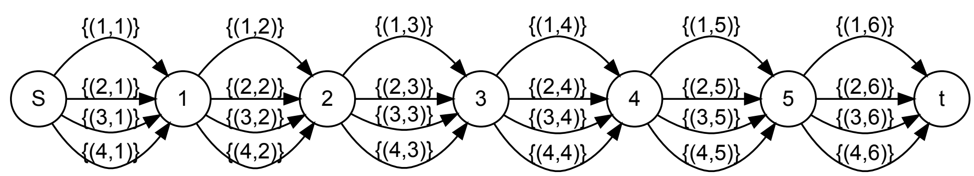
Figure 2.1: The underlying graph for the example problem



Shift Patterns

To construct a graph representing the set of feasible shift patterns, we first describe a graph $\mathcal{G}$ that represents the set of all shift patterns(that could be feasible or infeasible). Let $\mathcal{G}(\mathcal{N}, \mathcal{A})$ be a directed graph with vertex set $\mathcal{N} = \{\mathcal{N}_0, \mathcal{N}_1, ..., \mathcal{N}_J\}$, where $\mathcal{J} = \{1, 2, ..., J\}$ is the set of days in the planning horizon. Vertex $s = \mathcal{N}_0$ is the source and vertex $t = \mathcal{N}_J$ is the sink. For shift i on day j, we introduce an edge (i, j) in $\mathcal{A}$ incident from vertex $\mathcal{N}_{j-1}$ to vertex $\mathcal{N}_j$. A path from $s = \mathcal{N}_0$ to $t = \mathcal{N}_J$ represents a shift pattern with exactly one shift assigned in each day, starting from day 1 to day J. Figure 2.2 shows the graph for the example problem that considers 6 days and 4 shifts per day.

Figure 2.2: The graph for the example problem that considers 6 days and 4 shifts per day.



Among the shift patterns represented by the graph, some may violate work rules. To eliminate these infeasible ones, an algorithm to construct the graph representation is proposed that effectively splits the vertices of $\mathcal{G}$, so that the resulting graph represents only the feasible shift patterns. The algorithm relies on an efficient way to determine the feasibility of an (s, t)-path with respect to the resource constraints and the prohibited meta-sequences. We will first describe how we determine the feasibility for one (s, t)-path and then describe the way we construct the graph.

For every edge e, let $\sigma(e) \in \mathcal{I} \times \mathcal{J}$ denote the shift assignments associated with edge e. A *partial path* in $\mathcal{G}$ is a sequence of edges forming a directed path that starts from source s. For any partial path $P = (p_1, p_2, ..., p_n)$, it is said to be *extended along* edge e *to* partial path Q if $Q = (p_1, p_2, ..., p_n, e)$.

Resource Constraints

We may determine the feasibility of an (s, t)-path on graph $\mathcal{G}$ with respect to the resource constraints as follows. For every edge $e \in \mathcal{A}$, the *resource* *consumption* of resource r on edge e is defined as

$$w_r(e) = \{w_{ij}^r : (i,j) = \sigma(e)\} \qquad \forall r \in \mathcal{R}. \tag{2.8}$$

The *resource usage* of a partial path P on graph $\mathcal{G}$ is defined as

$$c_r(P) = \sum_{e \in P} w_r(e), \qquad \forall r \in \mathcal{R}. \tag{2.9}$$

As the resource consumptions on all the edges are non-negative and additive, for any partial path $P = (p_1, p_2, ..., p_n)$ that is extended along edge e, the resource usage of the extended partial path $Q = (p_1, p_2, ..., p_n, e)$ can be computed from the resource usage of P and the resource consumption on edge e as follows.

$$c_r(Q) = c_r(P) + w_r(e), \qquad \forall r \in \mathcal{R}. \tag{2.10}$$

The feasibility of a (s,t)-path with respect to the resource constraints can therefore be determined from the resource usages of its partial paths which can be computed iteratively using (2.10).

Prohibited Meta-sequences

We may determine the feasibility of an (s,t)-path on graph $\mathcal{G}$ with respect to the prohibited meta-sequences as follows. For every edge $e \in \mathcal{A}$ that is associated with assignment $(i,j) \in \mathcal{I} \times \mathcal{J}$, let $\sigma_{\mathcal{I}}(e) = i$ be the shift associated with edge e. For any partial path $P = (p_1, p_2, ..., p_n)$ and meta-sequence $m =(S(1),S(2),...,S(L))$ where $S(i) \subseteq \mathcal{I}$ for all $i \in \{1, 2, ..., L\}$, partial path P is said to *end with* the meta-sequence m if

1. $L \leq n$ and
2. $\sigma_{\mathcal{I}}(p_{n-L+i}) \in S(i), \ \forall i \in \{1, 2, ..., L\}$.

Note that a partial path does not end with meta-sequences that are longer than the path. For any partial path P in graph $\mathcal{G}$ and any prohibited meta-sequence $m =(S_m(1),S_m(2),...,S_m(L_m))$, and for $l \in \{1, 2, ..., L_m\}$, we define

$$h_l^m(P) = \begin{cases} 1, & \text{if } P \text{ ends with } (S_m(1), S_m(2), ..., S_m(l)); \\ 0, & \text{otherwise.} \end{cases} \tag{2.11}$$

Essentially, $h_l^m(P)$ indicates if the last l edges of path P matches the first l elements of meta-sequence m. It follows that an (s,t)-path corresponds to a shift pattern that satisfies prohibited meta-sequence $m \in \mathcal{S}$ if and only if the path consists of no partial path P where

$$h_{L_m}^m(P) = 1. \tag{2.12}$$

For any partial path P on graph $\mathcal{G}$ and prohibited meta-sequence $m \in \mathcal{S}$, we define the vector

$$H_m(P) = (h_1^m(P), h_2^m(P), ..., h_{L_m}^m(P))$$

as the *end-with vector* of P with respect to m, to track the overlap of P to the prohibited meta-sequence m. The feasibility of an (s,t)-path to the prohibited meta-sequence m can be determined by the corresponding end-with vectors of its partial paths.

Path Extension

The end-with vector when a partial path is extended can be computed easily as follows. For $m \in \mathcal{S}$ and $e \in \mathcal{A}$, let $b_m(e) = (b_1^m(e), b_2^m(e), ..., b_{L_m}^m(e))$ be a binary vector where

$$b_l^m(e) = \begin{cases} 1, & \text{if } \sigma_{\mathcal{I}}(e) \in S_m(l); \\ 0, & \text{otherwise,} \end{cases} \qquad \forall l \in \{1, 2, ..., L_m\}, \tag{2.13}$$

which we call the *inclusion vector* of edge e with respect to meta-sequence m. For any partial path P on graph $\mathcal{G}$, we set

$$h_0^m(P) = 1, \qquad \forall m \in \mathcal{S}. \tag{2.14}$$

Then, for any partial path $P = (p_1, p_2, ..., p_n)$ that is extended along edge e, the end-with vector of the extended partial path $Q = (p_1, p_2, ..., p_n, e)$ with respect to meta-sequence m can be computed as follows.

$$h_l^m(Q) = h_{l-1}^m(P) b_l^m(e), \qquad \forall l \in \{1, 2, ..., L_m\}, m \in \mathcal{S}. \tag{2.15}$$

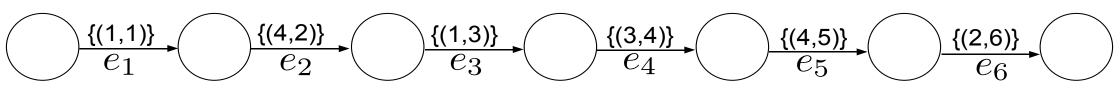
For notational simplicity, we define an operator $\star : \mathbb{B}^n \times \mathbb{B}^n \mapsto \mathbb{B}^n$ on two n-dimensional vectors x and y, as follows:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} \star \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} y_1 \\ x_1 y_2 \\ x_2 y_3 \\ \vdots \\ x_{n-1} y_n \end{pmatrix}.$$

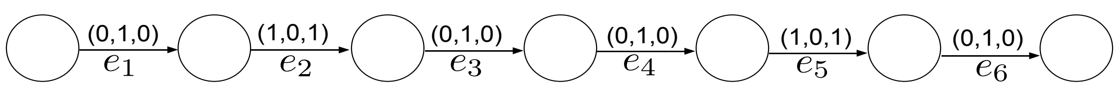
Then (2.15) can be rewritten as follows.

$$H_m(Q) = H_m(P) \star b_m(e), \qquad \forall m \in \mathcal{S}. \tag{2.16}$$

As an illustration, we consider a prohibited meta-sequence $(\{4\}, \{1,2,3\}, \{4\})$ and an (s,t)-path $(e_1, e_2, e_3, e_4, e_5, e_6)$ where the edges are associated with assignments $\{(1,1)\}$, $\{(4,2)\}$, $\{(1,3)\}$, $\{(3,4)\}$, $\{(4,5)\}$ and $\{(2,6)\}$ respectively. The path is illustrated below.

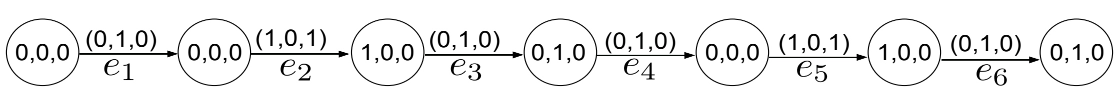


The inclusion vectors of the edges with respect to $m' = (\{4\}, \{1,2,3\}, \{4\})$ are obtained using (2.13). The inclusion vectors are indicated below on the corresponding edges.



For example, edge e_2, which assigns shift 4 in day 2, has the inclusion vector $(1,0,1)$. This is because shift 4 is in the first and the third components of $(\{4\}, \{1,2,3\}, \{4\})$, but not in the second component.

To determine the feasibility of the path, we start with an empty partial path with end-with vector $(0,0,0)$, and then extend the partial paths iteratively. The end-with vectors with respect to m of the extended partial paths are obtained using (2.16). This end-with vector of a partial path constitutes part of the "label" on the vertex the path ends on.



For example, partial path $(e_1, ..., e_4)$ has end-with vector $(0,0,0)$ with respect to meta-sequence m'. When $(e_1, ..., e_4)$ is extended along edge e_5 to partial path $(e_1, ..., e_5)$. The end-with vector of $(e_1, ..., e_5)$ with respect to m' is equal to $(0,0,0) \star (1,0,1) = (1,0,0)$, and that of $(e_1, ..., e_6)$ is $(0,1,0)$. As the last digit of the end-with vector is zero, by (2.12), the path corresponding to the shift pattern $(1,4,1,3,4,2)$ is feasible with respect to the prohibited meta-sequence $(\{4\}, \{1,2,3\}, \{4\})$.

The State of a Partial Path

A partial path P is said to be *feasible* when

$$c_r(P) \leq W_r, \quad \forall r \in \mathcal{R}, \qquad \text{and} \qquad h^m_{L_m}(P) = 0, \quad \forall m \in \mathcal{S}. \tag{2.17}$$

Otherwise, the partial path is said to be *infeasible*. It follows that an (s,t)-path corresponds to a feasible shift pattern if and only if it includes no infeasible partial path.

To determine its feasibility, we define the *state* of a partial path $P = (p_1, p_2, ..., p_n)$ as follows.

$$[d(P), \{c_r(P) : r \in \mathcal{R}\}, \{H_m(P) : m \in \mathcal{S}\}]$$

where $d(P) = n \in \mathcal{N}$ denotes the number of shift assignments along the path. If partial paths of the same state extend along the same edge, the states of the extended partial paths are identical. In our graph representation G, each vertex is associated with a state. If two partial paths have the same state, they both end at the same vertex in the graph G. Potentially, the number of states (hence vertices) could be as many as $(|J| \prod_{r \in \mathcal{R}} W_r \prod_{m \in \mathcal{S}} L_m)$ which is exponential in $|\mathcal{R}|$ and $|\mathcal{S}|$. However, since infeasible paths are pruned as the network is constructed, our computational results indicate that the network size grows only linearly with the number of meta-sequences. See Figure 2.11.

Construction Algorithm for Graph of Feasible Shift Patterns

We construct a graph $G(V,E)$ to represent the set of feasible shift patterns as follows, where the vertices in V represent states of feasible partial paths. The graph is constructed iteratively from the source $s \in \mathcal{N}$ by considering extension along an edge $(i,j) \in \mathcal{I} \times \mathcal{J}$. We start with an empty partial path with state $v_0 \in V$ where

$$\begin{aligned} d(v_0) &= 0, && \\ c_r(v_0) &= 0, && \forall r \in \mathcal{R}, \\ H_m(v_0) &= \mathbf{0}, && \forall m \in \mathcal{S}. \end{aligned}$$

At each iteration, we extend a feasible partial path along an edge (i, j) corresponding to a shift assignment for the next day. If the feasible partial paths corresponding to state $u \in V$ can be extended along edge $e \in \mathcal{A}$ to a feasible partial path of state $v \in V$, then an edge is introduced in E, incident from u to v which is associated with assignments $\sigma(e)$. The state of the extended partial path can be computed as follows.

$$\begin{aligned} d(v) &= d(u) + 1, && \\ c_r(v) &= c_r(u) + w_r(e), && \forall r \in \mathcal{R}, \\ H_m(v) &= H_m(u) \star b_m(e), && \forall m \in \mathcal{S}. \end{aligned}$$

Note that the paths can be extended along e only if $\sigma(e) = (i, j)$ with $j = d(u) + 1$. The extended partial path is feasible when (2.17) holds.

It takes $O(|\mathcal{R}| + \sum_{m \in \mathcal{S}} L_m)$ time to compute state v and to determine feasibility. Observing that feasible partial paths of the same state could be consolidated to end at the same vertex in V, a new vertex is introduced in V only when the state of the extended partial path has not been introduced in the previous iterations. We store states in V as key values that map to the states in a one-to-one correspondence. This enables individual states in V to be retrieved based on their keys in $O(|V|)$ time. There are efficient implementations of this search using associative arrays as data structures.

We define s as the source and $\{v \in V : d(v) = J\}$ as the sinks of graph G respectively. For any path in graph G, from the source to a sink, the feasibility of its partial paths are indicated by the states of the corresponding vertices along the path. Since we extend partial paths only when feasible, any path from a source to a sink corresponds to a feasible shift pattern. By searching for all extensions from all the feasible vertices, graph G represents the set of all feasible shift patterns. As there are at most $|\mathcal{I}|$ number of out-going edges for any vertex in $\mathcal{N}$, the algorithm runs with at most $|\mathcal{I}||V|$ iterations. Therefore, the algorithm runs in $O(|\mathcal{I}||V||\mathcal{R}| + |\mathcal{I}||V| \sum_{m \in \mathcal{S}} L_m + |\mathcal{I}||V|^2)$ time. The algorithm is efficient for graphs of small size. In the next subsection, we describe how the resulting graph can be simplified.

2.2.2 A Network-flow Model with Side Constraints

With the graph representation of feasible shift patterns, we reformulate the shift rostering problem as the follow network flow problem with side constraints. For all edges $e \in E$, shifts $i \in \mathcal{I}$ and days $j \in \mathcal{J}$, let

$$a_{ij}^e = \begin{cases} 1, & \text{if shift } i \text{ on day } j \text{ is associated with edge } e; \\ 0, & \text{otherwise.} \end{cases}$$

Let y_e be the flow (corresponding to the number of staff assigned) on edge $e \in E$. We want to find a K-flow on graph G minimizing the overall costs where the demand requirements are handled as side constraints.

$$(SRP2): \quad min \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} (\alpha_{ij}\ u_{ij} + \beta_{ij}\ o_{ij}) \tag{2.18}$$

$$s.t. \sum_{e \in \delta^-(v)} y_e - \sum_{e \in \delta^+(v)} y_e = 0, \qquad \forall v \in V \setminus \{s,t\}, \tag{2.19}$$

$$\sum_{e \in \delta^+(s)} y_e = K, \tag{2.20}$$

$$\sum_{e \in \delta^-(t)} y_e = K, \tag{2.21}$$

$$\sum_{e \in E} a_{ij}^e y_e + u_{ij} - o_{ij} = d_{ij}, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{2.22}$$

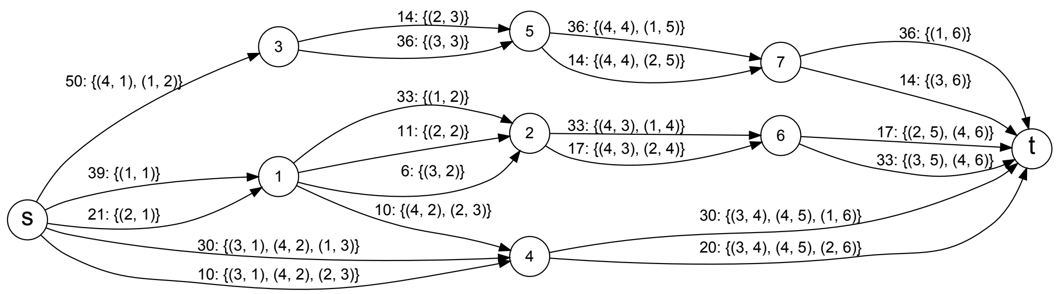
$$u_{ij}, o_{ij} \geq 0, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{2.23}$$

$$y_e \geq 0, integer, \qquad \forall e \in E. \tag{2.24}$$

Constraints (2.19) - (2.21) ensure that the solution is a K-flow on graph G. Constraints (2.22) consider the demand requirements for the shifts, where the number of staff assigned to shifts is indicated by the flow on the corresponding edge. One such solution to the example problem is illustrated in Figure 2.3, in which edges with no staff assigned are not shown. The label on an edge indicates the flow value and its associated shift assignments.

An optimal K-flow on the graph can be decomposed into K (s,t)-paths in $O(KJ)$ time where K is the number of staff and J is the number of days. The (s,t)-paths can then be transformed to K feasible shift patterns that are assigned to the K staff which is an optimal solution to the shift rostering problem. As the same K-flow could usually be decomposed into many different

Figure 2.3: The optimal flow for the example problem



sets of (s,t)-paths, the formulation $(SRP2)$ yields multiple optimal solutions to the problem which is beneficial in practice, where other considerations (e.g. seniority, staff preferences) not represented in the model can be addressed. The K-flow solution to $(SRP2)$ shown in Figure 2.3 can be decomposed into an optimal path-based solution shown in Table 2.4. Each row of the table shows an (s,t)-path that is assigned to the number of staff indicated in the left-most column. The corresponding vertices and shift pattern of the path

Table 2.4: An optimal solution to the example problem

No. of staff	Vertices of the path	Shift pattern
36	$(s, 3, 5, 7, t)$	$(4, 1, 3, 4, 1, 1)$
33	$(s, 1, 2, 6, t)$	$(1, 1, 4, 1, 3, 4)$
30	$(s, 4, t)$	$(3, 4, 1, 3, 4, 1)$
11	$(s, 1, 2, 6, t)$	$(2, 2, 4, 2, 2, 4)$
14	$(s, 3, 5, 7, t)$	$(4, 1, 2, 4, 2, 3)$
10	$(s, 1, 4, t)$	$(2, 4, 2, 3, 4, 2)$
10	$(s, 4, t)$	$(3, 4, 2, 3, 4, 2)$
6	$(s, 1, 2, 6, t)$	$(1, 3, 4, 2, 2, 4)$

are shown in column 2 and column 3 respectively. As noted, other alternate optimal path-based solutions could also be obtained. We also note that the size of the graph only depends on the work rules and is independent of the number of staff. Hence, the difficulty in solving $(SRP1)$ due to employee symmetry does not arise in the graph-based approach. When the graph G has a small size, the graph-based formulation is easy to solve. Although the graph constructed from the work rules could be large in general, work rules of practical scenarios often possess certain structure that may yield a graph of small size. Furthermore, we only need to construct the graph once. The same graph can be used when different number of staff is available, as long as the work rules have not changed.

It can be shown that prohibited meta-sequences of the following structures give a graph of small size.

1. The lengths of the prohibited meta-sequences are short;
2. All components in a prohibited meta-sequence $(S(1), S(2), ..., S(L))$ are disjoint. i.e. For all $i, j \in \{1, 2, ..., L\}$ where $i \neq j$, $S(i) \cap S(j) = \emptyset$;
3. Prohibited meta-sequences $(S(1), S(2), ..., S(L))$ with a pyramidal structure where $S(1) \subseteq S(2) \subseteq ... \subseteq S(L)$ or $S(1) \supseteq S(2) \supseteq ... \supseteq S(L)$.

These structures appear in some work rules that are commonly found in rostering problems.

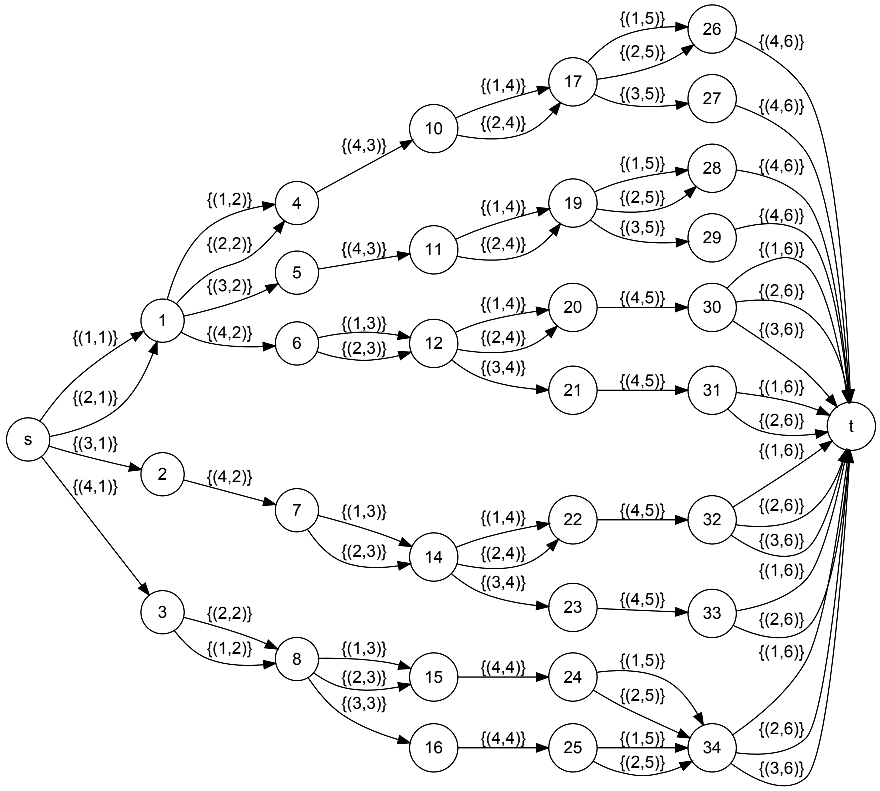
Graph Simplifications

The graph constructed for the example problem (without simplifications) is illustrated in Figure 2.4. After we have constructed the graph $G(V, E)$ representing the set of feasible shift patterns, we may simplify it to one that represents the same set of feasible shift patterns but has a smaller number of edges and vertices, by the following two operations: *merge* and *contract*.

Let edge $e \in E$ be incident from $\mathcal{T}(e) \in V$ to $\mathcal{H}(e) \in V$ and is associated with assignments $\sigma(e) \subseteq (\mathcal{I} \times \mathcal{J})$. For any vertex $v \in V$, let $\delta_{-}(v) \subseteq E$ and $\delta_{+}(v) \subseteq E$ denote the set of incoming edges and out-going edges of vertex v respectively. We note that in the graph shown in Figure 2.4, the subgraphs of path extensions are very similar on the edges near vertex t. Some of the duplications may be consolidated as shown in Figure 2.5. The subgraph shown on the left can be replaced by the subgraph shown on the right with one set of the duplicated edges $(1, 5)$ and $(2, 5)$ removed. This operation preserves the set of feasible shift patterns that the graph represents because the partial paths that end at vertex 24 and the partial paths that ends at vertex 25 would be extended along edges with the same set of shift assignments to the same vertex 34 and henceforth to the same subsequent vertices.

We achieve this kind of simplification of G using the *merge* operation. For any vertex-pair $(v_1, v_2) \in V \times V$ where $v_1 \neq v_2$, we say $\delta_{+}(v_1) \equiv \delta_{+}(v_2)$ if

Figure 2.4: The graph before simplifications



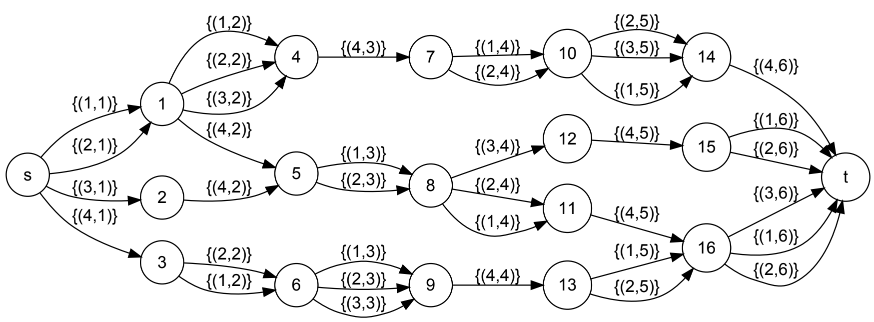
and only if for every edge $e_1 \in \delta_+(v_1)$, there is an edge $e_2 \in \delta_+(v_2)$ such that $\sigma(e_1) = \sigma(e_2), \mathcal{H}(e_1) = \mathcal{H}(e_2)$ and vice versa. If $\delta_+(v_1) \equiv \delta_+(v_2)$, then we merge v_1 and v_2 into a single vertex as follows. For all edges $e \in \delta_-(v_1)$, an edge incident from vertex $\mathcal{T}(e)$ to vertex v_2 is introduced, which is associated with assignments $\sigma(e)$. Vertex v_1 and all edges in $\delta_-(v_1)$ and $\delta_+(v_1)$ are removed. Essentially, the merge operation is applied only when partial paths ending on the two vertices would be extended along edges that are associated with the same shift assignments to the same subsequent vertices.

Figure 2.5: An example *merge* operation

Merging vertex 24 and vertex 25 of the graph on the left results in the graph shown on the right.

The merge operation always preserves the set of feasible shift patterns the graph represents. Let P_1 and P_2 be the edges that are incident to vertex v_1 and v_2 respectively. Let Q_1 and Q_2 be the edges that incident from vertex v_1 and v_2 respectively. The operation removes (s,t)-paths $(..., e_1, e_2, ...)$ for all $e_1 \in P_1$ and $e_2 \in Q_1$ (that traverses v_1) and introduces (s,t)-paths $(..., \bar{e}_1, \bar{e}_2, ...)$ (that traverses v_2) where $\bar{e}_1 = (\mathcal{T}(e_1), v_2)$ and $e_2 \in Q_2$ with $\sigma(e_1) = \sigma(\bar{e}_1), \sigma(e_2) = \sigma(\bar{e}_2)$ and $\mathcal{H}(\bar{e}_2) = \mathcal{H}(e_2)$. The operations preserve the set of feasible paths the graph represents, since the set of shift assignments associated with the paths that are removed is equivalent to the ones associated with the paths that are introduced. We apply the merge operations repeatedly until the graph cannot be further simplified. The resulting graph for the example problem at this point is illustrated in Figure 2.6.

Figure 2.6: The graph after merging



The vertices are renumbered for the sake of ease of illustration.

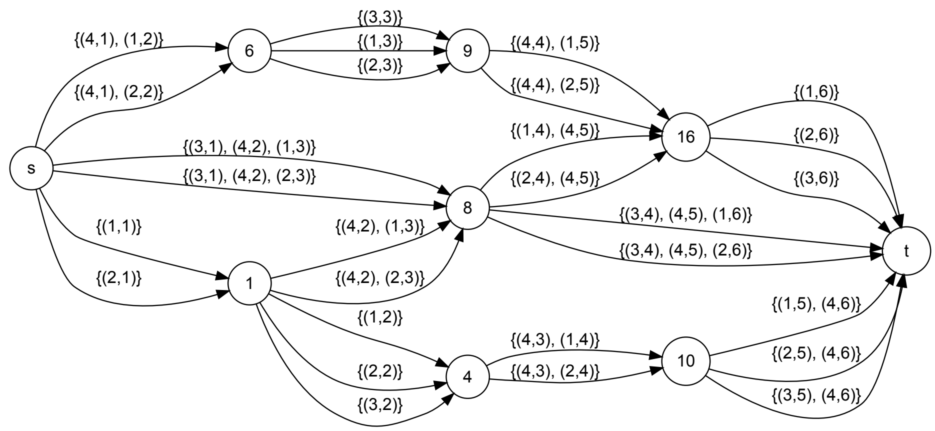
After applying the merge operations, we then simplify the graph by contracting simple paths. Figure 2.7 shows an example. We *contract* a vertex $v \in V$ as follows. For each edge-pair $(e_1, e_2) \in \delta_-(v) \times \delta_+(v)$, an edge that incidents from $\mathcal{T}(e_1)$ to $\mathcal{H}(e_2)$ is introduced, which is associated with assignments $\sigma(e_1) \cup \sigma(e_2)$. Vertex v and all edges in $\delta_-(v)$ and $\delta_+(v)$ are removed. Essentially, the sub-paths containing $(e_1, e_2) \in \delta_-(v) \times \delta_+(v)$ are removed,

Figure 2.7: An example contract operation

By contracting vertex 13 in the graph on the left results in the graph shown on the right.

and replaced by an edge e that represents the same set of shift assignments. Thus, this preserves the set of feasible shift patterns the graph represents. To ensure that the operation does not increase the number of edges, we contract a vertex $v \in V$ only when $|\delta_+(v)||\delta_-(v)| \leq |\delta_-(v)|+|\delta_+(v)|$, which holds when either $|\delta_-(v)| = 1$, or $|\delta_+(v)| = 1$, or $|\delta_+(v)|$ and $|\delta_-(v)|$ are both less than or equal to 2. The resulting graph for the example problem after merging and contracting is shown in Figure 2.8.

Figure 2.8: The graph after mergings and contractions



Since a merge operation or a contract operation would reduce the number of vertices by one, the total number of operations needed are no more than the number of vertices. The simplifications are efficient when the graph that is constructed from the work rules has a small size.

2.3 Computational Result

In this section, we computationally compare the two formulations on randomly generated instances. The work rules considered are those drawn from various industries as described in the benchmark dataset of Musliu (2006). We consider sets of work rules corresponding to 20 practical scenarios, denoted as P01, P02, ..., P19 and P20 and described in Section 2.3.1. We also evaluate other work rules which are randomly generated prohibited meta-sequences.

The demand(in number of staff required) for each shift in each day is randomly chosen from the set of values $\{0, 1, ..., 100\}$. The penalty cost of over-covers and under-covers for each shift in each day is randomly chosen from $\{1, 2, ..., 10\}$. Planning horizons of 30 days, 60 days, 90 days, 120 days, 150 days or 180 days are tested. All instances consider 10 shifts per day, among which exactly one is a day-off shift. Different shifts in a day may have different durations and/or overlap in time. For all the day-off shifts, the demands and the penalty costs are set to 0, so that staff prefer day-off to working shifts when the demand requirements are satisfied. The total number of staff to be scheduled is set as follows. Let $\hat{d}_{ij}$ be the staff requirement generated for shift $i \in \mathcal{I}$ in day $j \in \mathcal{J}$. The average number of staff required for each day is given by $l = \sum_{i \in \mathcal{I}, j \in \mathcal{J}} \frac{\hat{d}_{ij}}{|\mathcal{J}|}$. The *staffing ratio* of an instance, denoted as $\mu_2 \in \mathbb{R}^+$, is defined so that the number of staff to be scheduled is set to the nearest integer of $\mu_2 l$. A larger staffing ratio means more staff to be scheduled and a high chance of over-covers. To simulate different staffing levels, we test instances with staffing ratios of 1.2, 1.4, 1.6 and 1.8. We denote an instance with planning horizon μ_1, staffing ratio μ_2 and work rules μ_3, as Dμ_1-Sμ_2-μ_3. For example, instance D30-S1.4-P01 is a 30-day rostering problem with 1.4 staffing ratio and work-rule set P01.

All experiments were conducted on a Dell OptiPlex personal computer running Windows XP SP3 with an Intel Quad-Core processor 2.66 GHz and 3 GB of main memory. All algorithms are implemented in C++ and have been compiled using Visual Studio 2010. The solver ILOG CPLEX 12.3 with default settings were used to solve the mixed-integer programming(MIP) models (SRP1) and (SRP2) and their linear programming(LP) relaxations.

2.3.1 Work Rules

The practical work rules used in the computational tests are drawn from various industries as described in the benchmark dataset of Musliu (2006). We have rewritten the 20 different sets of practical work rules in terms of prohibited meta-sequences. Let $\mathcal{I}_M = \{1,2,3\}$, $\mathcal{I}_A = \{4,5,6\}$, $\mathcal{I}_N = \{7,8,9\}$, $\mathcal{I}_W = \{1,2,3,4,5,6,7,8,9\}$ and $\mathcal{I}_F = \{10\}$ denote the morning shifts, the afternoon shifts, the night shifts, the working shifts and the day-off shifts respectively. Moreover, let $\overline{\mathcal{I}_M} = \{4,5,6,7,8,9,10\}$, $\overline{\mathcal{I}_A} = \{1,2,3,7,8,9,10\}$ and $\overline{\mathcal{I}_N} = \{1,2,3,4,5,6,10\}$ be the non-morning shifts, the non-afternoon shifts and the non-night shifts respectively. Below are prohibited meta-sequences used in each of the 20 scenarios.

P01: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N})$, $(\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A})$, $(\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M})$, $(\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F)$, $(\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F)$, $(\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F)$, $(\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W)$, $(\mathcal{I}_W, \mathcal{I}_F, \mathcal{I}_W)$, $(\mathcal{I}_F, \mathcal{I}_F$

$, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M),$

P02: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W), (\mathcal{I}_W, \mathcal{I}_F, \mathcal{I}_W), (\mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\overline{\mathcal{I}_N}, \mathcal{I}_N, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\overline{\mathcal{I}_N}, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M),$

P03: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W), (\mathcal{I}_W, \mathcal{I}_F, \mathcal{I}_W), (\mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M),$

P04: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W), (\mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_F, \mathcal{I}_N), (\mathcal{I}_A, \mathcal{I}_F, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_F, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_F, \mathcal{I}_M),$

P05: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W), (\mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_F, \mathcal{I}_N), (\mathcal{I}_A, \mathcal{I}_F, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_F, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_F, \mathcal{I}_M),$

P06: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W), (\mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_F, \mathcal{I}_N), (\mathcal{I}_A, \mathcal{I}_F, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_F, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_F, \mathcal{I}_M),$

P07: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W), (\mathcal{I}_W, \mathcal{I}_F, \mathcal{I}_W), (\mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M),$

P08: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W), (\mathcal{I}_W, \mathcal{I}_F, \mathcal{I}_W), (\mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M),$

P09: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W), (\mathcal{I}_W, \mathcal{I}_F, \mathcal{I}_W), (\mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M),$

P10: $(\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}), (\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}), (\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_F, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_F), (\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W), (\mathcal{I}_W, \mathcal{I}_F, \mathcal{I}_W), (\mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F, \mathcal{I}_F), (\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N), (\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M), (\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A), (\mathcal{I}_N, \mathcal{I}_M), (\mathcal{I}_N, \mathcal{I}_A), (\mathcal{I}_A, \mathcal{I}_M),$

P11: ($\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}$), ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_W$, $\mathcal{I}_F$, $\mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_N, \mathcal{I}_M$), ($\mathcal{I}_N, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$),

P12: ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_W$, $\mathcal{I}_F$, $\mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$),

P13: ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_W$, $\mathcal{I}_F$, $\mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_N, \mathcal{I}_M$), ($\mathcal{I}_N, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$),

P14: ($\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}$), ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_N, \mathcal{I}_M$), ($\mathcal{I}_N, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$), ($\mathcal{I}_N$, $\mathcal{I}_F$, $\mathcal{I}_N$), ($\mathcal{I}_A$, $\mathcal{I}_F$, $\mathcal{I}_M$), ($\mathcal{I}_N$, $\mathcal{I}_F$, $\mathcal{I}_A$), ($\mathcal{I}_N$, $\mathcal{I}_F$, $\mathcal{I}_M$),

P15: ($\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}$), ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N$), ($\mathcal{I}_N, \mathcal{I}_M$), ($\mathcal{I}_N, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$), ($\mathcal{I}_N$, $\mathcal{I}_F$, $\mathcal{I}_N$), ($\mathcal{I}_A$, $\mathcal{I}_F$, $\mathcal{I}_M$), ($\mathcal{I}_N$, $\mathcal{I}_F$, $\mathcal{I}_A$), ($\mathcal{I}_N$, $\mathcal{I}_F$, $\mathcal{I}_M$),

P16: ($\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}$), ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_W$, $\mathcal{I}_F$, $\mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_N, \mathcal{I}_M$), ($\mathcal{I}_N, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$),

P17: ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_W$, $\mathcal{I}_F$, $\mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$),

P18: ($\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}$), ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_W$, $\mathcal{I}_F$, $\mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_N, \mathcal{I}_M$), ($\mathcal{I}_N, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$),

P19: ($\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}$), ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_W$, $\mathcal{I}_F$, $\mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_N, \mathcal{I}_M$), ($\mathcal{I}_N, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$),

P20: ($\overline{\mathcal{I}_N}, \mathcal{I}_N, \overline{\mathcal{I}_N}$), ($\overline{\mathcal{I}_M}, \mathcal{I}_M, \overline{\mathcal{I}_M}$), ($\overline{\mathcal{I}_A}, \mathcal{I}_A, \overline{\mathcal{I}_A}$), ($\mathcal{I}_F$, $\mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_F$, $\mathcal{I}_W, \mathcal{I}_W$, $\mathcal{I}_F$), ($\mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W, \mathcal{I}_W$), ($\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$, $\mathcal{I}_F$), ($\mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M, \mathcal{I}_M$), ($\mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A, \mathcal{I}_A$), ($\mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N, \mathcal{I}_N$), ($\mathcal{I}_N, \mathcal{I}_M$), ($\mathcal{I}_N, \mathcal{I}_A$), ($\mathcal{I}_A, \mathcal{I}_M$), ($\mathcal{I}_N$, $\mathcal{I}_F$, $\mathcal{I}_N$), ($\mathcal{I}_A$, $\mathcal{I}_F$, $\mathcal{I}_M$), ($\mathcal{I}_N$, $\mathcal{I}_F$, $\mathcal{I}_A$), ($\mathcal{I}_N$, $\mathcal{I}_F$, $\mathcal{I}_M$),

2.3.2 Small Instances

We cannot solve (SRP1) in a reasonable time, but we can solve the LP-relaxation of (SRP1) to have a sense on how difficult the problem is. We solve (SRP1) and its LP-relaxation on 20 small instances with a 30-day planning horizon, 1.4 staffing ratio and the 20 sets of practical work rules. The results are summarized in Table 2.5. Each row of the table shows the result of the instance indicated in the left-most column, including the number of variables and constraints of the formulation, the optimal solution to the LP-relaxation of (SRP1), the corresponding integrality gap and the solution time. As shown in Table 2.5, the formulation (SRP1) has hundreds of thousands of constraints and variables. The MIP formulations of this size cannot be solved efficiently. Even the LP-relaxation of (SRP1) in this size is difficult to solve. e.g. instance D30-S1.4-P02 with only 30 days requires 6 hours to solve the LP-relaxation.

We then solve the formulations using (SRP2) instead of (SRP1). The result is summarized in Table 2.5. Formulation (SRP2) has substantially fewer constraints and variables than (SPR1). As indicated in the right-most column, (SRP2) can be solved to optimality for all instances in only a few seconds. Furthermore, many optimal solutions could be obtained using (SRP2). The improvement of using (SRP2) over (SRP1) is significant on these small instances. As noted before, the size of the graph-based formulation depends on the structure of the work-rule constraints and is independent of the number of staff. In practical scenarios, the work-rule constraints usually carry some special structures that give a graph of small size. This explains why the graph-based formulation (SRP2) could solve instances which are otherwise practically impossible to solve using a canonical formulation (SRP1).

2.3.3 Large Instances

We then conduct a more extensive study of (SRP2) on instances with different planning horizons. Each instance has a planning horizon which is randomly generated with a value drawn from $\{30, 60, 90, 120, 150, 180\}$, a staffing ratio in $\{1.2, 1.4, 1.6, 1.8\}$ and a work-rule set in {P01, P02, . . . , P20}. There are 480 instances tested in total.

The average result for each planning horizon is summarized in Table 2.6. The table lists the time used in constructing the underlying graph without simplifications (*construction time*), the number of edges and vertices of the

Table 2.5: Solving (SRP1) and (SRP2) on small instances

		Canonical formulation (SRP1)						Graph-based formulation (SRP2)					
	No. of	No. of	No. of	LP-relaxation			MIP	No. of	No. of	LP-relaxation			MIP
Instance	staff	variables	constraints	Solution Value	Gap(%)	Time(s)	Time(s)	variables	constraints	Solution Value	Gap(%)	Time(s)	Time(s)
D30-S1.4-P01	648	195000	280236	223	43.1	3082.39	-	4573	966	392	0	2.88	3.67
D30-S1.4-P02	634	190800	372458	267	71.9	21417.60	-	3040	805	949	0	1.72	1.73
D30-S1.4-P03	630	189600	271830	690	38.2	3858.78	-	4735	1012	1117	0	2.92	2.95
D30-S1.4-P04	643	193500	316013	167	54.5	2112.86	-	4886	964	367	0	2.94	4.41
D30-S1.4-P05	623	187500	323014	420	18.8	3211.30	-	4883	994	517	0	2.97	3.19
D30-S1.4-P06	617	185700	318055	202	0.0	1906.13	-	5089	1035	202	0	3.14	4.45
D30-S1.4-P07	643	193500	277433	516	27.5	2811.95	-	4735	1012	712	0	2.91	3.41
D30-S1.4-P08	430	189600	255450	518	30.0	2884.67	-	4676	983	740	0	2.81	2.78
D30-S1.4-P09	603	181500	227028	832	35.5	1977.95	-	4391	940	1289	0	2.72	3.42
D30-S1.4-P10	608	183000	262348	366	41.7	2790.91	-	4735	1012	628	0	2.89	2.97
D30-S1.4-P11	652	196200	266316	841	38.0	2751.74	-	4398	920	1356	0	2.66	2.84
D30-S1.4-P12	660	198600	212820	38	15.6	776.30	-	6933	1225	45	0	4.55	8.97
D30-S1.4-P13	595	179100	226400	407	41.0	1779.63	-	4328	873	690	0	2.67	3.06
D30-S1.4-P14	623	187500	323014	341	51.8	2584.39	-	4883	994	708	0	3.02	3.14
D30-S1.4-P15	631	189900	310121	178	46.7	2755.38	-	4185	872	334	0	2.58	2.74
D30-S1.4-P16	617	185700	268078	460	42.3	3180.38	-	4440	947	797	0	2.78	3.84
D30-S1.4-P17	629	189300	186484	199	25.2	979.54	-	6999	1215	266	0	4.73	8.20
D30-S1.4-P18	640	192600	276140	793	3.6	3286.05	-	4735	1012	823	0	2.92	3.02
D30-S1.4-P19	639	192300	261012	90	77.9	1746.31	-	4398	920	408	0	2.64	4.03
D30-S1.4-P20	630	189600	309630	237	37.6	2402.38	-	4185	872	380	0	2.52	3.45

"-": no optimal solution found before the computer runs out of memory.

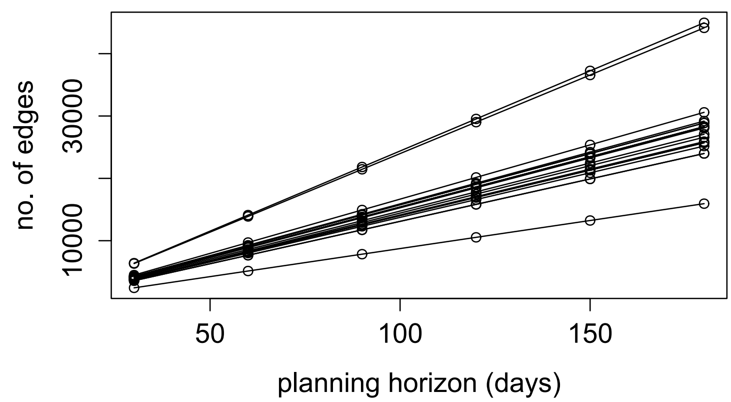
Table 2.6: Solving (SRP2) on large instances

	Planning horizon					
	30	60	90	120	150	180
Construction time(s)	0.49	1.22	1.31	1.65	2.16	2.69
No. of edges (before)	5664.15	12084.15	18504.15	24924.15	31344.15	37764.15
No. of vertices (before)	1297.10	2812.10	4327.10	5842.10	7357.10	8872.10
Simplification time(s)	0.18	1.08	3.26	7.25	13.64	22.97
No. of edges (after)	4161.35	9007.85	13854.35	18700.85	23547.35	28393.85
No. of vertices (after)	678.65	1476.65	2274.65	3072.65	3870.65	4668.65
Out-of-memory(%)	0	0	0	0	2.5	2.5
MIP time(s)	3.86	44.26	95.52	189.23	294.19	447.93
LP time(s)	2.99	12.75	28.62	50.80	80.19	114.18
No. of variables	4761.35	10207.85	15654.35	21100.85	26547.35	31993.85
No. of constraints	978.65	2076.65	3174.65	4272.65	5370.65	6468.65
Integrality gap(%)	0.00	0.00	0.00	0.00	0.00	0.00
Non-integers in LP(%)	5.03	5.53	5.69	5.82	5.87	5.85

underlying graph before simplifications, the time used in simplifying the underlying graph (*simplification time*), the number of edges and vertices of the underlying graph after simplifications, the time used in solving (SRP2) and its LP-relaxation to optimality(*MIP time* and *LP time* respectively), the number of variables and constraints of (SRP2), the integrality gap and the percentage of non-integers in the solution to the LP-relaxation. When no optimal solution to (SRP2) is found before the computer runs out of memory(instances D150-S1.6-P17, D150-S1.8-P12, D180-S1.4-P12 and D180-S1.6-P12), the "MIP time" reports the time at which the instance runs out of memory and the integrality gap does not include the result of these instances. "Out-of-memory" indicates the percentage of instances that run out-of-memory.

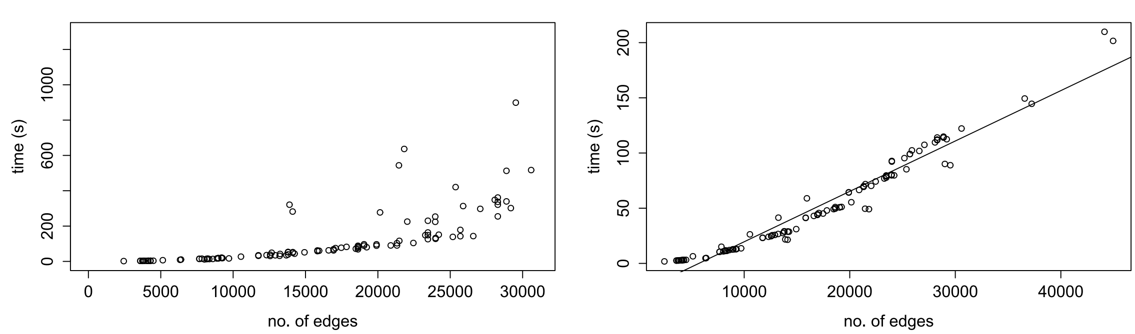
As shown in Table 2.6, the underlying graphs of (SRP2) can be constructed in a few seconds (the worst case is 2.85 minutes) which is negligible compared to the time in solving the corresponding MIP models. Figure 2.9 shows the number of edges as a function of the length of the planning horizons for all instances, with instances of the same set of work rules connected by a straight line. As shown in the figure, the graph size, the number of variables and the number of constraints of (SRP2) are also linearly increasing with the planning horizon. Furthermore, as shown in Table 2.6, the integrality gap

Figure 2.9: The size of the underlying graph



and the percentage of non-integers in the solution of the LP-relaxation are both small, which are beneficial to branch-and-bound. Moreover, as shown in Figure 2.10, the time used in solving the LP-relaxation of (SRP2) is linearly increasing with the number of edges of the underlying graph. This explains why some large rostering problems, with long planning horizons and large number of staff, could be solved readily using (SRP2). (SRP2) performs especially well for underlying graphs of small size.

Figure 2.10: The time used in solving (SRP2) (left-hand-side) and its LP-relaxation (right-hand-side)



To simulate different staffing levels, we vary the staffing ratios. Table 2.7 present the average time used in solving (SRP2) and its LP-relaxation. The staffing ratios are indicated in the rows while the planning horizons in the columns. The result shows that (SRP2) gives a steady performance under the different staffing levels.

Table 2.7: The time(s) used in solving (SRP2) with different staffing ratios

	Solving the integer model (SRP2)						Solving the LP-relaxation of (SRP2)					
	30	60	90	120	150	180	30	60	90	120	150	180
1.2	4.23	19.76	128.43	196.94	277.18	333.82	3.23	14.13	31.59	55.62	90.21	125.20
1.4	3.81	72.13	100.29	168.38	310.54	582.32	2.95	12.59	28.13	50.20	78.14	112.61
1.6	3.95	63.12	88.20	157.51	363.13	542.93	2.88	12.03	27.03	48.38	75.71	108.44
1.8	3.44	22.05	65.17	234.11	225.90	332.65	2.91	12.25	27.74	48.98	76.70	110.48

2.3.4 Random Prohibited Meta-sequences

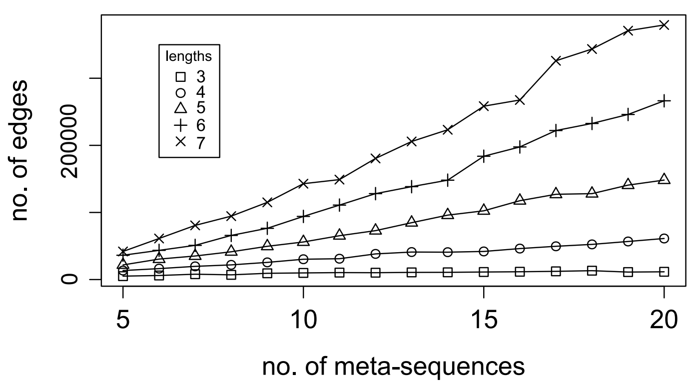
We then test (SRP2) on instances where work rules are specified by some randomly generated prohibited meta-sequences. We consider instances with 30 days and 1.4 staffing ratio, and randomly generate 5 to 20 prohibited meta-sequences of fixed length (ranging from 3 to 7). Every component in a prohibited meta-sequence consists of exactly 3 different shifts. There are 80 instances tested in total. The average results for a given length are summarised in Table 2.8. Tables 2.9 present the optimal solutions and the time used in solving the formulations for all the instances.

The results summarized in Table 2.8 are similar with the results on the practical work rules as shown in Table 2.6, except that the underlying graph become much larger even with a small number of prohibited meta-sequences. Figure 2.11 shows the number edges as a function of the number of prohibited meta-sequences. Each line in Figure 2.11 connects instances where the length of the meta-sequences are the same. The number of edges increases approximately linearly with the number of prohibited meta-sequences for a given length. This indicates that (SRP2) could scale well with the number of prohibited meta-sequences. The underlying graph is smaller with shorter prohibited meta-sequences(more restrictive work rules) and fewer prohibited meta-sequences (fewer work rules). As for the result of the practical scenarios, it is the work-rule structure that matters.

Table 2.8: Solving (SRP2) on instances with random meta-sequences

	The length of the meta-sequences				
	3	4	5	6	7
Construction time(s)	2.27	11.69	6.26	11.97	20.46
No. of edges (before)	10392.88	39149.94	90228.75	158926.88	239108.62
No. of vertices (before)	1714.50	5226.00	10920.25	18237.75	26458.56
Simplification time(s)	0.31	2.43	35.25	285.72	1390.80
No. of edges (after)	9908.12	36637.94	82335.06	140066.00	202527.69
No. of vertices (after)	1485.19	4339.56	8422.50	13144.81	17979.12
Out-of-memory(%)	0	0	25	50	75
MIP time(s)	24.92	1780.54	14287.70	17339.62	26667.95
LP time(s)	7.24	31.49	123.70	461.78	2168.99
No. of variables	10508.12	37237.94	82935.06	140666.00	203127.69
No. of constraints	1785.19	4639.56	8722.50	13444.81	18279.12
Integrality gap(%)	0.03	0	0	0	0
Non-integers in LP(%)	6.19	2.98	1.65	1.16	0.85

Figure 2.11: The size of the underlying graph



2.4 Conclusion

In this chapter, we introduce a shift rostering problem where the work rules are given in terms of prohibited meta-sequences and resource constraints. This provides much flexibility in modeling the complicated work rules found in practice. The canonical formulation could not solve the problem efficiently

Table 2.9: The solution value to (SRP2) and the time(s) used in solving (SRP2)

	The solution value					The time(s) used				
	3	4	5	6	7	3	4	5	6	7
5	31	6	18	63	0	3.40	14.39	48.02	59.24	94.49
6	98	0	0	6	0	4.55	20.49	57.46	73.80	161.05
7	0	0	0	0	-	8.01	34.85	50.82	585.80	*3508.13
8	0	0	0	78	-	6.72	29.91	1558.29	3656.04	*7906.08
9	12	39	53	0	23	34.30	484.52	140.00	602.53	1064.19
10	126	0	145	-	-	19.15	182.81	231.20	*5167.52	*8035.16
11	0	93	51	0	-	14.42	76.62	5835.90	1173.55	*10877.10
12	0	54	0	19	-	24.02	1896.64	5152.50	1252.33	*16677.70
13	98	0	0	7	-	24.27	233.53	10620.80	1197.56	*19568.20
14	57	0	0	-	50	20.68	426.03	586.75	*11858.70	16481.70
15	22	0	-	-	-	69.75	278.46	*26594.50	*21479.60	*73508.30
16	105	50	-	-	-	27.32	2356.72	*45296.30	*28576.90	*67187.00
17	52	8	0	-	-	41.54	5187.92	50884.90	*56977.40	*42494.10
18	0	352	-	-	-	41.02	5322.90	*26408.50	*34651.70	*63592.80
19	0	111	0	-	-	31.96	7328.59	966.45	*72122.50	*17845.80
20	10	0	-	-	-	27.60	4614.28	*54170.80	*37998.70	*77685.40

"-": no optimal solution found before the computer runs out of memory.
"*": the time at which the computer runs out of memory.

when there are a large number of staff and there are a large number of feasible shift patterns. We proposed a graph-based formulation where the set of feasible shift patterns are represented by the (s,t)-paths of a graph. As the size of the graph-based formulation depends on the structure of the work-rule constraints and not on the number of staff, the formulation could solve some large instances efficiently. Furthermore, we have identified some constraint structures that give a graph of small size and these structures often appear in work rules that are commonly found in practice. Therefore, as verified computationally, the graph-based formulation could solve many large instances to optimality within a few minutes whereas as the canonical formulation could not solve any of the instances in a reasonable time.

In our future work, we may extend the formulations in which work rules are handled as soft constraints. Furthermore, some staff heterogeneity may be addressed if staff could be categorised into groups that are subject to different sets of work rules. It is also worthwhile to identify more constraint structures that yield graphs of small size.

Chapter 3

A Tabu Search Heuristic for the Heterogeneous Vehicle Routing Problem on a Multigraph

In this chapter, we study a time-constrained heterogeneous vehicle routing problem on a multigraph. The problem is motivated from the distribution of beer and malt beverages in China, with some characteristics including the possibility of alternative paths of travel under the prevalence of road toll charges, fleet heterogeneity, and time-restricted delivery. We provide a mixed-integer linear programming formulation of the problem and propose a tabu search heuristic for its solution. The tabu search is designed to address the parallel arc structure on the network, which necessitates modifications of the basic search operations such as insertion. Our numerical experiments are set up to capture some practical features of beer distribution systems in China and show that the tabu search is highly effective in obtaining near-optimal solutions quickly. Other findings from the numerical experiments suggest considerable transportation cost savings attributable to the utilization of alternative route structure and reveal some insights to aid distributors on their vehicle dispatch policies.

3.1 Introduction and Practical Motivation

Vehicle routing problems (VRPs) arise frequently in the delivery and collection of items between a central depot and a number of customer locations.

In the basic form, VRP is concerned with determining a set of minimum cost vehicle routes such that each route starts and ends at the depot, each customer is visited exactly once, and the total load on a vehicle does not exceed the capacity. As one of the most widely-studied combinatorial optimization problems, VRP has been adapted to address a variety of practical considerations.

Typically, the problem is studied on a complete undirected graph with vertices corresponding to the customer locations and arcs corresponding to the links between those locations. When an underlying road network is not complete, this representation can be obtained by computing the shortest path between possible origin-destination points on the original network. As mentioned by Garaix et al. (2010), the shortest path is generally computed based on a single attribute such as travel time, and this results in the alternative routes with different attributes (travel time, cost, distance, etc.) not being considered in the solution space. Garaix et al. (2010) have addressed these alternative routes, which often represent realistic trade-offs (e.g., travel time versus cost), by building a multigraph representation of the road network and showed the cost savings for an on-demand transportation problem. Baldacci et al. (2006) have previously introduced the multigraph structure into a multiple disposal facilities and multiple inventory locations rollon-rolloff vehicle routing problem. In this chapter, we adopt the idea of alternative route consideration and study a time-constrained heterogeneous vehicle routing problem. Similar to the work by Garaix et al. (2010), we illustrate the gains achieved by the consideration of alternative routes.

In particular, we study the following problem. There is a mixed fleet of vehicles positioned at a depot and a set of customer locations with fixed demand requirements. Vehicles have different types according to the characteristics of capacity, availability, and cost. The goal is to find the least cost vehicle routes, starting and ending at the depot, such that each customer is served by exactly one vehicle and total demand on the vehicle does not exceed its capacity. In the literature, this problem is referred to as mixed fleet or heterogeneous vehicle routing problem and generalizes the capacitated vehicle routing problem by introducing different vehicle types (e.g., Gendreau et al. (1999) and Yaman (2006)). In our work, we consider an upper limit on the route duration and study the problem on a multigraph, in which multiple parallel arcs between each pair of vertices correspond to the alternative paths connecting the two customer locations in the underlying road network. Following the naming convention in Baldacci et al. (2006), we refer to it as the heterogeneous vehicle routing problem on a multigraph (HVRP-MG). We provide a flow-based mathematical formulation for the HVRP-MG and propose a tabu search heuristic for its solution. Tabu search is a widely-used heuristic method for VRPs and can find near-optimal solutions efficiently. Presence of parallel arcs, however, renders simple tabu search operations such as insertion complicated; therefore we develop a new procedure for estimating insertion costs and integrate it into the search. Our numerical experiments demonstrate the effectiveness of the proposed tabu search heuristic and provide further managerial insights from a sensitivity analysis. We contribute to the literature in two less well studied areas of VRPs: time-constrained heterogenous vehicle routing problems and vehicle routing problems with alternative paths.

The practical motivation for this study comes from the distribution of fast-moving consumer goods, specifically, malt beverages and beer in China. In this industry, producers often rely on a highly fragmented, complex, and multi-tiered supply chain to reach their customers. One distinct feature of these systems compared to their counterparts in other countries such as the United States is the abundance of agents in the middle-tiers, especially the wholesalers. The wholesalers satisfy demand requests coming from multiple customer sources including retailers, and they are served by a number of distributors who replenish their stocks from the producer. In a beer supply chain, while there may be around five distributors carrying one producer's brand exclusively, it is not uncommon to have more than one hundred independent wholesalers who may sell goods from multiple brands. Major transportation activities take place between the producer and the distributors via long-haul carriers, and between the wholesaler or distributor and the customers in the form of local deliveries. Firms may use specialized services of 3PL companies for their distribution needs, although operating own private fleet or using market resources is more widespread in the industry. The sheer size of the network leads to frequent movements of the goods from upstream parties to downstream parties, which in turn contribute to increased transportation costs.

The transportation cost structure in China is shaped by the current logistics infrastructure and regulations which present some unique properties.

For example, as reported by Dai and Zhou (2008), the highway toll charges account for as high as to 20%-30% of total logistic costs in China, whereas they are often assumed negligible compared to the driver wages and loading/unloading labor costs in the United States. The transporters have the option of incurring lower toll charges by utilizing free public roads, but the trade-off is slower and less-reliable service. Other differences arise from fleet heterogeneity which is usually more significant in the Chinese systems, and road access limitations for different vehicle types due to high population densities in the Chinese cities. Furthermore, retailers often require frequent deliveries from their wholesaler, mainly because of space limitations in the stores and/or their reluctance to carry high stock levels given the customers' small shopping volumes caused in part by the relatively low vehicle-ownership percentage in the Chinese households. As a result, wholesalers, who are involved in intense competition, often have to deal with deliveries of small quantities and unplanned vehicle routes while also considering the trade-offs between delivery costs and ability to provide service in short notice. (Some examples of works motivated from logistics problems in China include Chen et al. (2001), Fisher et al. (1986), Ma et al. (2012), and Yu and Qi (2014).)

These unique properties call for development of effective operational strategies to address alternative paths of travel (e.g., trading off cost and time), heterogeneous vehicles, and time-restricted delivery trips. Focusing on the shipment of products from the wholesaler to the retailers, we capture these characteristics by modeling a vehicle routing problem on a multigraph. We aim to add to the body of knowledge in the analysis and management of transportation systems in China.

The rest of this chapter is organized as follows. In Section 3.2, we review the related literature. Section 3.3 describes the problem and provides a mixed-integer linear programming formulation for the HVRP-MG. Solution methodology is discussed in Section 3.4 with a focus on the tabu search algorithm. In Section 3.5, we conduct a computational analysis where we evaluate the performance of the tabu search heuristic as compared to a mixed-integer linear programming approach and a less sophisticated insertion heuristic. We also perform a sensitivity analysis to derive some managerial insights on the delivery operations. Section 3.6 concludes the chapter with a discussion of research limitations and extensions.

3.2 Literature Review

Starting with the early work of Dantzig and Ramser (1959) and Clarke and Wright (1964), the classical VRP has been extensively studied in the operations research literature. Solution approaches to the VRP concentrate on exact methods, construction heuristics, and meta-heuristics, which are reviewed in the papers by Cordeau et al. (2002), Cordeau et al. (2005), Gendreau et al. (2002), Gendreau et al. (2007), Laporte (1992), and Laporte (2007). Likewise, the literature on the variants of VRP is quite rich, studying a wide array of practical extensions such as time windows, (e.g., Desrochers et al. (1988) and Solomon and Desrosiers (1988)), multiple trips by vehicles (e.g., Brandão and Mercer (1997) and Taillard et al. (1996)), multiple depots (e.g., Gillett and Johnson (1976), Laporte et al. (1988), and Renaud et al. (1996)), and multiple types of vehicles (e.g., Gendreau et al. (1999) and Golden et al. (1984)).

The research stream on VRPs with multiple types of vehicle is the most relevant to our study. These problems are referred to as heterogenous or mixed fleet VRPs (HVRPs) and differ from the classical VRP in that a mixed fleet of vehicles with distinct capacities, fixed operating costs, and variable costs is used to serve the customers. Balinski and Quandt (1964), Golden et al. (1984), Salhi et al. (1992), Salhi and Rand (1993), and Yaman (2006) provide formulations for the HVRP. Due to the problem complexity, the main solution approaches developed for the HVRP have been of heuristic type and no exact algorithms have been proposed. Construction heuristics and meta-heuristics have been applied to the variants of HVRPs (e.g., Desrochers and Verhoog (1991), Gendreau et al. (1999), Leung et al. (2013), Pisinger and Røpke (2007) Renaud and Boctor (2002), Salhi and Rand (1993), and Wassan and Osman (2002)); and in general, meta-heuristics such as tabu search or large scale neighborhood search are reported to have superior performances. A more detailed review of the literature on HVRP can be found in a recent survey by Baldacci et al. (2008). Different from the papers in this stream, we study an HVRP that allows multiple parallel arcs in between pairs of vertices. These arcs represent alternative paths of travel from a customer location to another in the underlying road network, and provide more flexibility in constructing routes based on different attributes of arcs (e.g., more costly but faster connections can be selected when delivery time limit is restrictive.)

Similar to Golden et al. (1984) and Yaman (2006), we extend the Miller-Tucker-Zemlin (MTZ) inequalities for the TSP (Miller et al. (1960)) to model the capacity and subtour elimination constraints in our formulation; but we also incorporate the multigraph structure. In view of the findings reported by Baldacci et al. (2008), we focus on heuristic approaches and propose a tabu search heuristic which leads to near-optimal solutions in our numerical tests.

Another area of research that is closely related to our work deals with vehicle routing problems on a multigraph. Baldacci et al. (2006) introduce this structure to study a multiple disposal facilities and multiple inventory locations rollon-rolloff vehicle routing problem. In particular, they show that the problem can be modeled as a single depot time-constrained VRP on a directed multigraph. Assuming that vehicles are identical and have unlimited capacity, they provide a set partitioning formulation of the problem and propose an iterative exact method. Our work differs from Baldacci et al. (2006) in that we incorporate a mixed fleet of vehicles and consider capacity limits in addition to the duration constraints on the vehicle routes. Furthermore, we formulate the problem using flow variables that indicate vehicle travels between customers and provide a heuristic solution approach based on tabu search. In a different study, Garaix et al. (2010) incorporate alternative paths into vehicle routing problems by building a multigraph representation of the underlying road network. As discussed by these authors, some difficulties arise when multiple parallel arcs are present between each pair of vertices in the graph. Specifically, while determining the exact schedule for a vehicle route is trivial in a simple graph after deciding the assignment of customers to vehicles and the visiting sequence in the routes, this is no longer true for a multigraph structure. Due to multiple arcs between vertices, additional decisions must be made regarding the specific arc selection. Garaix et al. (2010) propose a dynamic programming algorithm for arc selection in the context of a dial-a-ride problem. They examine how multigraph structure affects insertion operation and branch-and-price methods in solving these problems. Similar to Garaix et al. (2010), we consider multigraph structure in our model but our application area and solution approach differ from theirs as we study a heterogeneous vehicle routing problem and propose a tabu search heuristic.

We consider time limits on the vehicle routes, therefore, the literature on time (or distance) constrained VRPs is also relevant. Applications of these problems are provided in the papers by Assad (1988) and Laporte et al. (1984). With the objective of minimizing the total distance traveled, Laporte et al. (1984) and Laporte et al. (1985) provide formulations of the problem and develop exact solution methods based on constraint relaxation. Considering the minimization of total distance and number of vehicles used, Li et al. (1992) show that the optimal solutions under these objectives are closely related. We focus on minimizing the sum of travel and fixed dispatch costs while addressing time restriction for the routes as an additional constraint. Different from these papers, we model multiple types of vehicles, which necessitates a heuristic solution approach rather than exact methods for realistic-sized problems.

First proposed as a local search method for combinatorial optimization problems (Glover (1986) and Glover (1989)), tabu search has been widely applied to the VRPs with great success (e.g., Barbarosoglu and Ozgur (1999), Gendreau et al. (1994), Osman (1993), Rochat and Taillard (1995), Taillard (1993), Toth and Vigo (1998), and Xu and Kelly (1996)). The basic concept of tabu search is to explore the solution space iteratively by moving from one solution to the best neighboring solution, which is not in a tabu list. The tabu list is maintained to avoid cycling, where recently examined solutions are not considered for a number of iterations, unless they satisfy some aspiration criterion. The search may be improved by implementing intensification and diversification schemes which prevent the search being restricted to a limited portion of the search space and help explore the promising solutions more closely. Recent surveys on tabu search and other metaheuristics for VRPs can be found in Cordeau et al. (2005) and Gendreau et al. (2002). Tabu search implementations for the classical VRPs are extended to incorporate multiple types of vehicles in the context of HVRP (e.g., Brandão (2011), Gendreau et al. (1999), Salhi and Osman (1996), Wassan and Osman (2002)). We propose an implementation that considers the multigraph structure, which has not been addressed previously. While our tabu search implementation maintains some of the attributes proposed by these papers, e.g., allowing the search to move to infeasible solutions and penalizing solutions with frequently moved vertices (Gendreau et al. (1999) and Ho and Gendreau (2006)), it is adapted to handle the structure of multiple parallel arcs between vertices through an efficient arc selection procedure that is incorporated in the search.

3.3 Problem Formulation

In this section, we present a mixed-integer linear programming formulation for the HVRP-MG based on flow variables. Let $G(V, E)$ be a directed multigraph where V is a set of vertices and E is a set of arcs. Vertex $v_0 \in V$ denotes a depot from which all vehicles are operated, and the remaining vertices represent n customers. Each customer $i \in V \setminus \{v_0\}$ requires a certain number of units to be delivered to its location, representing the demand $d_i \in \mathbb{Z}^+$. Associated with this delivery is a service time denoted with $s_i \in \mathbb{R}^+$. E may contain parallel arcs between each pair of vertices which correspond to the alternative paths connecting the two locations in the underlying road network (Garaix et al. (2010)). There is a heterogenous fleet of vehicles with distinct capacities, fixed operating costs, and travel costs. The fleet is categorized into different types of vehicles, indexed by $\mathcal{K}$, so that vehicles of the same type are identical. For each type $k \in \mathcal{K}$, let $Q_k \in \mathbb{Z}^+$ denote the vehicle capacity, $f_k \in \mathbb{R}^+$ denote the fixed dispatch cost, and $m_k \in \mathbb{Z}^+$ denote the number of vehicles available. The travel time on an arc $e \in E$ is given by $t_e \in \mathbb{R}^+$, and when a vehicle of type k travels through arc e, a travel cost $c_e^k \in \mathbb{R}^+$ is incurred. The objective is to determine a set of vehicle routes with the minimum total cost, subject to the following requirements:

(i) Each route starts and ends at the depot.

(ii) Each customer is visited only once by exactly one vehicle.

(iii) The total demand served on a route of type-k vehicle does not exceed the vehicle capacity, Q_k.

(iv) All vehicles return to the depot within a given time limit L.

(v) The number of type-k vehicles in use does not exceed the number of vehicles available, m_k.

All problem parameters are assumed to be known with certainty. Consistent with Garaix et al. (2010), the arcs in the multigraph represent Pareto-optimal road paths only. For example, if each arc has two attributes, travel time and cost, then this implies that no arc exists for road paths that are dominated with respect to these two criteria; in other words, any path with a longer travel time and a higher travel cost can be ignored.

We next proceed to the formulation. In the following, we define the decision variables, $x_e^k = \{0, 1\}$, $y_{ij} \in \mathbb{R}^+$, and $w_{ij} \in \mathbb{R}^+$. Let

$$x_e^k = \begin{cases} 1, & \text{if a vehicle of type } k \text{ travels on arc } e; \\ 0, & \text{otherwise;} \end{cases} \qquad e \in E, k \in \mathcal{K}.$$

For all $i, j \in V$ with $i \neq j$, let $y_{ij} \in \mathbb{R}^+$ denote the total (cumulative) demand delivered when the vehicle leaves customer i to serve customer j. Similarly, let $w_{ij} \in \mathbb{R}^+$ denote the cumulative sum of service and travel times when the vehicle leaves customer i to serve customer j. When there is no vehicle travelling from customer i to customer j, both y_{ij} and w_{ij} are set to zero. For notational simplicity, let $E_{ij} \subset E$ denote the set of arcs from vertex i to vertex j, $\delta^+(i) \subset E$ denote the set of arcs that leave vertex i, and $\delta^-(i) \subset E$ denote the set of arcs that are incident to vertex i. Furthermore, it is convenient to treat the depot as a vertex with zero demand and zero service time.

Then, the HVRP-MG can be formulated as follows:

$$\min \sum_{k \in \mathcal{K}} f_k \sum_{e \in \delta^+(v_0)} x_e^k + \sum_{k \in \mathcal{K}} \sum_{e \in E} c_e^k x_e^k, \tag{3.1}$$

$$\text{s.t.: } \sum_{k \in \mathcal{K}} \sum_{e \in \delta^+(i)} x_e^k = 1, \forall i \in V \setminus \{v_0\}, \tag{3.2}$$

$$\sum_{e \in \delta^+(i)} x_e^k - \sum_{e \in \delta^-(i)} x_e^k = 0, \forall k \in \mathcal{K}, i \in V, \tag{3.3}$$

$$\sum_{e \in \delta^+(v_0)} x_e^k \leq m_k, \forall k \in \mathcal{K}, \tag{3.4}$$

$$\sum_{j \in V: j \neq i} y_{ij} - \sum_{j \in V: j \neq i} y_{ji} = d_i, \forall i \in V \setminus \{v_0\}, \tag{3.5}$$

$$y_{ij} \leq \sum_{k \in \mathcal{K}} \sum_{e \in E_{ij}} (Q_k - d_j) x_e^k, \forall i, j \in V : i \neq j, \tag{3.6}$$

$$\sum_{j \in V \setminus \{v_0\}} y_{v_0 j} = 0, \tag{3.7}$$

$$\sum_{j \in V: j \neq i} w_{ij} - \sum_{j \in V: j \neq i} w_{ji} = s_i + \sum_{k \in \mathcal{K}} \sum_{e \in \delta^-(i)} t_e x_e^k, \forall i \in V \setminus \{v_0\}, \tag{3.8}$$

$$w_{ij} \leq (L - s_j - t_e) \sum_{k \in \mathcal{K}} \sum_{e \in E_{ij}} x_e^k, \forall i, j \in V : i \neq j, \tag{3.9}$$

$$\sum_{j \in V \setminus \{v_0\}} w_{v_0 j} = 0, \tag{3.10}$$

$$w_{ij} \in \mathbb{R}^+, \forall i, j \in V : i \neq j, \tag{3.11}$$

$$y_{ij} \in \mathbb{R}^+, \forall i, j \in V : i \neq j, \tag{3.12}$$

$$x_e^k \in \{0, 1\}, \forall e \in E, k \in \mathcal{K}. \tag{3.13}$$

There are $|E||\mathcal{K}|$ binary variables and $2|V|^2$ non-negative continuous variables. The objective is to minimize the total fixed costs and travel costs. Constraints (3.2) ensure that each customer is serviced by exactly one vehicle on a single delivery, that is, the demand is not split. Constraints (3.3) balance the number of vehicles entering and leaving a vertex. Constraints (3.4) limit the number of vehicles in use. Constraints (3.5) - (3.6) ensure that all vehicle routes satisfy the capacity constraints. Finally, constraints (3.8) - (3.10) ensure that all vehicle routes satisfy the duration constraints. As also noted by Yaman (2006), no subtours will appear in any vehicle route due to the capacity (or duration) constraints. Notice that if constraints (3.4) are removed, the model would determine the optimal fleet size for each vehicle type simultaneously.

Yaman (2006) presents a number of mixed-integer linear programming formulations for the HVRP. Our model is most similar to the formulation with disaggregated flow variables in the sense that the duration and capacity constraints are handled using variables associated with the arcs and vehicle types. As also noted by Yaman (2006), one advantage of disaggregating the flow variables by vehicle types is the ease in treating different variable costs, which is an important aspect in our setting. In addition, we incorporate the multigraph structure into the formulation. An HVRP model with similar definitions of flow variables y_{ij} and binary variables x_e^k is given by Baldacci et al. (2008), but there is no consideration of service duration constraints or parallel arcs in their formulation.

3.4 Solution Methodology

Since the HVRP-MG has been formulated as a mixed-integer linear programming problem (3.3), commercial optimization software such as CPLEX can be used to obtain the optimal solution for small-sized instances. For instances of practical or large scale, the problem formulation becomes quite large and such tools cannot obtain optimal solutions within a reasonable amount of computation time. Therefore, we propose a heuristic-based approach for solving the HVRP-MG. In particular, we develop a tabu search heuristic which has been proven successful for a wide variety of vehicle routing problems. This section describes the tabu search heuristic.

Tabu search is a local search method which begins with an initial solution and explores the solution space by iteratively examining the neighboring solutions that are found by simple local modifications of the current solution. To avoid poor local optimum, the search moves to the best neighboring solution even if this move results in deterioration of the objective function. Recently visited solutions are forbidden for a number of iterations, i.e., they are placed in a tabu list, in order to prevent cycling. Additional features developed for tabu search can be applied to improve the search, see, e.g., Rochat and Taillard (1995), Gendreau et al. (1999), Taillard (1999) and Ho and Gendreau (2006).

Basic components of tabu search heuristic include neighborhood structure, initial solution, tabu moves, aspiration criterion, and diversification/intensification mechanisms. We next describe how these components are designed in our study to solve the HVRP-MG. For better illustration, we focus on the case in which the arcs are differentiated with respect to travel time and travel cost attributes; therefore there are two parallel arcs in between every pair of vertices. The procedure can be generalized to handle multiple parallel arcs.

3.4.1 Penalized Objective Function

Following Gendreau et al. (1994), Gendreau et al. (1999) and Ho and Gendreau (2006) and others, we allow infeasible solutions in the search space for the tabu search. This idea is typically implemented by relaxing some of the constraints and incorporating them into the objective function with the use of self-adjusting penalty parameters; hence the name *penalized objective function*.

Let $\mathcal{X}$ denote the set of solutions that satisfy requirements (i) and (ii), that is, every route starts and ends at the depot, and every customer is visited only once by exactly one vehicle. For a solution $x \in \mathcal{X}$, let $\mathcal{R}(x)$ denote the vehicle routes that contain at least one customer, and for a vehicle route $p \in \mathcal{R}(x)$, let $V(p)$ and $E(p)$ be the vertices and arcs in the route. If a vehicle of type $k \in \mathcal{K}$ is assigned to route p, the travel cost $c(p)$, overload $q(p)$ and overtime $t(p)$, representing the violation of capacity and duration constraints, respectively, can be written as:

$$c(p) = f_k + \sum_{e \in E(p)} c_e^k,$$

$$q(p) = \Big[\sum_{i \in V(p)} d_i - Q_k\Big]^+,$$

$$t(p) = \Big[\sum_{e \in E(p)} t_e + \sum_{i \in V(p)} s_i - L\Big]^+.$$

After incorporating the penalties for violations, the penalized objective function value of a solution $x \in \mathcal{X}$ is found by $z(x) = \sum_{r \in \mathcal{R}(x)} (c(r) + \alpha\ q(r) + \beta\ t(r))$, where $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}^+$ are the penalty weights that are self-adjusting in the search. Similar to the implementations in Cordeau et al. (2001), Ho and Gendreau (2006), and Xue et al. (2014), the penalty weights are initially set to 1. If the next solution satisfies the corresponding constraint, the penalty weight is divided by $\delta + 1$ where $\delta \in \mathbb{R}^+$ is a user defined parameter. Otherwise, it is multiplied by $\delta + 1$.

3.4.2 Initial Solution

A set of initial solutions is constructed with a parallel insertion heuristic method which iteratively inserts unassigned customers into vehicle routes. It begins by assigning exactly one arbitrarily selected customer to each vehicle route. Then, the remaining customers are inserted one by one, following a randomized order, into a vehicle route that minimizes the insertion cost, that is, the change in the penalized objective function value $z(.)$. We estimate insertion costs by using an efficient algorithm which performs vertex sequencing and arc selection, as described in Section 3.4.4. Insertion costs are frequently calculated when constructing feasible solutions in the course of the search.

3.4.3 Neighborhood Structure

The solutions in the neighborhood of a given solution $x \in \mathcal{X}$, denoted as $\mathcal{N}(x)$, are the solutions which are obtained after applying a single move operation on the current solution. The move operation involves relocating a customer from its current route to another route at a location that minimizes the penalized objective function.

To prevent cycling, if a customer has been moved from route r to route s in the i^{th} iteration, then moving the same customer back to route r is forbidden for θ iterations, where θ is a user controlled parameter. A tabu move is allowed only when the objective function value of the resulting solution is better than that of the current best feasible solution found by the search. This is referred to as the aspiration criterion and it aims to prevent the search from stagnation.

For diversification purposes, operations that are performed frequently in the search are penalized. We use a mechanism similar to that described by Gendreau et al. (1999) and Ho and Gendreau (2006), which is based on a penalty function $\phi(.)$ applied to non-improving solutions. For a given solution $x \in \mathcal{X}$, $\phi(x) = \lambda c(x)\sqrt{n}\vartheta_{ik}$, where n is the number of customers, ϑ_{ik} counts the number of times customer i has been moved to route k, and λ is a positive parameter that controls the intensity of diversification. Different from Ho and Gendreau (2006), the number of non-empty vehicles is not included in the penalty function since it is reflected in the fixed cost terms of the objective function.

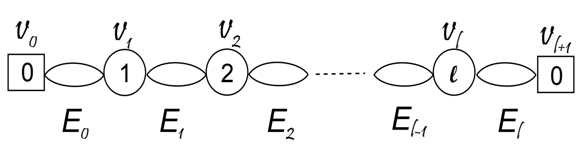
The best solution in the neighborhood, $\overline{x} \in \mathcal{N}(x)$ is the solution that minimizes $z(\overline{x}) + \phi(\overline{x})$ where $\overline{x}$ is non tabu, unless it satisfies the aspiration criterion.

3.4.4 Vertex Sequencing and Arc Selection

Insertion is an elementary operation that is performed frequently during the search when building the neighboring solutions — a customer is inserted into a vehicle route at a position that minimizes the change in the penalized objective function. It is an easy operation for a simple graph; however, it is difficult in the presence of multiple arcs. This is true even when the sequence of vertices is fixed, because decisions must be made regarding which arc to choose between every two successive vertices.

Suppose that $\mathcal{V} = (v_0, v_1, ..., v_l, v_{l+1})$ is the vertex sequence after a customer is inserted, where v_0 and v_{l+1} denote the depot. (See Figure 3.1 for an example). For all $i \in \{0, 1, 2, ..., l\}$, let E_i be the set of arcs between vertices v_i and v_{i+1}. For arc e, let $\widehat{c_e}$ be the travel cost for the corresponding vehicle and $\widehat{t_e}$ be the travel time. The problem is to select a combination of arcs $\mathcal{E} \in E_0 \times ... \times E_l$ minimizing $K_1 + \sum_{e \in \mathcal{E}} \widehat{c_e} + \beta\big[\sum_{e \in \mathcal{E}} \widehat{t_e} - K_2\big]^+$ where K_1 and K_2 are non-negative constants given by $K_1 = f_k + \alpha\big[\sum_{i \in \mathcal{V}} d_i - Q_k\big]^+$ and $K_2 = L - \sum_{i \in \mathcal{V}} s_i$ for the corresponding vehicle type $k \in \mathcal{K}$. Essentially, the problem corresponds to a multiple choice knapsack problem which is $\mathcal{NP}$-hard. In the multiple choice knapsack problem, a set of items are subdivided into l mutually exclusive classes denoted with E_i, $i \in \{1, 2, ..., l\}$, and exactly one item must be taken from each class in a way to minimize the total cost of the items selected. In the vehicle routing context with multigraphs, Garaix et al. (2010) introduced the problem as the Fixed Sequence Arc Selection Problem (FSASP) and proposed an exact solution method based on dynamic programming. While the dynamic programming approach can determine the insertion costs accurately, it is rather time consuming to implement in tabu search since the $\mathcal{NP}$-hard subproblem needs to be solved frequently during the search. Therefore, we propose a heuristic for the FSASP which efficiently determines the arc selection for a fixed sequence of vertices. The FSAS (Fixed Sequence Arc Selection) procedure (presented below) runs in polynomial time and takes into consideration that, if the arc selection is not altered after an insertion, the insertion cost may not reflect the actual value of the customer sequence and may mislead the search to an undesirable direction. Notice that this is never an issue for a simple graph. See Figure 3.2 for an example, where the insertion of customer 2 into a vehicle route $(1, 3)$ leads to a different arc selection between vertex 0 and vertex 3 in the multigraph but no such alteration is applicable in the simple graph.

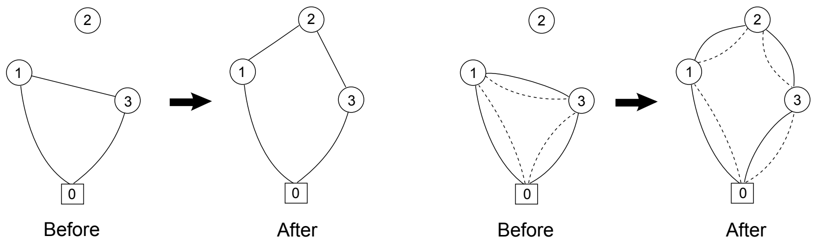
Figure 3.1: Arc Selection for a Fixed Sequence of Vertices



FSAS (Fixed Sequence Arc Selection) Procedure

Step 1 Determine the longest-time path. If the duration constraint is satisfied, return the longest-time path (which has the smallest cost).

Figure 3.2: Insertion operation in a simple graph (left-hand-side) and in a multigraph (right-hand-side).



Step 2 Otherwise, determine the shortest-time path. If the total duration of the shortest-time path is greater than or equal to the time limit, return the path with the following arcs.

$$e_i^* = \arg\min_{e \in E_i} \ \widehat{c}_e + \beta\widehat{t_e}, \qquad \forall i \in \{0, 1, ..., l\}.$$

Step 3 Otherwise, start with the shortest-time path and then pick the arcs, one by one, following a predetermined order based on a ratio of cost to time difference, until the penalized objective function is no longer improving:

(a) Denote the currently selected arc in E_i as $\bar{e_i}$.

(b) For all $i \in \{0, 1, ..., l\}$ and $e \in E_i \setminus \{\bar{e_i}\}$, determine the ratio of cost to time difference $r(e) = \frac{\widehat{c_e} - \widehat{c_{\bar{e_i}}}}{\widehat{t_{\bar{e_i}}} - \widehat{t_e}}$. Sort arcs $e \in E$ with $e \neq \bar{e_i}$ in the non-decreasing order of $r(e)$. Let $\tilde{E}$ be the sorted list.

(c) Consider the arcs in $\tilde{E}$ one by one in order. Let $\tilde{e_j} = \arg\min r(e)$ be in E_j. Replace $\bar{e_j}$ by $\tilde{e_j}$, and remove all arcs in E_j from $\tilde{E}$, that is, let $\tilde{E} = \tilde{E} \setminus E_j$.

(d) If $z(.)$ does not improve with the substitution, stop. Otherwise, repeat from Step (c).

For a fixed sequence of n vertices with at most k parallel arcs between a vertex-pair, the algorithm is implemented with time complexity $O(knlog(kn))$. To consider alternative sequencing of vertices, the above procedure is applied iteratively. In each iteration, a customer is picked and reinserted into the same vehicle route at a position that minimizes the penalized objective function $z(.)$. This is repeated until $z(.)$ is no longer improving. The resulting vertex sequence and arc selection is used to estimate the insertion cost which determines the search direction.

3.5 Numerical Analysis

In this section, we test the performance of the tabu search heuristic and provide some managerial insights from sensitivity analyses. The computational experiments are performed on generated data which reflect the practical applications that have motivated our work and give us flexibility in constructing different scenarios for the sensitivity analysis.

3.5.1 Experimental Setting

Table 3.1 shows the values of the model parameters in the base case. The distribution of customers is intended to mimic a densely populated region close to the depot and a sparsely populated remote region. For simplicity, we assume that two types of vehicles are in use and those with larger capacity have higher dispatch and travel costs (per unit distance) due to higher fuel costs and toll charges. This is consistent with research on per-mile costs of vans (semi-trucks or pick-ups) and trucks (e.g., Barnes and Langworthy (2003)), as well as the industry practice (Dai and Zhou (2008)). As mentioned previously, we focus on the case where the arcs have two attributes, travel cost and travel time. We assume that the travel cost and time are symmetric, that is, for a given vehicle type, the travel cost and time going from vertex i to vertex j are the same as those going from vertex j to vertex i. There are two parallel arcs in between every vertex pair which represent alternative routes with respect to travel time and cost characteristics. The travel time (t_{ij}) and cost c_{ij}^k of one of these arcs are generated according to the base-case values shown in Table 3.1. The values of the second arc, which is more costly but has a shorter travel time, are generated as follows. If vehicle k travels from vertex i to vertex j at a cost c_{ij}^k and a time t_{ij}, then an additional arc is introduced with a cost $R_1 c_{ij}^k$ and a time $R_2 t_{ij}$ where $R_1 \sim$ Uniform[1.1, 1.3] and $R_2 \sim$ Uniform[0.7, 0.9].

The values for the user controlled algorithmic parameters, i.e., λ, δ, and θ, will be reported in the next section when elaborating the numerical experiments. Algorithmic parameters are tuned by using one set of instances (training set), but all approaches are evaluated by using another set of instances (test set) and no parameter tuning is done on the test set to avoid over-fitting. All of the experiments have been conducted on a desktop personal computer running Windows 7 with an Intel Core i7-2600 processor 3.4

Table 3.1: Parameter values

Type k	Fuel cost δ_k	Toll charge η_{ij}^k per unit distance
Small capacity vehicle	Uniform[0.5,1.1]	Uniform[0.2,0.3] (appear on 50% of the arcs)
Large capacity vehicle	Uniform[1.4,2.0]	Uniform[0.4,0.5] (appear on 50% of the arcs)

Parameter		Values
Demand	d_i	Uniform[5,35], integer
Service time	s_i	Uniform$[1+0.2d_i,\ 2+0.2d_i]$
Location coordinates		$r \sim$ Uniform[0,25] (80% of customers)
$(r \cdot cos\gamma,\ r \cdot sin\gamma)$		$r \sim$ Uniform[25,100] (20% of customers)
		$\gamma \sim$ Uniform$[0,2\pi]$
Travel time	t_{ij}	Manhattan distance between i and j
Travel cost	c_{ij}^k	$(\delta_k + \eta_{ij}^k)\ t_{ij}$
Duration limit	L	260 time units
No. of Vehicles	m_k	Small capacity vehicles: $\max(3, \lceil \sum_{i \in V \setminus \{v_0\}} d_i/150 \rceil)$
		Large capacity vehicles: $\max(3, \lceil \sum_{i \in V \setminus \{v_0\}} d_i/300 \rceil)$
Vehicle Capacity	Q_k	Small capacity vehicles: 150
		Large capacity vehicles: 300
Dispatch cost	f_k	Small capacity vehicles: Uniform[95,105]
		Large capacity vehicles: Uniform[145,155]

GHz and 4 GB of main memory. Algorithms have been implemented in C++ and compiled using Visual Studio 2013.

3.5.2 Performance of the Solution Approaches

For comparison purposes, we test the following three solution approaches to the HVRP-MG.

- Mixed-Integer Linear Programming (MIP): The mixed-integer programming model (3.1) - (3.13) described in Section 3.3 is solved using CPLEX 12.5 with the default settings. The best feasible solution obtained within 2 hours of CPU time is reported. This serves as a reference to the other approaches.

- Insertion Heuristic (IH): The insertion heuristic described in Sections 3.4.2 and 3.4.4 is used to generate solutions with various randomized insertion order and number of vehicles, where the insertion costs are estimated with the FSAS procedure. The best solution obtained within 2 seconds of CPU time is reported. To increase the likelihood of obtaining feasible solutions, both of the penalty weights α and β are set to large positive numbers.

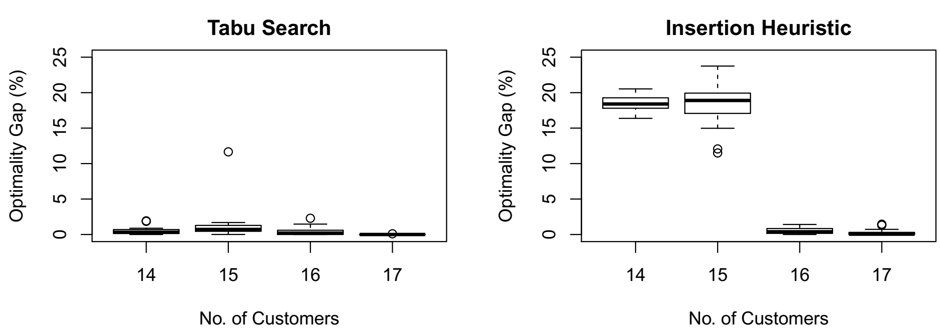
- Tabu Search Heuristic (TS): Initially, 5 solutions are constructed by the insertion heuristic. To encourage diversified structure, the penalty weights are initialized to small positive numbers. Both penalty weights α and β are set to 1 for testing. A tabu search is carried out on each of 5 solutions for 100 iterations. The best solution found in these iterations is selected as the starting point for the main tabu search. Then, the best solution obtained by tabu search within 2 seconds of CPU time is reported. The parameters used in the tabu search heuristic are shown in Table 3.2.

Table 3.2: Parameter values for the tabu search heuristic

Parameter	Value
Diversification intensity (λ)	0.0001
Penalty update factor (δ)	0.5
Tabu tenure (θ iterations)	$\lceil 5log_{10}(n) \rceil$

We begin the analysis by testing the quality of the heuristic solutions as compared to the optimal solution. We first use small-sized instances, where the number of customers is selected from the set {14,15,16,17}. For each case, 25 instances are generated and the optimal solutions are provided by the MIP approach. The optimality gap is calculated as $\frac{\text{H–Opt}}{\text{Opt}}100\%$ where H stands for the solution found by TS or IH, and Opt is the optimal solution. Figure 3.3 illustrates the performance of TS and IH. TS produces near-optimal solutions in almost all instances. We have observed that IH is able to produce good feasible solutions quickly and obtains higher quality ones for larger-sized instances.

Figure 3.3: Optimality gap on small-sized instances



Medium-sized instances with number of customers from the set {20, 25, 30} could not be solved to optimality within a reasonable amount of computational time, therefore, we use the best solutions obtained by MIP in a 2 hours CPU time limit as the benchmark results. For each number of customers, 5 instances are generated, and the fleet is comprised of 3 small-capacity and 3 large-capacity vehicles in all instances. Table 3.3 summarizes the results. As the number of customers increases, the performance of MIP significantly worsens due to the problem size and computational time limit. For these instances, IH and TS produce better solutions than MIP, with TS outperforming others in most of the cases.

Table 3.3: Performance of MIP, IH and TS on medium-sized instances

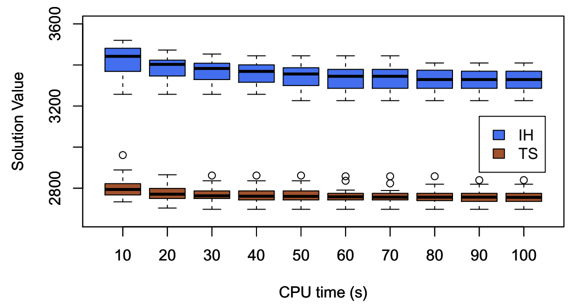
	No. of	Objective Value		
Instance	Customers	MIP	IH	TS
1	20	818.39	837.89	826.55
2	20	775.70	781.78	775.03
3	20	799.49	802.80	801.33
4	20	798.27	811.00	798.27
5	20	791.52	801.20	793.86
6	25	1006.46	1050.82	1042.75
7	25	1074.90	1041.11	1057.51
8	25	989.58	1071.87	991.48
9	25	991.39	1020.86	972.92
10	25	970.37	1024.53	1020.87
11	30	1066.25	1110.65	1034.13
12	30	1050.06	1134.39	1009.06
13	30	1049.53	1079.13	1032.99
14	30	1068.88	1125.93	1011.15
15	30	1071.43	1114.25	1055.85

We next consider large-sized instances with 100 customers and compare the performance of TS and IH with respect to different CPU time limits. A total of 25 instances are generated based on the parameter setting in Table 3.1, and the best solutions found within the corresponding CPU time limit are reported. Figure 3.4 shows the results. TS clearly outperforms IH for all the instances. Typically, IH generates a large number of feasible solutions in a short amount of time, while TS is able to find considerably better solutions within a few seconds. To propose high-quality solutions in practical time limits, our further numerical efforts will focus on TS.

3.5.3 Sensitivity Analysis and Managerial Insights

We perform sensitivity analysis to generate further insights that can aid decision making. To address realistic settings, we consider instances with 100

Figure 3.4: Performance of TS and IH under different computational time limits

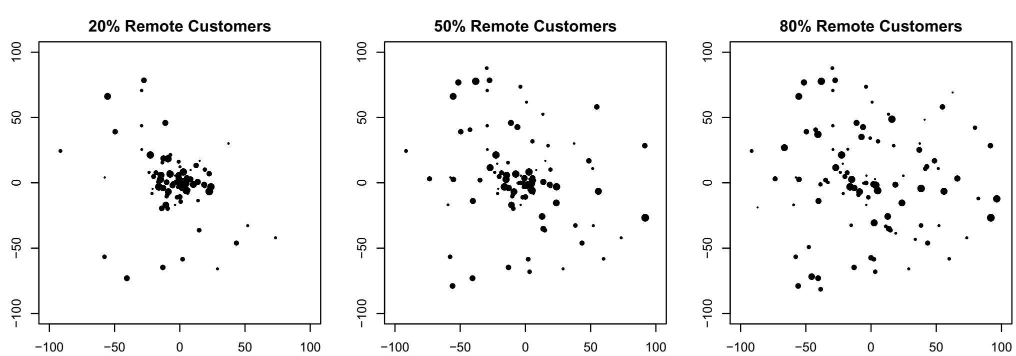


customers under various test scenarios. We use TS to solve all instances and terminate the search after 6000 iterations.

In the base-case, the problem parameters take the values shown in Table 3.1. Additional scenarios are generated by varying the route duration limit L, vehicle capacities, and the distribution of the customers. When the effect of one parameter is tested, other parameters are kept at their base-case values.

We test the values of L, the route duration limit, from 200 to 300 in increments of 10 units. The base-case values of vehicle capacities are denoted with a ratio of 1 and other scenarios are generated by varying this ratio within the set $\{0.6, 0.7, 0.8, 0.9, 1\}$, e.g., the ratio 0.6 implies 40% lower capacities than the base case. To obtain different distribution of customers, we vary the percentage of customers located in the remote region. This percentage is 20% in the base case, and other values tested in the experiments are {20%, 35%, 50%, 65%, 80%}. See Figure 3.5 for an example illustration of the customer locations at different parameter values. A larger point represents a customer with a larger demand at the location.

Figure 3.5: Distribution of customers

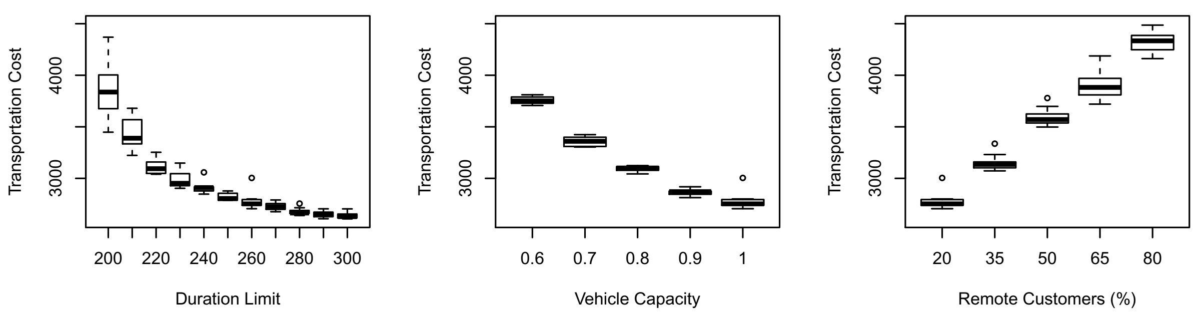


Overall, the maximum available fleet size is 8, with 6 being large-capacity vehicles. For each specific combination of parameter values, we test 10 instances.

Impacts of route duration limit, vehicle capacities, and customer distribution on transportation costs

Figure 3.6 shows the total cost of transportation including the dispatch and travel costs (i.e., the solution value) for the tested scenarios. The results illustrate the usual trade-off between transportation costs and customer service. As the route duration limit increases, it is possible to reduce transportation costs; however this comes at the expense of customer service, approximated by the total lead time to fulfill all customer orders. Some cost savings are obtained when the vehicles have larger capacities due to economies of scale or when only a small percentage of customers are located far from the depot due to reduced travel costs. Often times, these characteristics are difficult to control by an operating firm, therefore it is important that a balance is sought between the targeted lead time and transportation costs.

Figure 3.6: The impact of the route duration limit, vehicle capacities and customer distribution on transportation costs



Benefits of considering alternative routes

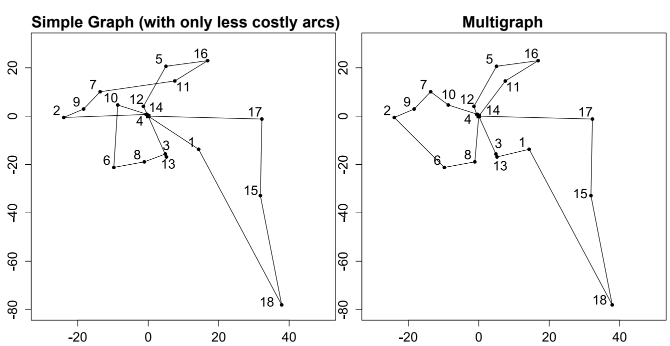
In practice, road networks present alternative ways (routes) of travel from one location to another with respect to attributes such as cost and time. When VRPs are studied on simple graphs, some of these attributes are omitted in constructing the graph since only one link is assumed between each pair of nodes. By incorporating alternative routes through a multigraph structure, these attributes can be explicitly considered with multiple links between nodes and potentially generate some benefits such as cost savings and flexibility in distribution planning. Garaix et al. (2010) have quantified such benefits in an on-demand transportation problem. In this section, we investigate the potential benefits for a VRP and identify circumstances under which significant gains can be generated by considering alternative routes.

Before analyzing the results, we present a small example to illustrate how the presence of multiple arcs, representing the alternative routes in the underlying network, leads to cost savings.

Example 1. *The arcs are built based on the experimental setting in Section 3.5.1; i.e., there are two arcs in between every pair of vertices, one representing the faster but more costly link and the other slower but less costly link. The route duration is limited to 260 time units. We first solve the problem on a simple graph by only considering the less costly arcs, and then on the multigraph by considering both types of arcs. The corresponding solutions are shown in Figure 3.7 and Table 3.4.*

In the simple graph, vehicle 2 returns to depot after visiting customer 1, but in the multigraph, the route is extended to include customers 3 and 13 by using the arcs {(1,13),(13,3),(3,0)} which are all of faster but more costly types. These arcs facilitate route extension without violating the route duration limit which would have been infeasible in the simple graph. While the travel cost of vehicle 2 increases, the total transportation cost is reduced from 657.92 to 652.11.

Figure 3.7: Effects of considering alternative routes in Example 1. (Left) Simple graph with only less costly arcs, (Right) Multigraph



As illustrated in the example, savings in travel costs, e.g., fuel costs and toll charges can be generated by taking advantage of the flexibility offered by the parallel arcs. For larger problems, it is also possible to observe savings in dispatch costs because service can be delivered with a smaller number of vehicles.

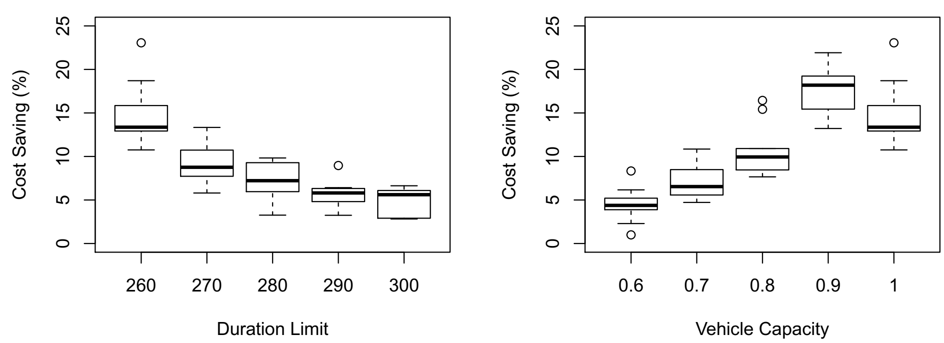
Table 3.4: HVRP solutions in Example 1

	Vehicle	Travel Time	Vehicle Load	Dispatch Cost	Travel Cost	Route
	1	170	120	101.94	199.92	(4,2,9,7,11,16,5,12)
Simple Graph*	2	258.01	92	101.94	294.14	(17,15,18,1)
	3	121.47	147	101.94	163.86	(14,10,6,8,3,13)
	1	107.79	88	101.94	162.85	(4,12,5,16,11)
Multigraph	2	259.82	136	101.94	303.03	(17,15,18,1,13,3)
	3	148.77	135	101.94	186.23	(14,10,7,9,2,6,8)

* Considering only less costly arcs between pairs of vertices

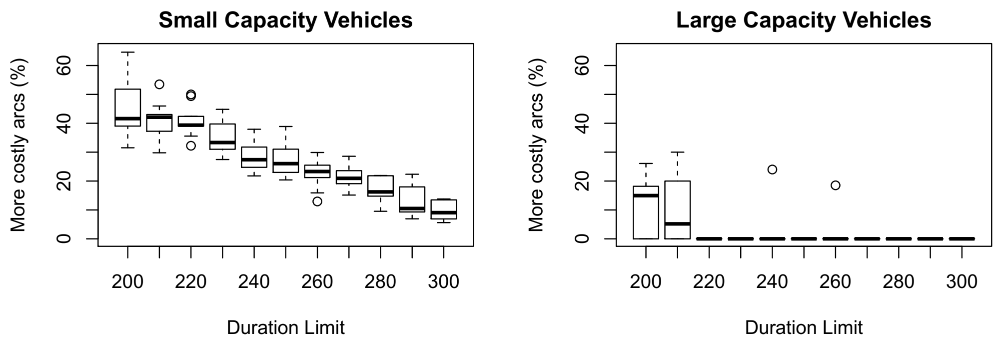
Next, we investigate the impact of alternative routes under various settings of the problem parameters. Figure 3.8 illustrates the transportation cost savings when alternative routes are considered. We calculate the cost savings as the percentage change between the total transportation cost of a simple graph, containing only the set of less-costly and longer-duration arcs and that of a multigraph with both sets of arcs, i.e., $\frac{\text{S}-\text{M}}{\text{S}}100\%$, where S and M represent the total cost in the simple graph and multigraph respectively. As can be seen in Figure 3.8, transportation cost savings of 5% to 20% can be observed when parallel arcs are considered. This is a considerable amount in distribution planning.

Figure 3.8: Transportation cost savings due to parallel arcs



Note that the savings are more pronounced when the route duration limit is at lower values. When the time limit is more restrictive, the set of more-costly arcs that facilitate faster travel are selected more often and therefore their impact in generating savings is greater. Furthermore, it is sometimes necessary to consider alternative routes in distribution planning, especially when products have to be delivered within a short due date. For example, if we disallow the parallel arcs and set the duration limit to a value less than 250, no feasible solution could be found in some of the instances. Parallel

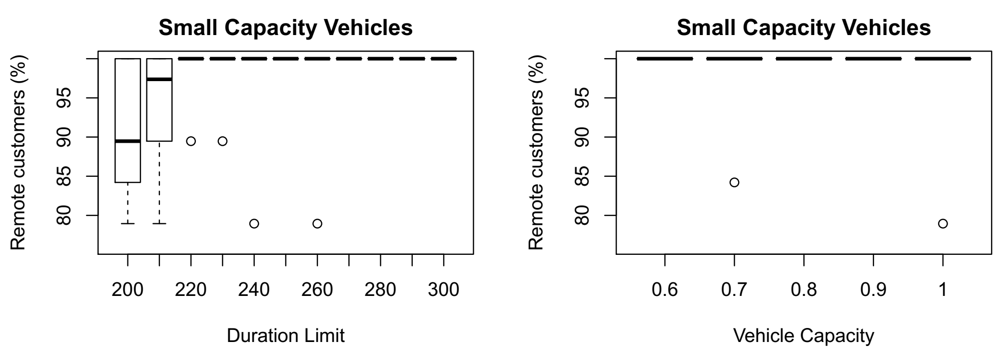
Figure 3.9: The percentage of more costly (and shorter duration) arcs traversed by vehicle type



arcs enable the construction of vehicle trips within time limitations.

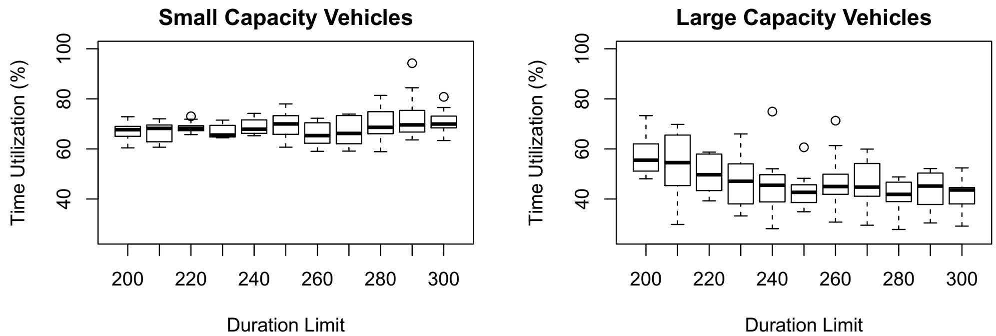
Figure 3.9 illustrates the utilization of parallel arcs in the routes of small and large capacity vehicles. We measure the utilization with a percentage, which is given by the ratio of the total number of more costly arcs traveled by a vehicle type on all trips to the total number of arcs traversed by the same vehicle type. The results show that the percentage of more costly arcs in the routes of small capacity vehicles is higher, implying that these vehicles tend to utilize the parallel arc structure more heavily. As the route duration limit becomes less restrictive, the percentage decreases steadily since the need for faster routes is reduced. This explains why large cost savings are observed for small duration limits, as we have seen in Figure 3.8.

Figure 3.10: The percentage of remote customers served by the small capacity vehicles



Figures 3.8 and 3.9 show that the transportation cost savings are generated mainly by utilizing the more costly arcs in the routes of the small capacity vehicles and that the large capacity vehicles tend to avoid the more costly arcs. It is worthwhile to take a deeper look into the reasons behind this observation. To this end, we consider the customer distribution and the

Figure 3.11: Time utilization by vehicle type



assignment of customers to vehicle types. Figure 3.10 shows the percentage of remote customers served by the small capacity vehicles at different values of the route duration limit and vehicle capacity. The percentage is measured by the ratio of the number of remote customers visited by the small capacity vehicles to the total number of remote customers. In Figure 3.11, we illustrate the time utilization in the routes of small and large capacity vehicles, where the utilization is measured by the ratio of total travel time to the route duration limit. We observe that most of the remote customers are served by the small capacity vehicles. Only when the duration limit is very restrictive, a few customers in the remote region are served by the large capacity vehicles. At the same time, the small capacity vehicles tend to travel longer distances to serve the customers in the more remote regions, while the large capacity vehicles travel shorter distances to serve the customers nearby. Therefore, the trade-off between travel cost and time is more critical for the small capacity vehicles, and consequently, alternative routes are more useful for this type of vehicles.

Further analysis on customer distribution

We perform further analysis on the impact of customer distribution over the network. By varying the percentage of customers located in the remote region, we can obtain different customer distributions, corresponding to, for example, uniformly or largely densely (sparsely) populated networks. (See Figure 3.5.)

In the experiments, we illustrate the percentage of remote customers served by the small capacity vehicles and the percentage of the more costly arcs in the routes of these vehicles. In addition, we show the time and capacity utilization with respect to different vehicle types. The capacity utilization is measured in a similar way to the time utilization, that is, it is the ratio of total load delivered by the vehicle to its available capacity. Figures 3.12-3.14 illustrate the findings.

When more customers are located in the remote region, the small capacity vehicles tend to utilize a larger percentage of the more costly arcs, i.e., faster links, in order to serve a higher number of remote customers within the route duration limit. In fact, even when customers are uniformly distributed, the small capacity vehicles serve more than 80% of the customers in the remote region. Furthermore, the small capacity vehicles tend to have higher time and capacity utilization than the large capacity vehicles. This suggests that the operating firm can achieve cost advantages and efficient use of resources by dispatching the small capacity vehicles to deliver the demand requests of the remotely-located customers.

Figure 3.12: Percentage of remote customers and more costly arcs in the routes of small capacity vehicles under different customer distributions

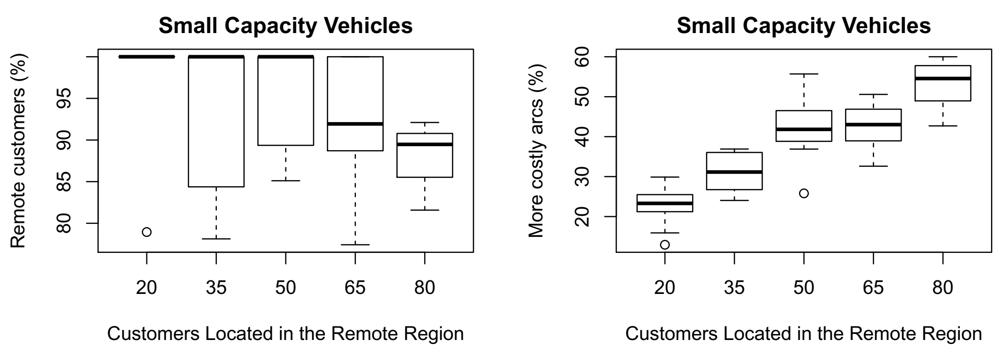


Figure 3.13: Capacity utilization by vehicle type under different customer distributions

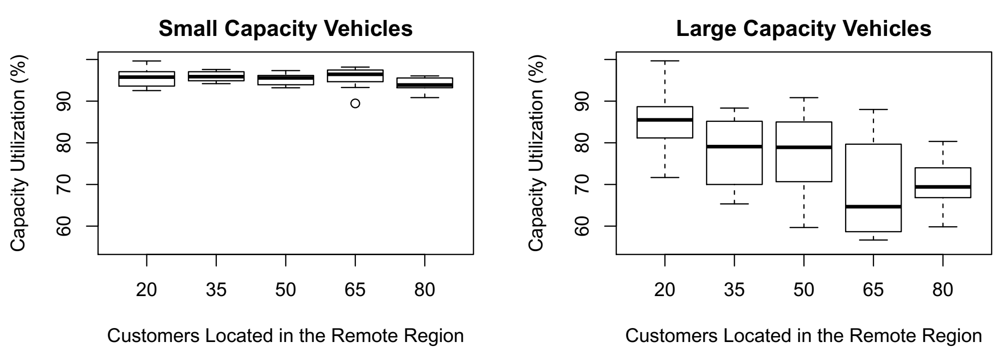
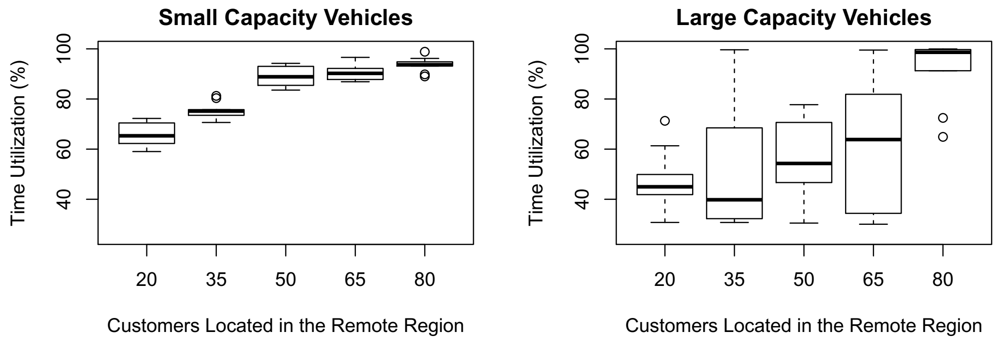


Figure 3.14: Time utilization by vehicle type under different customer distributions



3.6 Concluding Remarks

Vehicle routing has been a central component in the operation of logistics and distribution systems. When determining the routes of a fleet of vehicles, decision makers may need to consider multiple criteria such as operational cost and customer service. For example, while the primary objective could be minimizing the total transportation cost, delivery time and frequency could be other important considerations. In the related literature, such criteria are often addressed by developing minimum-cost vehicle routes under time or distance constraints. These models can be used to understand the trade-off between different criteria, depending on which decision makers can modify/customize vehicle trips. Another flexibility is related with the alternative paths of travel between locations on the network. For example, vehicles can travel from an origin point to a destination point on a road network by following alternative routes of travel, e.g., faster travel at a higher cost or distance. To consider the multiple attributes of these links in the solution space, it is necessary to expand the network representation of a problem by introducing parallel arcs between the vertices, which leads to a multigraph structure.

In this chapter, we adopt the idea of alternative route consideration and study a time-constrained heterogeneous vehicle routing problem on a multigraph. We provide a mathematical formulation of the problem and develop a tabu search heuristic as the main solution approach. Our numerical investigations show that the tabu search heuristic, which is designed to address the parallel arc structure of the network, is very effective in solving the problem. The tested instances illustrate the impact of alternative route consideration and also reveal some insights for the operation of the distribution system.

Our results suggest that considerable savings in transportation costs can be obtained by utilizing the alternative route structure, especially when the items must be delivered to customers in a restrictive duration limit. This arises because the multigraph structure enables solutions that take advantage of the multiple attributes of the arcs, e.g., an arc with a higher travel cost but shorter travel time can be added to a vehicle route, reducing the need for additional vehicle dispatches. As practices such as just-in-time delivery and supply chain integration are putting pressure on firms to commit to quick deliveries, alternative routes can be useful to gain benefits in transportation costs and flexibility. We also find that the parallel arc structure makes the highest impact on cost savings when the vehicles with smaller capacity are dispatched to serve the customers who are in remote locations and the vehicles with larger capacity are utilized to serve customers nearby in a minimum-cost way. Overall, our models and numerical findings can be useful in developing effective dispatch policies for distributers.

Although we have designed our numerical experiments by considering guidelines from practice, it would be interesting to test the developed policies by using real world data. In such a study, additional aspects of distribution operations such as delays, break-downs, changing customer requests could be considered to dynamically design the vehicle routes. It could also be worthwhile to incorporate other computational schemes to increase the efficiency of the tabu search.

Chapter 4

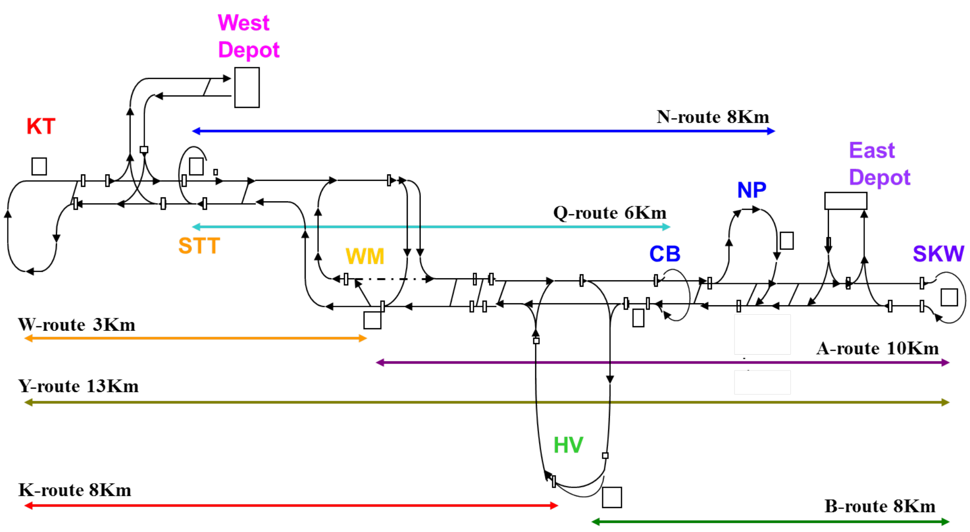
Real-Time Tram Scheduling

The research described in this chapter is motivated by the operations of a public transit company in Hong Kong. We investigate how real-time information can be utilized in combination with historical data to improve the controllers' routing and scheduling decisions practically. A dynamic and integrated vehicle and crew scheduling problem is introduced with the following characteristics: 1) The travel times are stochastic and time-dependent, and its realizations are only revealed during the execution of the plan. 2) The schedule can be revised when updated information is provided or when unexpected events occur. 3) Motormen and trams are scheduled simultaneously. The objective is to maximize the route frequencies and mileage in order to provide good service to passengers, and simultaneously minimize overtime and mealbreak delays for motormen. To mitigate unexpected delays due to uncertainties in operations, various mathematical models are proposed for revising the schedules in real-time under a rolling-horizon framework. The efficiency and the effectiveness of the formulations are evaluated via simulation using real-world data.

4.1 Introduction

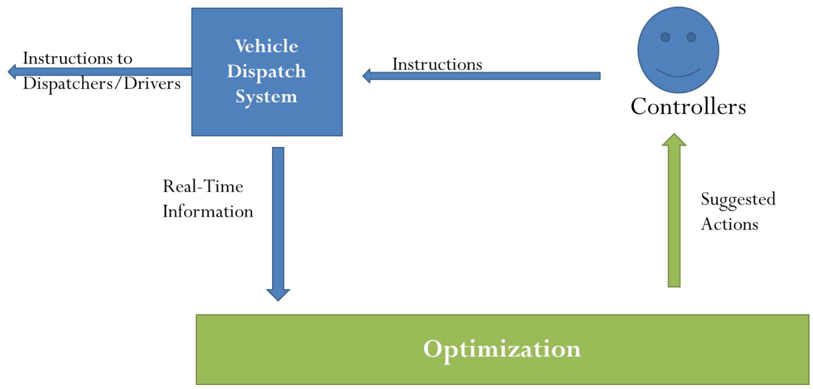
We are interested in developing practical solution approaches for real-time dispatch of a dynamic and stochastic vehicle and crew scheduling problem. The practical motivation for our research arose from the operations of a public transit company in Hong Kong. The company provides passenger transportation service in a densely populated area with 210,000 passengers per day and more than 50 stations. There are around 120 trams and 250 motormen available per day to provide the transportation service. Figure 4.1 illustrates the standard commercial routes of the company.

Figure 4.1: Standard commercial routes of the company



Information is centrally available to the control room to aid real-time decision making. Tram locations are detected using some location-sensors (e.g. RFID) and forwarded to the control room instantaneously. Moreover, inspectors are also sent to the termini and stations to monitor real-time traffic conditions, and manage the motormen and trams at the termini. All these location and traffic information is gathered and visualized using a computer system in the control room, and is monitored by the controllers. With the advance in location sensor technologies and database management system, historical and real-time information can be made readily available at a relatively cheap cost nowadays. We investigate how these valuable information can be utilized to improve the real-time routing and scheduling decisions of the controllers via a decision support system. Figure 4.2 illustrate the information flow of the proposed decision support system. When updated information is provided or when unexpected events occur, motorman schedules are revised by using a mathematical model. The corresponding suggested actions are shown to the controllers. The controllers may make the final decisions based on the proposed schedules. Appropriate instructions are forwarded to the inspectors and motormen via the vehicle dispatch system for the changes in the planned schedule.

Figure 4.2: A real-time decision support system



In this chapter, we introduce the dynamic and integrated vehicle and crew scheduling problem for real-time control with the following characteristics. 1) The travel times are stochastic and time dependent, and its realizations are only revealed during the execution of the plan. 2) The schedule can be revised when updated information is provided or when unexpected events occur. 3) Vehicles and drivers are scheduled simultaneously.

The objective is to maximize the route frequencies in order to provide good service to passengers, and minimize the violation of staff regulations (meal-break delays and overtimes) due to travel-time uncertainties.

4.1.1 Literature Review

In the following paragraphs, we summarize some literature related to integrated vehicle and crew scheduling problems, dynamic vehicle routing problems and disruption management. Extensive literature review on the related problems are provided by Pillac et al. (2013) for the dynamic vehicle routing problems, and Cacchiani et al. (2014) for an overview of recovery models and algorithms for real-time railway disturbance and disruption management.

Integrated Vehicle and Crew Scheduling

Although the decisions in vehicle routing and crew scheduling are interrelated, they are traditionally solved in a sequential way. Scheduling simultaneously the vehicles and crew yields significant benefit. Existing approaches for the integrated version are scarce, and are confined to static and deterministic approaches.

Freling et al. (2003), Huisman et al. (2005), Huisman et al. (2006), and Mesquita and Paias (2008) studied scheduling of bus lines based on column generation and Lagrangian approaches. Zäpfel and Bögl (2008) solve the problem in four phases: initialization, route generation, personnel assignment, solution evaluation. Tabu search and genetic algorithm are used separately to guide the solution approach.

Zäpfel and Bögl (2008) and Wen et al. (2011) investigated schedules for pickup-and-delivery using meta-heuristic approaches. Wen et al. (2011) proposed a multilevel variable neighborhood search heuristic. When the maximum working hours of drivers applies daily rather than accumulatively across different days, the problem become many daily planning problems that could be solved independently.

Dealing with Uncertainties

Most existing work on vehicle and crew scheduling problems consider deterministic travel times. However, travel times may not be realized as what are expected, due to the weather conditions, special events, traffic conditions, etc.

Laporte et al. (1992) formulated the vehicle routing problem with stochastic travel times, and solved it to optimality via branch-and-cut. The uncertainties in travel times were handled using chance constraints or expected penalties.

Huisman et al. (2006) demonstrates the potential benefits of solving a dynamic versus a deterministic version of the problem. They assume that the travel times are known exactly a certain amount of time before actual operations. This assumption is reasonable when the traffic conditions do not vary too much in a short time.

Stochastic programming approaches are effective in dealing with uncertainties. Improved solution quality could be obtained using real-time information to determine recourse actions arising from these uncertainties. An extensive summary of stochastic programming approaches can be found in Birge and Louveaux (2011).

Disruption Management

Disruption management approaches often are considered with respect to a tentative timetable, where the primary concern is to minimize the deviation from the planned schedule.

Cacchiani et al. (2014) presented an overview of recovery models and algorithms for real-time railway disturbance and disruption management. In case when a disturbance occurs, a common objective is to minimize the delays of trains or the delays of passengers. In case of a disruption, the system is recovered to the normal situation as soon as possible while the number of (additional) cancelled trips is minimized.

Jespersen-Groth et al. (2009) have given a comprehensive description of the problems related to disruption management arising in the railway systems in Europe. The disruption management process often involve solving sequentially three interrelated problems: timetable adjustment, rolling stock rescheduling, and crew rescheduling.

Walker et al. (2005) attempt to manipulate simultaneously the timetable and the crew schedule to deal with disruptions. By solving an integer programming model based on branch-and-bound with column and constraints generation, disruptions appear in a single rail-line can be handled effectively. The objective is to minimize simultaneously the deviation of the new timetable from the original one and the crew cost of the revised schedule.

Huisman (2007) developed a heuristic for a crew rescheduling problem in the case of planned track maintenance. Drivers are rescheduled to a subset of duties that is generated using a column-generation procedure. Potthoff et al. (2010) extend the work of Huisman (2007) in which the subset of duties is dynamically selected.

4.1.2 Description of the Motivating Problem

Commercial Routes and Deadheads

A *route* refers to a sequence of stations starting at an origin terminus and ending at a destination terminus. A route could be a *commercial route* that carries passengers, or a *deadhead* that carries no passenger (e.g. connecting two consecutive commercial routes). Standard commercial routes are designed in long term planning based on the estimated passenger demands. In practice, standard commercial routes occasionally need to be extended (more stations are introduced) or reduced (some stations are skipped), due to traffic delays which impact the mealbreak/signoff time of the motormen.

In our motivating application, we obtain information on the extended or reduced commercial routes commonly used in practice from the recent historical records stored in the database. We do not consider routes that are seldom used in the past in order to avoid unrealistic suggestions to the controllers. The possible commercial routes and deadheads could be updated before the day of operations in order to reflect the controllers' preference and the changes in traffic conditions. Table 4.1 shows some example commercial routes. Routes are categorized into groups. Routes in the same group have similar sequence of stations.

Table 4.1: Commercial routes and deadheads

Route	Origin	Destination	Type	Group	Distance
AE	WM	SKW	Standard	G_8	10350
AE+	WST	SKW	Extended	G_8	12008
AW	SKW	WM	Standard	G_1	10251
AW+	SKW	WST	Extended	G_1	12020
BE	HV	SKW	Standard	G_9	8031
BE-	HV	NP	Reduced	G_9	4275
BE–	HV	CB	Reduced	G_9	2061
QE	WST	CB	Standard	G_{12}	6038
QE+	KT	CB	Extended	G_{12}	6902
QE–	WM	CB	Reduced	G_{12}	4380
QW	CB	WST	Standard	G_5	6145
QW+	CB	KT	Extended	G_5	7328
QW-	CB	WM	Reduced	G_5	4376
⋮	⋮	⋮	⋮	⋮	⋮

Targeted Route Frequencies

Each route group is associated with some targeted route frequencies (i.e. headways) that are predetermined in long term planning (based on the origin-destination passenger demand and the targeted service levels). Table 4.2 shows the targeted route frequencies for all the route groups for different time periods of the day. These route-frequency requirements are expressed in a form of time-based demands described below. We divide the planning horizon into a number of time periods. For each time period, the targeted number of trams demanded for travelling in the corresponding routes are predetermined for providing adequate service levels. For example, as shown in the first row of Table 4.2, there are 18 trams required for travelling in the routes of group G_1 in the time period 4:00am — 5:00am.

Table 4.2: The number of trams required for all groups and time periods

Group	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	$\cdots$
G_1	18	11	21	24	19	19	16	26	20	27	$\cdots$
G_2	11	5	10	11	11	9	11	12	19	12	$\cdots$
G_3	6	7	11	13	21	9	16	16	21	16	$\cdots$
G_4	3	4	4	12	13	7	13	9	11	5	$\cdots$
G_5	4	3	3	4	3	1	6	5	9	14	$\cdots$
G_6	0	0	0	0	0	0	0	0	0	0	$\cdots$
G_7	0	0	3	4	4	4	4	4	4	4	$\cdots$
G_8	0	7	16	18	25	25	39	21	28	21	$\cdots$
G_9	0	12	6	13	12	14	12	15	15	20	$\cdots$
G_{10}	7	14	12	19	8	16	16	22	22	17	$\cdots$
G_{11}	5	5	11	15	8	11	9	10	7	14	$\cdots$
G_{12}	13	7	8	4	8	17	13	8	18	11	$\cdots$
G_{13}	0	0	0	0	0	0	0	0	0	0	$\cdots$
G_{14}	0	0	3	4	4	4	4	4	4	4	$\cdots$

Motorman Duties

Each motorman is assigned with an unique *duty* that describes the working period and the meal-break period of a day. The duties are designed in operational planning using a sophisticated commercial software package developed by the company, and are assigned to motormen in a rotating manner. Duties assigned to motormen are usually modified and finalized before the day of operations for special arrangements like sick leaves, switch of duties, etc. Duties of a day are categorized into shifts. Duties of the same shift have slightly different working periods and meal-break periods. Table 4.3 shows some example duties. While not entirely true in practice, we assume that all motormen have enough skill levels to operate any of the trams.

Working overtime or delays in meal-break time are undesirable since they are generally not preferred by the motormen. Even worse is that overtime hours are costly for the company because of additional overtime payment. When a meal-break is delayed, the meal-break end time should be adjusted

Table 4.3: Motorman Duties

Motorman	Signon Time	Signoff Time	Signon Terminus	Signoff Terminus	Meal-Break Start Time	Meal-Break End Time	Meal-Break Start Terminus	Meal-Break End Terminus
M_1	5:27	13:52	ED	CB	8:57	9:55	ED	ED
M_2	5:47	15:43	ED	CB	10:30	11:06	ED	ED
M_3	6:29	15:24	ED	CB	11:24	11:59	ED	ED
M_4	6:36	15:48	ED	CB	10:32	11:09	ED	ED
M_5	6:40	15:55	ED	CB	11:42	12:30	ED	ED
M_6	6:53	15:54	ED	CB	11:54	12:45	ED	ED
M_7	14:44	23:35	CB	ED	18:30	19:13	ED	ED
M_8	14:27	23:40	CB	ED	18:58	19:36	ED	ED
M_9	14:39	24:37	CB	ED	20:20	21:03	ED	ED
M_{10}	15:50	25:13	CB	ED	20:39	21:24	ED	ED
M_{11}	15:49	24:14	CB	ED	19:59	20:48	ED	ED
M_{12}	14:49	24:50	CB	ED	19:18	19:54	ED	ED
M_{13}	5:21	14:45	ED	ED	9:42	10:20	ED	ED
M_{14}	5:22	14:24	ED	ED	9:21	10:05	ED	ED
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

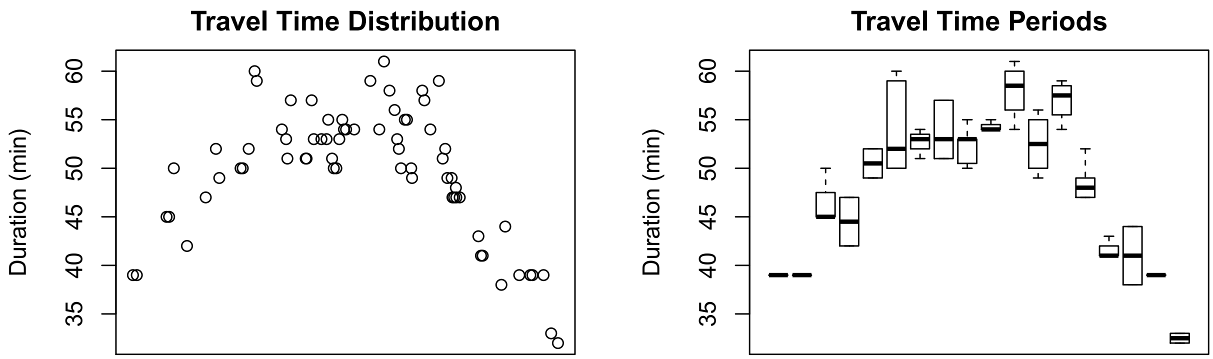
such that the duration of the meal-break is preserved. Furthermore, the meal-break period and the working periods can only start (and end) at the corresponding terminus described in the duty. Several termini are *relief points* where motormen meal-break, signon and signoff take place. Moreover, idling trams or idling motormen can only stay at these relief points. Trams and motormen that are available for a day are known before the day of the operations. At the beginning of the planning horizon, the initial termini of the trams depend on the operations of the previous day.

Time-Dependent and Stochastic Travel Times

Most transportation planning models in operational planning are deterministic and do not consider uncertainties in operations. However, public transit systems are inherently random. This is particularly true for buses and trams that share the use of the road with other traffic (v.s. trains that run on dedicated tracks). Therefore, disruptions on the planned daily schedule often occur in the daily operations due to unexpected traffic conditions, vehicle breakdowns, accidents, planned or unplanned special events, etc. When delays due to these uncertainties accumulate and propagate in the execution and operation of the planned schedule, poor service and high operational cost result.

To approach more realism in our model, we represent uncertain travel times as stochastic random variables with time-dependent distributions. The planning horizon is divided into a number of time periods (each with a length of 60 minutes in our case). The travel times for each route starting within a time period are assumed to follow a given probability distribution. Furthermore, there is a minimal stopping time at each of the termini, which is known as the *dwell time*. We consider deterministic dwell times in our model.

Figure 4.3: Travel-Time Distribution



The travel-time distribution of a commercial route in a single day is illustrated in Figure 4.3. The graph shown in the left-hand-side is the actual duration spent in travelling the route when it starts at the corresponding time. The graph shown in the right-hand-side illustrates the corresponding travel time distributions for all the periods in the day. The means and the standard deviations for commercial routes and deadheads for all time periods are predetermined using historical data. To illustrate the dynamicity, the means and standard deviations of the travel times for all time periods are shown in Table 4.4 and 4.5 respectively. For example, the travel time of route AE has a mean 52.5 minutes and a standard deviation 2.062 minutes if the route starts in the time period 4:00am — 5:00am.

Table 4.4: The mean travel times in each period

Route	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	⋯
AE	52.5	52.5	61.152	68.36	72.562	74.774	77.867	81.038	81.106	80.702	⋯
AE+	60.91	60.91	70.948	79.311	84.186	86.752	90.34	94.019	94.099	93.629	⋯
AW	53.716	53.716	60.113	70.721	74.266	73.751	76.224	77.091	78.442	79.031	⋯
AW+	62.986	62.986	70.486	82.925	87.081	86.478	89.377	90.394	91.979	92.669	⋯
BE	47.387	47.387	47.387	51.901	55.851	54.714	55.39	56.888	57.158	57.799	⋯
BE-	25.224	25.224	25.224	27.628	29.73	29.125	29.485	30.282	30.426	30.767	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 4.5: The standard deviation of the travel times in each period

Route	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	⋯
AE	2.062	2.062	5.257	3.62	4.56	4.466	6.596	5	4.65	4.641	⋯
AE+	2.392	2.392	6.099	4.2	5.291	5.182	7.652	5.802	5.395	5.385	⋯
AW	3.005	3.005	4.77	5.554	4.374	4.124	4.32	4.429	4.954	4.683	⋯
AW+	3.524	3.524	5.593	6.512	5.129	4.836	5.065	5.194	5.809	5.491	⋯
BE	3.644	3.644	3.644	3.173	4.235	3.579	3.486	3.897	3.56	3.707	⋯
BE-	1.94	1.94	1.94	1.689	2.254	1.905	1.856	2.074	1.895	1.973	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Objectives

We revise the planned schedules of motormen at a particular time of the day. The *schedule* of a motorman describes a sequence of tasks to be performed in the remaining working period of the day. A task could be a run in a route, a meal-break or to signoff. Each motorman should perform the tasks one by one and as early as possible. For all the routes planned in the schedules, a tram should be assigned. Since trams are assumed to be identical in our case, a motorman does not need to wait for the assigned tram in reality, as long as there are other trams available at the terminus. Furthermore, each task is associated with a scheduled start time. When trams are insufficient, the idling motorman with the earliest scheduled start time in the first run at the terminus should pick the next available tram first. Table 4.6 shows a realized schedule for one motorman.

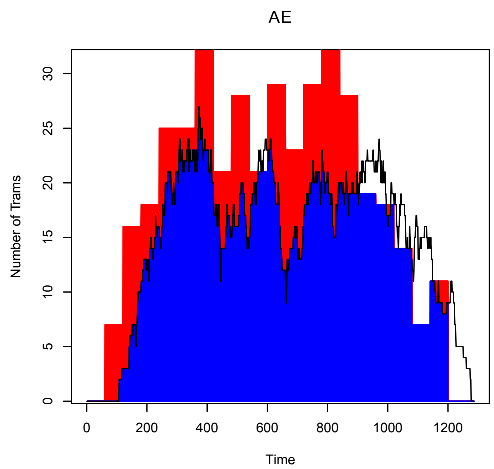
The goal in real-time control is to minimize the violation of staff regulations due to travel-time uncertainties, and to meet the targeted route frequencies in order to provide good service to the passengers. Meal-break delay and working overtime are discouraged using a penalty cost (per minute delay). The objective is to minimize the total overtime and meal-break delay, and maximize the coverage of passenger demand and mileage (total distance travelled in all commercial routes).

Figure 4.4 illustrates the coverage of a particular route on a day. The black line indicates the number of trams working in the routes over the course of a day. The histogram shows the demand in number of trams. The region of the histogram that is coloured in red illustrates the undercover, while the region coloured in blue illustrates the coverage.

Table 4.6: Realized schedule for one motorman

Motorman	Tram	Origin	Destination	Start Time	End Time	Route
M_4	V_1	ED	SKW	6:36	6:41	ED-SKW
M_4	V_1	SKW	KT	6:41	8:09	YW
M_4	V_1	KT	CB	8:09	9:05	QE+
M_4	V_1	CB	HV	9:05	9:19	BW–
M_4	V_1	HV	SKW	9:19	10:15	BE
M_4	V_1	SKW	ED	10:15	10:24	SKW-ED
M_4		ED	ED	10:24	11:01	*mealbreak*
M_4	V_2	ED	SKW	11:01	11:09	ED-SKW
M_4	V_2	SKW	WST	11:09	12:40	AW+
M_4	V_2	WST	NP	12:40	13:49	NE
M_4	V_2	NP	HV	13:49	14:20	BW-
M_4	V_2	HV	WM	14:20	15:00	KW–
M_4	V_2	WM	CB	15:00	15:44	QE–
M_4		CB		15:44		*signoff*

Figure 4.4: The coverage of a particular route on a day



4.2 Mathematical Models

In this section, we provide mathematical models for revising the planned schedule in real-time.

4.2.1 Parameters

Let $\mathcal{R}$ denote the set of routes, $\mathcal{T}$ denote the set of termini, $\mathcal{G}$ denote the set of route groups, and $\mathcal{M}$ denote the set of motormen that are currently available or will be available within the planning horizon (e.g. within the next 10 minutes). Let $\mathcal{S}$ denote the set of route sequences. A *route sequence* refers to a sequence of routes in $\mathcal{R}$ such that the destination of a route is equal to the origin of the next route in the route sequence.

A route sequence is *compatible with* a motorman when the route sequence starts with the available terminus of the motorman, and ends with the terminus of motorman's meal-break start terminus or sign-off terminus (if meal-break has been finished). The empty sequence is included to represent immediately meal-break or signoff at the available terminus. Note that all the possible route sequences between any two termini can be predetermined, and the same set of possible route sequences are used for rescheduling throughout the day.

A bipartite graph $G(V, E)$ is constructed with vertices $V = \mathcal{M} \cup \mathcal{S}$ and edges E such that, there is an edge connecting $u \in \mathcal{M}$ and $v \in \mathcal{S}$ if and only if route sequence v is compatible with motorman u. For notational convenience, let $\delta_+(v) = \{(p, q) \in E : p = v\}$ for all motormen $v \in \mathcal{M}$, which are the edges incident with vertex v and are representing all the compatible route sequences for motormen v.

Edges in E are associated with a number of attributes, in order to determine the objectives. For all edges $e = (d, s) \in E$, let β_e denote the expected meal-break delay (in minutes) if motorman d is assigned with route sequence s, and α_e denote the expected overtime (in minutes) if motorman d is assigned with route sequence s. For all route group $g \in \mathcal{G}$ and edge $e \in E$, let

$$a_g^e = \begin{cases} 1, & \text{if the first commercial route belongs to group } g; \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, let γ_e be the total distance for all the commercial routes in the corresponding route sequence. Table 4.7 summarizes the attributes.

Let $d_g \in \mathbb{Z}^+$ be the number of trams needed for travelling in the commercial routes of group g at the current time, which is referred as the *demand*. Let $w_1, w_2, w_3, w_4 \in \mathbb{Z}^+$ be some user-defined parameters to adjust the weights on the objective criteria.

Table 4.7: Summary of the edge attributes

Attribute	Notation	
Overtime	α_e	Positive number
Meal-break delay	β_e	Positive number
Indicate the group of the first commercial route	a_g^e	$\{0,1\}$
Mileage	γ_e	Positive number

4.2.2 Formulation

Let u_g be the undercover of group $g \in \mathcal{G}$, i.e the additional number of trams need to satisfy the demand of g. For all edges $e \in E$, let

$$x_e = \begin{cases} 1, & \text{if edge } e \text{ is selected;} \\ 0, & \text{otherwise.} \end{cases}$$

The problem is formulated as the following integer programming model.

$$\begin{aligned}
min \ & \sum_{e\in E} w_1\alpha_e x_e + \sum_{e\in E} w_2\beta_e x_e + \sum_{g\in\mathcal{G}} w_3 u_g - \sum_{e\in E} w_4\gamma_e x_e, && && (4.1)\\
s.t. \ & \sum_{e\in E} a_g^e x_e + u_g \geq d_g, && \forall g \in \mathcal{G}, && (4.2)\\
& \sum_{e\in\delta_+(v)} x_e = 1, && \forall v \in \mathcal{M}, && (4.3)\\
& u_g \geq 0, && \forall g \in \mathcal{G}, && (4.4)\\
& x_e \in \{0,1\}, && \forall e \in E. && (4.5)
\end{aligned}$$

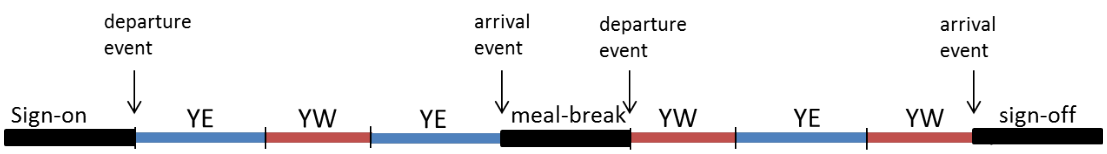
The objective function is to minimize the total overtime and the total meal-break delays, and at the same time, maximize the coverage and the total distance. Constraints (4.2) are used to satisfy the demand for trams travelling in the routes of each group. Constraints (4.3) ensure that all the motormen are assigned with a compatible route sequence. Since model (4.1) – (4.5) corresponds to a transportation problem, we may obtain integer optimal solutions by solving its linear-programming relaxation. Constraints (4.5) are therefore replaced by

$$0 \leq x_e \leq 1, \qquad \forall e \in E. \qquad (4.6)$$

4.2.3 Tram Availability Considerations

Essentially, model (4.1) - (4.4), (4.6) revises motorman schedules under the assumption that there are unlimited number of trams available at each of the termini, which is rarely the situation in our case. When this schedule is adhered to in practice, this results in unexpected overtime and meal-break delays due to unavailability of trams. In this subsection, we introduce tram availability considerations to the model.

To ensure that there is an available tram when a motorman is ready to work, we need to keep track of the number of trams after a *departure event* (a new route is started with a motorman after meal-break or sign-on) or an *arrival event* (a motorman finished a route and started a meal-break or sign-off). The events are illustrated using a motorman's schedule as follows. Events are associated on the edges. An event of an edge is used to represent the possible change in number of tram at a terminus if the edge is selected.



Let $\mathcal{E}$ denote the set of events associated on the edges in E. An event is indexed according to the terminus and time at which it may occur. Let N^l be the number of events that may occur at terminus $l \in \mathcal{T}$. For all $l \in \mathcal{T}$ and $i \in \{1, 2, ..., N_l\}$, let $q_i^l \in \mathcal{E}$ denote the i^{th} event that may occur at terminus l, let $e(l, i) \in E$ denote the edge that is associated with event $q_i^l \in \mathcal{E}$, and let

$$\delta_i^l = \begin{cases} 1, & \text{if event } q_i^l \text{ is an arrival event;} \\ -1, & \text{if event } q_i^l \text{ is a departure event.} \end{cases}$$

Let T_l be number of trams idling at terminus l at the current time. The following constraints are introduced to guarantee the availability of a tram, provided that travel times are deterministic.

$$T_l + \sum_{t=1}^{i} \delta_t^l \, x_{e(l,t)} \geq 0, \qquad \forall l \in \mathcal{T}, i = 1, 2, ..., N_l. \tag{4.7}$$

Let s_i^l be a decision variable to indicate the number of trams after the i^{th} event at terminus l. Constraints (4.7) can be rewritten as (4.8) – (4.10).

$$s_0^l = T_l, \qquad \forall l \in \mathcal{T}, \tag{4.8}$$

$$s^l_{i-1} + \delta^l_i \; x_{e(l,i)} = s^l_i, \qquad \forall l \in \mathcal{T}, i \in \{1, 2..., N_l\}, \tag{4.9}$$
$$s^l_i \geq 0, \qquad \forall l \in \mathcal{T}, i \in \{0, 1..., N_l\}. \tag{4.10}$$

Essentially, constraints (4.7) (or (4.8) – (4.10)) ensure that there is an available tram before motormen ready to work. However, in many cases, there are insufficient trams for motormen to start working immediately after meal-break or signon. Therefore, idling activities with durations {0 min, 10 min, ..., 60 min} are added at the beginning of the route sequences for motormen that have just finished a meal-break or just signed-on.

4.2.4 Multiple-Period Demand

Constraints (4.2) are used to meet the demand at the current time. This is greedy in nature because demands in the future are ignored. We may consider demands in the subsequent 3 periods that are estimated in real-time. Each period is set to one hour. Let P denote the set of periods. For all route group $g \in \mathcal{G}$ and period $p \in P$, let d_{gp} be the number of trams required for travelling in group g in period p. Furthermore, for all $e \in E$, $g \in \mathcal{G}$ and $p \in P$, let

$$a^e_{gp} = \begin{cases} 1, & \text{if selecting edge } e \text{ would run a route of group } g \text{ in period } p; \\ 0, & \text{otherwise.} \end{cases}$$

To consider multiple-period demands, (4.1), (4.2) and (4.4) are replaced by (4.11) – (4.13) shown below.

$$min \sum_{e \in E} w_1 \alpha_e x_e + \sum_{e \in E} w_2 \beta_e x_e + \sum_{p \in P} \sum_{g \in \mathcal{G}} w_3 u_{gp} - \sum_{e \in E} w_4 \gamma_e x_e, \tag{4.11}$$
$$\sum_{e \in E} a^e_{gp} x_e + u_{gp} \geq d_{gp}, \qquad \forall g \in \mathcal{G}, p \in P, \tag{4.12}$$
$$u_{gp} \geq 0, \qquad \forall g \in \mathcal{G}, p \in P. \tag{4.13}$$

4.3 Preliminary Results

In this section, we evaluate the efficiency and the effectiveness of the mathematical models that are described in Section 4.2 and are summarized in Table 4.8. We evaluate these methods via simulation with realistic data.

The simulation and the experimental settings are described in Section 4.3.1. To compare the performance of the proposed models, the efficiency and the effectiveness of the models are reported in Section 4.3.2 and 4.3.3 respectively.

Table 4.8: Summary of the methods

Model	Formulation	Description
1		do not revise the schedule
2	(4.1) – (4.4), (4.6)	consider current demand
3	(4.1) – (4.5), (4.8) – (4.10)	consider current demand and tram availability
4	(4.3), (4.5), (4.11) – (4.13)	consider multiple-period demand
5	(4.3), (4.5), (4.8) – (4.13)	consider multiple-period demand and tram availability

4.3.1 A Simulation Model for Evaluation

We developed a discrete-event simulation to evaluate the proposed methods. The simulation starts with the planned schedule that is currently in use (determined using a sophisticated software developed by the company). In a simulation test, we revise the schedules of motormen who are arriving within a fixed planning horizon using one of the methods shown in Table 4.8, until all motormen have been signed-off. We briefly describe the major components of the simulation in the following paragraphs.

Queues

Each relief point is associated with a tram queue and a motorman queue. If there is insufficient motormen, a tram is idling in a tram queue of the location. Similarly, motormen are idling in a motorman queue of the location when there is insufficient trams. Since the timetable is tentative in our case, a route is started when a tram and a motorman are both available at the same location. Therefore, the tram queue and the motorman queue should never be both non-empty at the same location. When there are trams and motormen both available at the same location, motorman with earlier scheduled start time of its next route has higher priority to start the next route with an available tram. In reality, this policy is maintained by some inspectors staying at the locations.

Event List

The simulation maintains a list of pending events. There are four types of events as described below.

- Motorman Arrival Event: when a motorman become available at a location (after a meal-break or when sign-on).
- Tram Arrival Event: when a tram became available at a location.
- Arrival Event: when a route is finished, a motorman and a tram become available at the destination.
- Scheduling Event: the motorman schedules are revised.

All pending events are organized as a priority queue, sorted by event time. At each iteration, the event with the earliest event time is processed as follows.

- Motorman Arrival Event: If there is an idling tram in the tram queue, the motorman should start the next route according to the route sequence; Otherwise, wait in the motorman queue. When there is more than one tram available, select the tram idled for the least time.
- Tram Arrival Event: If there is a motorman idling in the motorman queue, start the next route with the motorman; Otherwise, wait in the vehicle queue. When there is more than one motorman idling, the motorman with the earliest scheduled start time of its next route will depart with the available tram first.
- Arrival Event: The motorman should perform the next task according to the route sequence.
  - Case 1: The motorman starts the next route using the same vehicle.
  - Case 2: The motorman starts a meal-break. The tram become available at the destination terminus, the corresponding tram arrival event is then started immediately. Besides, a motorman arrival event is added to the event list where the event time is set to the time at which the meal-break is finished.
  - Case 3: The motorman signoffs. The tram become available at the destination terminus, the corresponding tram arrival event is then started immediately.

- Scheduling Event: The motorman schedules are revised using a method shown in Table 4.8.

The simulation ends when all the motormen have been signed-off. i.e. the event list is empty.

Data Collection & Implementation Issues

We use realistic data of a public transit company in Hong Kong for the simulation tests. All historical data is stored and managed in a database by the company. We examine the historical data and construct the instances for simulation. The best identified model in the simulation tests will be adopted in the actual environment in the company.

User-defined parameters used in the formulations are set according to the values shown in Table 4.9.

Table 4.9: User-defined parameters for the formulations

Parameter	Description	Unit	Value
w_1	Total overtime	minutes	1000
w_2	Total meal-break delays	minutes	10
w_3	Undercover	trams	1
w_4	Mileage	10^3 KM	-1

We conduct the experiments on a computer running Windows Server 2008 with an Intel Xeon CPU E5-2603 v2 (with two 1.8 GHz processors) and 16 GB of main memory. All models and simulations have been implemented in C++ and compiled using Visual Studio 2013. Mathematical models are solved using CPLEX 12.5 to optimality. The parameters of CPLEX are set to emphasis on optimality over feasibility. Furthermore, the network simplex algorithm of CPLEX is used to solve the linear programming models.

4.3.2 Efficiency

It is desirable that the solutions to the proposed mathematical models can be obtained quickly, before the data is updated again. In this subsection, we examine the solution times spent when using the various models.

We examine the performance for each of the models via a simulation of 10 days. The planned schedule is revised every 30 minutes. Table 4.10

Table 4.10: CPU times

Method	Times Revised	Mean CPU Time (s)	Worst Case CPU Time (s)
1	0	0	0
2	431	61.95	123.43
3	433	214.52	624.78
4	434	62.08	128.42
5	435	203.53	573.58

summarizes the CPU times of the models in the simulation tests. The table shows the number of times the formulation is solved in the simulation tests, the mean CPU time, and the worst case CPU time for solving the model once.

Models 2 and 4 (or methods 3 and 5) have similar CPU times. This suggests that multiple-period demands can be handled efficiently in the formulations. Models 3 and 5 have larger CPU times than models 2 and 4, which indicates that the tram availability constraints are relatively difficult to handle in the formulation. Overall speaking, the models can be solved in a reasonable time.

4.3.3 Effectiveness

In this subsection, we compare the effectiveness of the models vis the simulation, and justify the benefits of using the decision support system.

We examine the performance for each of the models via a simulation of 10 days. The planned schedule is revised every 30 minutes. Table 4.11 presents the average daily performance: the total overtime, mealbreak delay, idle time and early signoff time. Table 4.12 shows the worst-case performance for all motormen over the 10 days. The quantities shown in Table 4.11 and 4.12 are measured in minutes. Table 4.13 shows the coverage and mileage.

We compare model 1 with the other models to examine the potential benefits of using the decision support systems. As shown in the results, overtime and meal-break delays can be dramatically reduced by revising the schedules dynamically (using either of the mathematical models). Model 1 produces better coverage than the other methods because the planned schedule is followed strictly with additional overtime working hours.

According to the results, when we take into account more considerations into the formulation, the solution is generally has a better coverage, larger

Table 4.11: Performance about motorman schedules

Model	Overtime	Mealbreak Delay	Idle Time	Early Signoff
1	3999.6	1944.8	3479.5	3225.6
2	388.8	164.8	2888.9	1460.2
3	363.6	181.6	2843.2	1486.4
4	334.4	174.8	2961.9	1259.2
5	355.2	154.9	2877.5	1234.1

Table 4.12: The worst-case for all motormen

Model	Overtime	Mealbreak Delay	Idle Time	Early Signoff
1	110	83	179	218
2	27	33	80	109
3	31	27	123	101
4	29	28	83	85
5	34	24	150	94

Table 4.13: Passenger transportation service

Method	Coverage (%)	Mileage (KM)
1	70.36849	16893.5
2	67.72135	17605.47
3	67.97865	17577.9
4	68.77448	17187.1
5	69.19499	17188.08

mileage, smaller overtime, and smaller mealbeak delay. This leads to higher service level for passengers and better schedules for the motormen.

4.4 Conclusion

In this chapter, we introduce the dynamic and integrated vehicle and crew scheduling problem arising in a public transit company in Hong Kong. To mitigate unexpected delays due to uncertainties in operations, a number of mathematical models are proposed for revising the schedules in real-time under a rolling-horizon framework. Furthermore, we developed an event-based simulation model to evaluate our proposed models with realistic data. The simulation results suggest that it is worthwhile to consider multiple-period demands in the formulation. When time allowed, tram availabilities should be considered as well for better solutions.

In our future work, we may extend the mathematical models using stochastic programming approaches to fully utilize the historical and real-time data. Furthermore, it is also interesting to apply robust optimization approaches for schedules that have service-level guarantee.

Chapter 5

Summary

This thesis studies three routing and scheduling problems arising in manpower and transportation planning. These problems are rooted in real applications and carry interesting characteristics. By exploiting the problem structures, this thesis provides effective mathematical models and algorithms for solving the problems practically.

The Shift Rostering Problem

The second chapter of this thesis investigates a shift-rostering problem (SRP) — the assignment of staff to shifts over a planning horizon such that work rules are observed. Canonical formulations use decision variables to represent assignments of shifts to individual staff. Since the formulation size depends on the number of staff, canonical formulations would be too complex if there are many staff involved. For column-generation approaches, a subset of feasible shift-patterns is considered. The decision variables are used to represent the number of staff assigned to the shifts in the patterns. A resource-constrained shortest path subproblem is often used to generate feasible shift-patterns iteratively. However, with many work-rule requirements, having to solve the difficult subproblems frequently could be time-consuming. The purpose of the second chapter is to develop a novel exact approach for solving the SRP with many homogeneous staff.

The proposed approach consists of four steps. First of all, work rules are written in terms of prohibited meta-sequences and resource constraints. Work rules such as workload balancing and day-off assignments can be naturally formulated in this framework. An efficient algorithm is then used to construct an underlying graph representing all the feasible patterns. By solving a network flow model with side constraints, an optimal flow is obtained. Since all the work rules are handled using the underlying graph, the optimal flow represents many optimal solutions. By disaggregating the optimal flow into paths and transforming the paths into patterns, a preferred optimal solution is selected. The proposed approach produces a large number of optimal solutions, which offers flexibility in considering other managerial concerns when deciding on the roster.

In our future work, we may extend the formulations in which work rules are handled as soft constraints. Moreover, some staff heterogeneity may be addressed if staff could be categorised into groups that are subject to different sets of work rules. We have identified some constraint structures that give a graph of small size and these structures often appear in work rules that are commonly found in practice. It is also worthwhile to identify more constraint structures that yield graphs of small size. Furthermore, it is interesting to apply the method developed for SRPs to other problems.

The Heterogeneous Vehicle Routing Problem on a Multigraph

The road network underlying a distribution system presents multiple travel options for the vehicles. For example, a vehicle going from one customer location to another may consider different paths of travel based on criteria such as travel time, cost and distance. These alternative routes are typically not considered in the analysis of vehicle routing problems which are often studied on a simple graph. A multigraph structure, however, would enable the operators to build vehicle routes by utilizing the parallel arcs between each pair of customer locations which can help them address realistic trade-offs such as transportation costs and delivery time.

In the third chapter of this thesis, we study a time-constrained heterogeneous vehicle routing problem on a multigraph. The problem is motivated by the distribution of beer and malt beverages in China, with some characteristics including the possibility of alternative paths of travel under the prevalence of road toll charges, fleet heterogeneity, and time-restricted deliveries. We provide a mixed-integer linear programming formulation of the problem and propose a tabu search heuristic for its solution. The tabu search is designed to address the parallel arc structure on the network, which necessitates modifications of the basic search operations such as insertion. Our numerical experiments are set up to capture some practical features of beer distribution systems in China and show that the tabu search is highly effective in obtaining near-optimal solutions quickly. Other findings from the numerical experiments suggest considerable transportation cost savings attributable to the utilization of alternative route structure and reveal some insights to aid distributors on their vehicle dispatch policies.

In our future work, we may consider routing problems where arcs are associated with multiple attributes. e.g. cost, time, distance, etc. Moreover, it is worthwhile to compare our tabu search algorithm with other metaheuristics, e.g. large neighbourhood search and genetic algorithm. Furthermore, we may develop more dispatch policies and managerial insights in future experimental tests.

The Real-Time Tram Scheduling Problem

The research described in the fourth chapter is motivated by the operations of a public transit company in Hong Kong. Most transportation planning models are deterministic and do not consider uncertainties in operations. Therefore, disruptions on planned schedules often occur due to unexpected traffic conditions, bad weather, accidents, etc. Delays due to these uncertainties in operations may accumulate and propagate in the execution, resulting in poor services and high operational costs. To mitigate these unexpected delays, controllers are responsible for rescheduling trams and motormen in real-time to meet the targeted route frequencies, while respecting work rules for the duties assigned to motormen. With the advance in RFID technologies and database management systems, nowadays the locations of trams can be determined instantaneously at a relatively low cost. We investigate how real-time information can be utilized in combination with historical data to improve the controllers' real-time routing and scheduling decisions practically.

A real-time tram scheduling problem is introduced in this thesis with the following characteristics: 1) The travel times are stochastic and time-dependent, and its realizations are only revealed during the execution of the plan. 2) The schedule can be revised when updated information is provided or when unexpected events occur. 3) Motormen and trams are scheduled simultaneously. The objective is to maximize the route frequencies and mileage in order to provide good service to passengers, and simultaneously minimize overtime and mealbreak delays for motormen. We developed a number of mathematical models for revising the motorman and tram schedules dynamically under a rolling-horizon framework. A general event-driven simulation model is used to evaluate the efficiency and the effectiveness of the models with real-world data.

In our future work, we may extend the mathematical models using stochastic programming approaches to fully utilize the historical and real-time data. Moreover, robust optimization approaches could be applied to obtain schedules that have some service-level guarantee. Furthermore, metaheuristics could be developed for solving the problem more efficiently.

Bibliography

Ahuja, R.K., T.L. Magnanti, J.B. Orlin. 1993. *Network flows: theory, algorithms, and applications*. Englewood Cliffs, N.J. : Prentice Hall.

Assad, A.A. 1988. Modeling and implementation issues in vehicle routing. B.L. Golden, A.A. Assad, eds., *Vehicle Routing: Methods and Studies*. Elsevier Science, Amsterdam, 7–45.

Balakrishnan, N., R.T. Wong. 1990. A network model for the rotating workforce scheduling problem. *Networks* **20**(1) 25–42.

Baldacci, R., M. Battarra, D. Vigo. 2008. Routing a heterogeneous fleet of vehicles. B.L. Golden, S. Raghavan, E. Wasil, eds., *The vehicle routing problem: latest advances and new challenges*, *Operations Research/Computer Science Interfaces*, vol. 43. Springer US, 3–27.

Baldacci, R., L. Bodin, A. Mingozzi. 2006. The multiple disposal facilities and multiple inventory locations rollon-rolloff vehicle routing problem. *Computers & Operations Research* **33**(9) 2667–2702.

Balinski, M.L., R.E. Quandt. 1964. On an integer program for a delivery problem. *Operations Research* **12**(2) 300–304.

Barbarosoglu, G., D. Ozgur. 1999. A tabu search algorithm for the vehicle routing problem. *Computers & Operations Research* **26**(3) 255–270.

Barnes, G., P. Langworthy. 2003. The per-mile costs of operating automobiles and trucks. Research report MN/RC 2003-19, Minnesota Department of Transportation Office of Research Services.

Birge, J.R., F. Louveaux. 2011. *Introduction to stochastic programming*. Springer.

Brandão, J. 2011. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers & Operations Research* **38**(1) 140–151.

Brandão, J., A. Mercer. 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research* **100**(1) 180–191.

Burke, E.K., P. De Causmaecker, G.V. Berghe, H. Van Landeghem. 2004. The state of the art of nurse rostering. *Journal of Scheduling* **7**(6) 441–499.

Cacchiani, V., D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, J. Wagenaar. 2014. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological* **63** 15–37.

Cappanera, P., G. Gallo. 2004. A multicommodity flow approach to the crew rostering problem. *Operations Research* **52**(4) 583–596.

Caprara, A., M. Fischetti, P. Toth, D. Vigo, P.L. Guida. 1997. Algorithms for railway crew management. *Mathematical Programming* **79**(1–3) 125–141.

Caprara, A., M. Monaci, P. Toth. 2003. Models and algorithms for a staff scheduling problem. *Mathematical Programming* **98**(1–3) 445–476.

Caprara, A., P. Toth, D. Vigo, M. Fischetti. 1998. Modeling and solving the crew rostering problem. *Operations Research* **46**(6) 820–830.

Cheang, B., H. Li, A. Lim, B. Rodrigues. 2003. Nurse rostering problems—a bibliographic survey. *European Journal of Operations Research* **151**(3) 447–460.

Chen, F.Y., M. Goh, J. Lee, J. Ou, M. Sharafalia, C.P. Teo, P.G. Goh, P. Sivanathan. 2001. ST Logistics: distributing consumer goods in China. *International Transactions in Operational Research* **8**(2) 203–219.

Clarke, G., J.W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* **12** 568–581.

Cordeau, J.-F., M. Gendreau, A. Hertz, G. Laporte, J.S. Sormany. 2005. New heuristics for the vehicle routing problem. A. Langevin, D. Riope, eds., *Logistics systems and optimization*. Springer US, 279–297.

Cordeau, J.-F., M. Gendreau, G. Laporte, J.-Y. Potvin, F. Semet. 2002. A guide to vehicle routing heuristics. *The Journal of Operational Research Society* **53** 512–522.

Cordeau, J.-F., G. Laporte, A. Mercier. 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *The Journal of the Operational Research Society* **52**(8) 928–936.

Dai, J.J., C. Zhou. 2008. Beer distribution in China. Research report, The Supply Chain Logistics Institute, The School of Industrial and Systems Engineering, Georgia Institute of Technology.

Dantzig, G.B. 1954. A Comment on Edie's "Traffic Delays at Toll Booths". *Journal of the Operations Research Society of America* **2**(3) 339–341.

Dantzig, G.B., J.H. Ramser. 1959. The truck dispatching problem. *Management Science* **6**(1) 80–91.

Dantzig, G.B., P. Wolfe. 1960. Decomposition principle for linear programs. *Operations Research* **8**(1) 101–111.

Desaulniers, G., M. D. Hickman. 2007. Public transit. C. Barnhart, G. Laporte, eds., *Handbooks in OR and MS*, vol. 14, chap. 2. Elsevier B.V.

Desrochers, M., Lenstra J.K., Savelsbergh M.W.P., F. Soumis. 1988. Vehicle routing with time windows: Optimization and approximation. B.L. Golden, A.A. Assad, eds., *Vehicle Routing: Methods and Studies*. Elsevier Science Publishers B.V. (North-Holland), 65–84.

Desrochers, M., T.W. Verhoog. 1991. A new heuristic for the fleet size and mix vehicle routing problem. *Computers & Operations Research* **18**(3) 263–274.

Ernst, A. T., H. Jiang, M. Krishnamoorthy, D. Sier. 2004. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operations Research* **153**(1) 3–27.

Felici, G., C. Gentile. 2004. A polyhedral approach for the staff rostering problem. *Management Science* **50**(3) 381–393.

Fisher, M.L., J. Huang, B.X. Tang. 1986. Scheduling bulk-pickup-delivery vehicles in Shanghai. *Interfaces* **16**(2) 18–23.

Freling, R., D. Huisman, A. Wagelmans. 2003. Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling* **6**(1) 63–85.

Gamache, M., F. Soumis, G. Marquis, J. Desrosiers. 1999. A column generation approach for large-scale aircrew rostering problems. *Operations Research* **47**(2) 247–263.

Garaix, T., C. Artigues, D. Feillet, D. Josselin. 2010. Vehicle routing problems with alternative paths: An application to on-demand transportation. *European Journal of Operations Research* **204**(1) 62–75.

Gendreau, M., A. Hertz, G. Laporte. 1994. A tabu search heuristic for the vehicle routing problem. *Management Science* **40** 1276–1290.

Gendreau, M., G. Laporte, C. Musaraganyi, É.D. Taillard. 1999. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research* **26**(12) 1153–1173.

Gendreau, M., G. Laporte, J.-Y. Potvin. 2002. Metaheuristics for the capacitated VRP. P. Toth, D. Vigo, eds., *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 129–154.

Gendreau, M., J.-Y. Potvin, O. Bräysy, G. Hasle, A. Løkketangen. 2007. Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. S. Raghavan B.L. Golden, E.A. Wasil, eds., *The Vehicle Routing Problem: Latest Advances and Challenges*. Springer, 143–170.

Gillett, B.E., J.G. Johnson. 1976. Multi-terminal vehicle-dispatch algorithm. *Omega* **4**(6) 711–718.

Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* **13(5)**(1) 533–549.

Glover, F. 1989. Tabu Search Part I. *ORSA Journal on Computing* **1**(3) 1–190.

Golden, B.L., A.A. Assad, L. Levy, F.G. Gheysens. 1984. The fleet size and mix vehicle routing problem. *Computers & Operations Research* **11** 49–66.

Ho, S.C., M. Gendreau. 2006. Path relinking for the vehicle routing problem. *Journal of Heuristics* **12**(1–2) 55–72.

Huisman, D. 2007. A column generation approach for the rail crew rescheduling problem. *European Journal of Operations Research* **180**(1) 163–173.

Huisman, D., R. Freling, A.P.M. Wagelmans. 2005. Multiple-depot integrated vehicle and crew scheduling. *Transportation Science* **39**(4) 491–502.

Huisman, D., R. Freling, A.P.M. Wagelmans. 2006. A solution approach for dynamic vehicle and crew scheduling. *European Journal of Operations Research* **197**(2) 209–518.

Jespersen-Groth, J., D. Potthoff, J. Clausen, D. Huisman, L. Kroon, G. Maróti, M.N. Nielsen. 2009. Disruption management in passenger railway transportation. *Robust and Online Large-Scale Optimization*. Springer Berlin Heidelberg.

Laporte, G. 1992. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* **59**(3) 345–358.

Laporte, G. 2007. What you should know about the vehicle routing problem. *Naval Research Logistics* **54**(8) 811–819.

Laporte, G., M. Desrochers, Y. Norbert. 1984. Two exact algorithms for the distance-constrained vehicle routing problem. *Networks* **14**(1) 161–172.

Laporte, G., F. Louveaux, H. Mercure. 1992. The vehicle routing problem with stochastic travel times. *Transportation Science* **26**(3) 161–170.

Laporte, G., Y. Nobert, S. Taillefer. 1988. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science* **22**(3) 161–172.

Laporte, G., Y. Norbert, M. Desrochers. 1985. Optimal routing under capacity and distance restrictions. *Operations Research* **33**(5) 1050–1073.

Leung, S.C.H., Z. Zhang, D. Zhang, X. Hua, M.K. Lim. 2013. A metaheuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *European Journal of Operational Research* **225**(2) 199–210.

Li, C-L., D. Simchi-Levi, M. Desrochers. 1992. On the distance constrained vehicle routing problem. *Operations Research* **40**(4) 790–799.

Ma, H., B. Cheang, A. Lim, L. Zhang, Y. Zhu. 2012. An investigation into the vehicle routing problem with time windows and link capacity constraints. *Omega* **40**(3) 336–347.

Margot, F. 2010. Symmetry in integer linear programming. M. Jünger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, G. Reinelt, G. Rinaldi, L.A. Wolsey, eds., *50 Years of Integer Programming 1958-2008*. Springer, 647–686.

Mesquita, M., A. Paias. 2008. Set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem. *Computers & Operations Research* **35**(5) 1562–1575.

Millar, H.H., M. Kiragu. 1998. Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. *European Journal of Operations Research* **104**(3) 582–592.

Miller, C.E., A.W. Tucker, R.A. Zemlin. 1960. Integer programming formulation of traveling salesman problems. *Journal of ACM* **7**(4) 326–329.

Moz, M., M. Pato. 2004. Solving the problem of rerostering nurse schedules with hard constraints: new multicommodity flow models. *Annals of Operations Research* **128**(1–4) 179–197.

Musliu, N. 2006. Heuristic methods for automatic rotating workforce scheduling. *International Journal of Computational Intelligence Research.* **2**(4) 309–326.

Nicoletti, B. 1975. Automatic crew rostering. *Transportation Science* **9**(1) 33–42.

Osman, I.H. 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* **41**(4) 421–451.

Pillac, V., M. Gendreau, G. Guéret, A. L. Medaglia. 2013. A review of dynamic vehicle routing problems. *European Journal of Operations Research* **225**(1) 1–11.

Pisinger, D., S. Røpke. 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research* **34**(8) 2403–2435.

Potthoff, D., D. Huisman, G. Desaulniers. 2010. Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science* **44**(4) 493–505.

Renaud, J., F.F. Boctor. 2002. A sweep-based algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operations Research* **140**(3) 618–628.

Renaud, J., G. Laporte, F.F. Boctor. 1996. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research* **23**(3) 229–235.

Rochat, Y., É.D. Taillard. 1995. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics* **1**(1) 147–167.

Salhi, S., I.H. Osman. 1996. Local search strategies for the vehicle fleet mix problem. V.J. Rayward-Smith, I.H. Osman, C.R. Reeves, G.D. Smith, eds., *Modern Heuristic Search Methods*. Wiley, Chichester, 131–154.

Salhi, S., G.K. Rand. 1993. Incorporating vehicle routing into the vehicle fleet composition problem. *European Journal of Operational Research* **66**(3) 313–330.

Salhi, S., M. Sari, D. Saidi, N. Touati. 1992. Adaptation of some vehicle fleet mix heuristics. *Omega* **20**(5–6) 653–660.

Sarin, S.C., S. Aggarwal. 2001. Modeling and algorithmic development of a staff scheduling problem. *European Journal of Operations Research* **128**(3) 558–569.

Segal, M. 1974. The operator-scheduling problem: A network-flow approach. *Operations Research* **22**(4) 808–823.

Solomon, M.M., J. Desrosiers. 1988. Time window constrained routing and scheduling problems. *Transportation Science* **22**(1) 1–13.

Steinzen, I., V. Gintner, L. Suhl, N. Kliewer. 2010. A time-space network approach for the integrated vehicle- and crew-scheduling problem with multiple depots. *Transportation Science* **44**(3) 367–382.

Taillard, É.D. 1993. Parallel iterative search methods for vehicle routing problems. *Networks* **23**(8) 661–673.

Taillard, É.D. 1999. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research* **33**(1) 1–14.

Taillard, É.D., G. Laporte, M. Gendreau. 1996. Vehicle routing with multiple use of vehicles. *The Journal of Operational Research Society* **47**(8) 1065–1070.

Toth, P., D. Vigo. 1998. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing* **15**(4) 333–346.

Walker, C.G., J.N. Snowdon, D.M. Ryan. 2005. Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers & Operations Research* **32**(8) 2077–2094.

Wassan, N.A., I.H. Osman. 2002. Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society* **53**(7) 738–782.

Wen, M., E. Krapper, J. Larsen, T. K. Stidsen. 2011. A multilevel variable neighborhood search heuristic for a practical routing and driver scheduling problem. *Networks* **58**(4) 311–322.

Xu, J., J.P. Kelly. 1996. A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science* **30**(4) 379–393.

Xue, Z., C. Zhang, W.-H. Lin, L. Miao, P. Yang. 2014. A tabu search heuristic for the local container drayage problem under a new operation mode. *Transportation Research Part E* **62** 136–150.

Yaman, H. 2006. Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Mathematical Programming* **106**(2) 365–390.

Yu, M., X. Qi. 2014. A vehicle routing problem with multiple overlapped batches. *Transportation Research Part E* **61** 40–55.

Zäpfel, G., M. Bögl. 2008. Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics* **113**(2) 980–996.