YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER THESIS

VEHICLE AND CREW SCHEDULING PROBLEM IN PUBLIC BUS TRANSPORTATION

Hande ÖZTOP

Thesis Co-Advisor: Prof. Dr. Levent KANDİLLER
Thesis Co-Advisor: Assoc. Prof. Dr. Deniz TÜRSEL ELİİYİ

Department of Industrial Engineering

Presentation Date: 12.08.2016

Bornova-İZMİR
2016

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of master of science.

Prof. Dr. Levent KANDİLLER (Co-Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of master of science.

Assoc. Prof. Dr. Deniz TÜRSEL ELİİYİ (Co-Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of master of science.

Prof. Dr. Haluk SOYUER

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of master of science.

Assist. Prof. Dr. Özgür KABADURMUŞ

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of master of science.

Assist. Prof. Dr. Uğur ELİİYİ

Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School of Natural and Applied Sciences

ABSTRACT

VEHICLE AND CREW SCHEDULING PROBLEM IN PUBLIC BUS TRANSPORTATION

ÖZTOP, Hande
MSc in Industrial Engineering
Co-Supervisor: Prof. Dr. Levent KANDİLLER
Co-Supervisor: Assoc. Prof. Dr. Deniz TÜRSEL ELİİYİ
August 2016, 73 pages

In this thesis, the vehicle and crew scheduling phases of the transportation planning process are studied, motivated by the real problem of a public bus transportation authority. The objective is to determine the optimal number of different types of vehicles and crew members (drivers) to cover a given set of trips and deadheads, regarding working and spread time limitations of drivers at minimum cost. Binary programming models are formulated for each subproblem. In crew scheduling, an iterative valid inequality generation scheme is developed for eliminating task sequences violating the working time constraints. Performances of the developed solution methodologies for the subproblems are investigated through detailed experimentations, and the results show that the proposed optimal-seeking solution procedures are quite effective in terms of solution times. Furthermore, sequential and integrated approaches are proposed for the whole problem. As an integrated approach, a binary programming model is formulated and optimally solved for small-sized problem instances. However, larger instances cannot be solved within reasonable time limits due to exponentially increasing solution times. Therefore, a sequential approach is proposed. The performance of the developed approach is investigated through detailed experimentation and the results show that our approach is quite efficient for instances with up to 120 trips. Additionally, the sequential approach is compared with the integrated one for small-sized instances and found to be quite effective in finding near optimal solutions within very reasonable computation times.

Keywords: Vehicle Scheduling, Crew Scheduling, Public Transportation, Time Limitations, Eligibility Constraints, Fixed Job Scheduling.

ÖZET

TOPLU TAŞIMADA ARAÇ VE SÜRÜCÜ ÇİZELGELEME PROBLEMLERİ

Hande ÖZTOP
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Danışmanı: Prof. Dr. Levent KANDİLLER
Tez Danışmanı: Doç. Dr. Deniz TÜRSEL ELİİYİ
Ağustos 2016, 73 sayfa

Bu tezde, toplu taşıma operasyonlarının araç ve sürücü çizelgeleme aşamaları, bir toplu taşıma idaresinin gerçek hayat probleminden esinlenilerek çalışılmıştır. Problemde amaç önceden belirlenmiş seferleri ve araç atamalarından kaynaklanacak ölü kilometre seferlerini, sürücülerin toplam çalışma ve vardiya sürelerini dikkate alarak taşımacılığı minimum maliyetle karşılamak için gereken farklı tipteki araç ve sürücülerin sayısını optimal şekilde belirlemektir. Her iki alt problem için tamsayılı programlama modelleri geliştirilmiştir. Sürücü çizelgelemede, toplam çalışma süresini aşan görev sıralamalarını elemek üzere tekrarlamalı geçerli eşitsizlik yaratma yöntemi geliştirilmiştir. Her alt problem için geliştirilen çözüm yöntemlerinin performansları detaylı deneylerle araştırılmıştır ve sonuçlar önerilen optimal arama çözüm yöntemlerinin çözüm süreleri açısından oldukça etkili olduğunu göstermiştir. Bunun yanında, bütüncül problem için sıralı ve entegre olmak üzere iki yaklaşım önerilmiştir. Entegre yaklaşımda tamsayılı bir programlama modeli geliştirilmiş ve küçük boyutlu örnek problemler optimal olarak çözülmüştür. Ancak üstel artan çözüm süreleri nedeniyle büyük boyutlu problemler makul süreler içerisinde çözülememiştir. Bu nedenle araç ve sürücü çizelgeleme problemleri için geliştirilmiş olan tamsayılı programlama modellerinin sırayla çözüldüğü bir sıralı yaklaşım önerilmiştir. Bu yaklaşımın performansı kapsamlı sayısal deneyle araştırılmıştır ve sonuçlar sıralı yaklaşımın en fazla 120 sefere sahip örnekler için oldukça etkin ve verimli olduğunu göstermiştir. Ayrıca sıralı yaklaşım küçük boyutlu örnekler üzerinden entegre yaklaşım ile kıyaslanmıştır ve sonuçlar sıralı yaklaşımın çok makul sürede optimale yakın sonuçlar bulmada oldukça etkin olduğunu göstermiştir.

Anahtar sözcükler: Araç Çizelgeleme, Sürücü Çizelgeleme, Toplu Taşıma, Zaman Kısıtlamaları, Uygunluk Kısıtları, Sabit İş Çizelgeleme.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors Prof. Dr. Levent Kandiller and Assoc. Prof. Dr. Deniz Türsel Eliiyi for their support on my thesis, for their motivation and immense knowledge. Their guidance helped me very much during this thesis study.

I am indebted to all jury members for their valuable comments. I would like to thank to Dr. Uğur Eliiyi especially, for his support and insightful comments on operational planning processes in public transportation companies.

Finally, I would like to thank to my parents. They were always there supporting me and encouraging me with their best wishes.

Hande ÖZTOP
İzmir, 2016

TEXT OF OATH

I declare and honestly confirm that my study, titled “Vehicle and Crew Scheduling Problem in Public Bus Transportation” and presented as a Master’s Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions, that all sources from which I have benefited are listed in the bibliography, and that I have benefited from these sources by means of making references.

Hande ÖZTOP
İzmir, 2016

TABLE OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

INDEX OF ABBREVIATIONS

BDSP	Bus Driver Scheduling Problem
CRP	Crew Rostering Problem
CSP	Crew Scheduling Problem
FJSP	Fixed Job Scheduling Problem
GA	Genetic Algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
ISP	Interval Scheduling Problem
IVCSP	Integrated Vehicle and Crew Scheduling Problem
MD-VSP	Multiple Depot Vehicle Scheduling Problem
MVT-VSP	Vehicle Scheduling Problem with Multiple Vehicle Types
OFJSP	Operational Fixed Job Scheduling Problem
PMMLCG	Prime Modulus Multiplicative Linear Congruential Generator
POW	Piece of Work
RCSSP	Resource Constrained Shortest Path Problem
REF	Resource Extension Function
SD-VSP	Single Depot Vehicle Scheduling Problem
TFJSP	Tactical Fixed Job Scheduling Problem
TS	Tabu Search
VCSP	Vehicle and Crew Scheduling Problem
VJSP	Variable Job Scheduling Problem
VSP	Vehicle Scheduling Problem

1 INTRODUCTION

Public transportation companies are faced with difficulties in the transportation planning process by reason of population growth, the requirements for quality service and the need for efficient usage of the resources. Hence, the impact of planning systems in public transportation has been increasing, since significant cost savings are possible if the available resources such as crew and vehicles are used efficiently. As a consequence, the need for planning decision support systems that potentially reduce costs subject to specific operational restrictions has been increasing in operating public transportation.

In this thesis, the main cost incurring phases of the transportation planning process, namely the vehicle and crew scheduling activities are studied, inspired from the real life problem of the public bus transportation authority in Izmir. As of February 2016, the authority runs 316 bus lines with 1,401 buses and 4,092 employees among which 2,405 are bus drivers (Eshot, 2016). The daily passenger demand is approximately one million passengers. Regarding the authority's large vehicle fleet and crew, it can be said that if the vehicles and crew are scheduled efficiently, the operator can make considerable cost savings.

There are some operational restrictions for vehicles and drivers that should be taken into account. The trips require different types of vehicles having different characteristics such as capacity, average speed, fuel consumption, etc. Therefore, there are several vehicle classes based on the characteristics required to perform certain trips. When the operational constraints for the crew are concerned, each driver has a spread time limit from the start time to the end time of his/her shift, including the idle times. Furthermore, a driver cannot exceed the maximum total working time limit. The processing times of the tasks assigned to each driver are included in his/her working time as well as the sequence-dependent setup times, as the drivers must travel between the start and end locations in order to perform the assigned tasks. Since the trips require different types of vehicles, different types of crew with different capabilities are required. Therefore, there are several crew classes required to use certain vehicle types based on individual competencies. These vehicle and crew classes complicate the problem together with the time limitations.

In the literature, vehicle scheduling problem is studied for single/multiple depot, multiple vehicle types and vehicle classes extensions, commonly with the objective of minimizing both fixed vehicle costs and variable operational costs. Network flow, transportation and multi-commodity formulations are commonly used in order to model these extensions of the vehicle scheduling problem. In this thesis, a fixed job scheduling approach is proposed for the vehicle scheduling problem.

Regarding crew scheduling, column generation approach is commonly used in the literature, where the master problem is a set partitioning/covering problem and the subproblem is a resource constrained shortest path problem. In studies using this approach, different crew types are considered with respect to work day type such as part-time, full time, etc., with a set of operator rules and relevant regulations such as working time limit, number/length of breaks, maximum driving time limit, spread time limit, etc. In this thesis, a fixed job scheduling approach is also proposed for the crew scheduling problem having different crew classes with dissimilar capabilities and time limitations.

In the literature, most of the studies deal with crew scheduling and vehicle scheduling individually. There are new studies considering both problems, using a sequential or an integrated approach. In the sequential approach, vehicle and crew scheduling problems are solved sequentially whereas they are solved simultaneously in the integrated one. In these studies, homogenous/heterogeneous vehicle fleet and single/multiple depots are considered as well as a set of operator regulations and different crew types differing with respect to work day type, generally using a column generation approach. However, vehicle and crew classes are not considered together in these studies. In this thesis, a fixed job scheduling-based sequential approach and an integrated formulation are proposed for the vehicle and crew scheduling problem, considering vehicle and crew classes collectively, regarding crew working and spread time limitations. We claim that the vehicle and crew scheduling problems studied in this thesis can be applied to real life problems of public transportation companies, as many realistic operational constraints such as different vehicle/crew classes and crew time limitations are taken into account.

The complete public transportation process is explained in Chapter 2 together with the related terminology. General problem descriptions are also given in this chapter in order to provide a better understanding of the whole planning process.

Thereafter, a detailed literature review for the vehicle and crew scheduling problems and the integration approaches are provided in Chapter 3. As the vehicle and crew scheduling problems are formulated as tactical fixed job scheduling problems (TFJSP) in the thesis, a literature review for TFJSP is also included in Chapter 3. The contribution of this thesis to the existing literature is elaborated as well.

In Chapter 4, the proposed formulation for the vehicle scheduling problem is presented together with the necessary assumptions and notation. Consecutively, a computational study is described in order to evaluate the performance of the proposed formulation on its own. Similarly in Chapter 5, the proposed approach for the crew scheduling problem is presented together with its necessary assumptions and notation. Thereafter, the computational study is presented for evaluating the performance of the proposed formulation and approach.

In Chapter 6, both of the vehicle and crew scheduling problems are considered together. In Section 6.1, a binary programming model is presented for the integration of vehicle and crew scheduling problems, as well as the necessary assumptions and notation. In Section 6.2, a sequential approach is employed using the computational results of the vehicle scheduling problem in Chapter 4. The computational study and results for the sequential approach are presented in the same section. Thereafter, the sequential approach is compared with the integrated one for small-sized instances in Section 6.3.

Finally, the general results of the proposed approaches are discussed in Chapter 7 as well as the contribution of this thesis to the existing literature. Furthermore, potential future research topics are addressed.

2 PUBLIC TRANSIT-OPERATION PLANNING PROCESS

The public transit operation planning process includes five basic activities, usually performed in sequence: network route design, timetable development, vehicle scheduling, crew scheduling and crew rostering (Ceder, 2007). As shown in Figure 2.1, these activities are interrelated with each other; usually, the output of an activity at a higher level is an input for the activity at the subsequent level. In this thesis, the vehicle and crew scheduling activities are considered for public bus transportation and the definitions are made accordingly. The detailed terminology is given in Section 2.1.

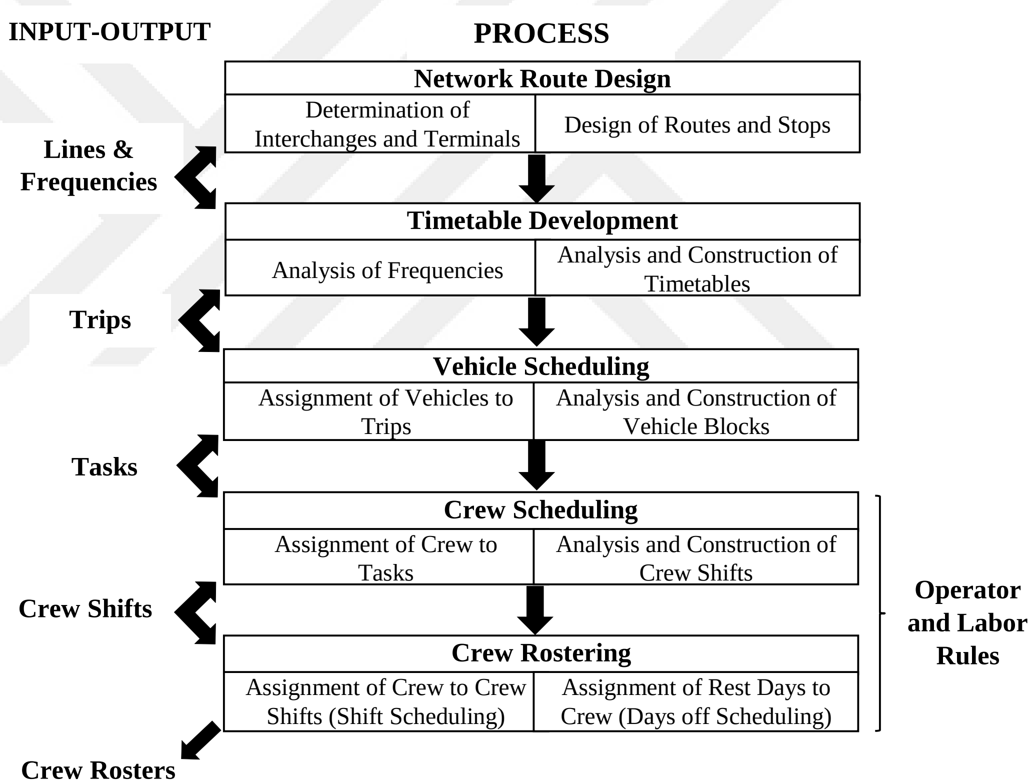

Figure 2.1 The Transit Operation Planning Process.

The public bus transportation service is composed of a set of *bus lines*, commonly defined by a number, which correspond to a bus traveling between two locations in the city. In network route design stage, the bus stops, terminals and interchanges are determined, as well as the route of each bus line. Then, the frequency of each bus line is determined based on public demand. Consequently, timetables are constructed, which lead to *trips* having start and end termini along with departure and destination times. In vehicle scheduling problem (VSP), the vehicles are assigned to trips. Each daily vehicle schedule known as a *vehicle block* is defined as a bus journey starting at a depot and returning to the same or a different one.

The crew scheduling problem (CSP) is a short term crew planning where the daily shifts are formed for the crew, covering all vehicle blocks. There is a set of *tasks* corresponding to vehicle blocks, and the aim is to define the sequence of these tasks as *shifts* in such a way that each task must be assigned to a shift without any overlaps. The feasibility of a crew shift depends on a set of operator and labor rules such as maximum driving time limit, break time duration, etc. The crew rostering problem (CRP) includes long term crew planning, where crew rosters are generated from crew shifts. This stage consists of *days off scheduling* and *shift scheduling*. Days off scheduling deals with the assignment of rest days to the crew over a planning horizon while shift scheduling deals with the assignment of the crew to constructed shifts.

Several software have been developed for transit operation planning process such as HASTUS (Rousseau and Blais, 1985), IMPACTS (Smith and Wren, 1988), HOT (Daduna and Mojsilovic, 1988), TRACS II (Kwan et al., 1999), GIST Decision Support System (Lourenço et al., 2001), GoalBus (GOAL Systems, 2016) and IVU Suite (IVU Traffic Technologies AG, 2016). These systems include the aforementioned planning activities at different levels.

2.1 Terminology

The relevant public bus transportation definitions used in the thesis are given below. It should be noted that *vehicle* and *bus* terms are used as substitutes, as well as *crew* and *driver*.

(Bus) Line: It is defined by a start location, an end location and a set of intermediate stops. It is commonly identified by a number.

Vehicle (Bus): There are several types of buses such as mini bus, solo bus, articulated bus, etc.; having different capacities and fuel consumptions.

Crew (Driver): There are several types of drivers that have different capabilities and driving licences.

Timetable: Time schedule of each bus line over a day. It contains a set of trips that are characterized by a start and end location as well as a start and end time.

Trip: A bus journey between two locations. It is represented by its start time, end time, start location, end location and arrival times at intermediate stops. Each trip must be performed by one vehicle.

Vehicle Block: The sequence of trips performed by a vehicle that start and end at the depot. It is daily schedule for a single vehicle.

Relief Point (Relief Opportunity, Relief Location): The planned locations and times where and when a change of driver may happen. Common relief points are start and end location of the bus line. However, intermediate stop on a line can be a relief point.

Depot: A parking garage with limited space for vehicles to stay overnight. It is a relief point.

D-trip: The portion of a trip which is created by dividing trips at relief points. Each d-trip must be served by a single vehicle and a single driver.

Deadhead: Driving of a vehicle without passengers. It can be a pull out (depot to start location of a trip), a pull in (end location of a trip to depot) and between two trips (from an end location of the trip to a start location of another trip).

Task: Activity performed on single vehicle by single driver without an interruption. It can be a d-trip or deadhead.

Piece of Work (POW): A sequence of tasks performed by a single driver on same vehicle without a break or walking activity. The feasibility of a piece of work is usually restricted by a minimum and maximum length.

Break: The rest time between two pieces of work. It can be long break or short break. It can be taken at any relief point.

Walking (Driver Movement): Crew movement without driving a vehicle. It can be movement from depot to a relief point, from a relief point to depot or between two relief points.

Shift (Duty): A daily work shift of a driver. It is generated by a set of pieces of work and breaks. The feasibility of a shift is usually restricted by number of piece of works, break duration, working time and spread time limits. Common shift types are early, late, night according to its beginning and ending time; and straight, split based on the duration of shift and the included number of pieces of works.

Spread Time: The total time between the start time and end time of the shift including the idle times.

Working Time: Sum of the duration of tasks and walking times in a shift, excluding idle times.

Sign on/off Time: Preparation time for a work shift. Shifts starting at the depot start with a sign on time. Consecutively, shifts ending at the depot end with a sign off time.

Round-Trip: Two consecutive trips from location A to B and then back to A.

Full Time (Straight) Shift: A work shift including one or two pieces of work with a short break between two pieces of work.

Split Shift: A work shift including two or more short pieces of work. It has a long break between two pieces of work such that a driver can go home during this break.

Roster: A work schedule for a driver over the long planning horizon which is composed of daily shifts and rest days.

The task definitions (d-trips and deadheads) are shown with an illustrative example in Figure 2.2.

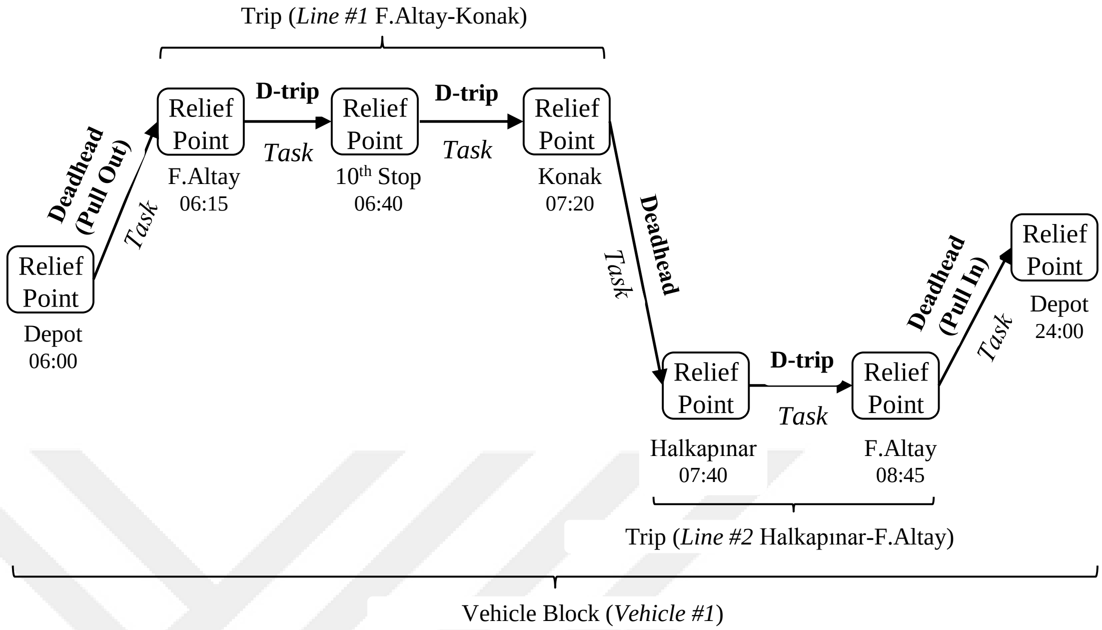

Figure 2.2 Partition of a Vehicle Block into Tasks.

In a given vehicle block, there are two trips belonging to two different lines, along with their start/end times and locations. As shown in the figure, start/end locations of the trips are relief points. Additionally, intermediate stop (10^{th} stop) of the trip belonging to *Line #1* is a relief point and the arrival time to this stop is known. The departure time from depot is determined according to the earliest departure time from the depot. Similarly, the arrival time to depot is determined as the latest arrival time to the depot. In this example, the vehicle returns to the depot earlier than the latest arrival time. This vehicle block can be summarized as follows: the vehicle leaves the depot at 6:00 and performs a deadhead task to reach the start location of its first trip (*Line #1*). After performing this trip, it performs another deadhead task to reach the start location of its second trip (*Line #2*). Subsequently, it performs a deadhead task to return to the depot. The trip belonging to *Line #1* includes two d-trip tasks. Thus, this trip can be performed by at most two crew members. On the other hand, the trip belonging to *Line #2* includes only one d-trip task, indicating that it must be served by only one crew member.

Crew shift generation from the vehicle blocks is shown with an illustrative example in Figure 2.3. As shown in the figure, there are two vehicle blocks and three crew shifts.

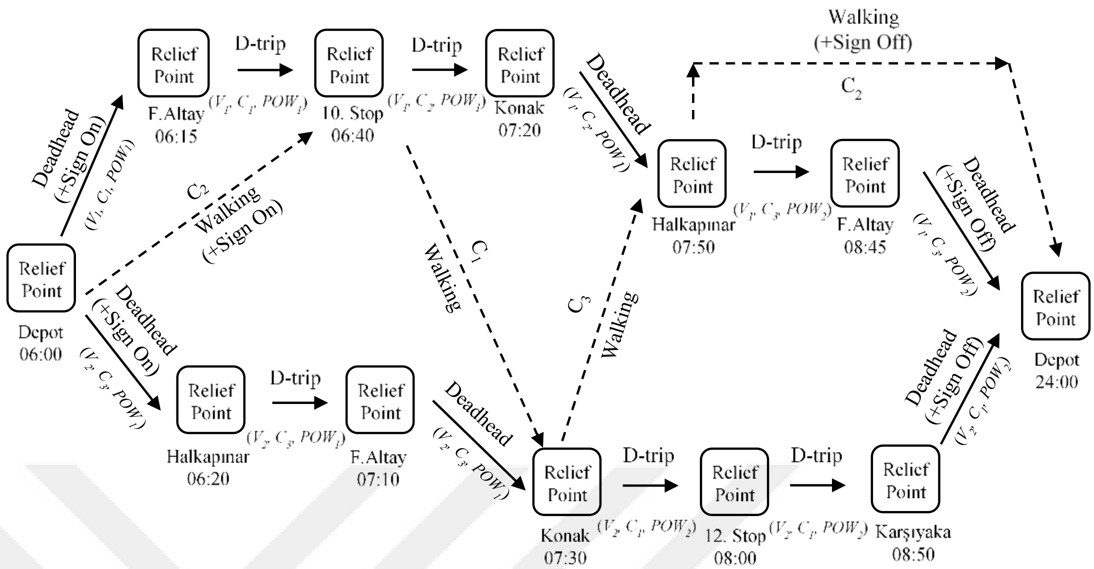

Figure 2.3 Crew Shift Generation from Vehicle Blocks.

Crew #1 starts his shift with a deadhead from the depot to the start location of its first trip (F.Altay) with *Vehicle #1*. After performing the d-trip with same vehicle, he leaves the vehicle at the relief point (10th stop), walks to the start location of its next trip (Konak) and performs the trip with *Vehicle #2*. Finally, he ends his shift with a deadhead from the end location of his final trip (Karşıyaka) to the depot with *Vehicle #2*. Meanwhile, *Crew #2* starts his shift by walking to the relief point of his first trip (10th stop), takes *Vehicle #1* and performs the d-trip from the relief point to the end location of the trip (Konak). After performing the d-trip, he performs a deadhead from the end location of the trip (Konak) to the start location of his next trip (Halkapınar) on same vehicle and leaves the vehicle there. Finally, he ends shift by walking from Halkapınar to the depot. In the example, *Crew #3* starts shift with a deadhead from the depot to Halkapınar with *Vehicle #2*. After performing the whole trip with the same vehicle, he performs a deadhead from the end location of the trip (F.Altay) to the start location of the next trip (Konak) on the same vehicle, leaves the vehicle there and walks to start location of his next trip (Halkapınar). He then performs the whole trip with *Vehicle #1* and ends shift with a deadhead from F.Altay to the depot. As it can be seen, a crew shift may start/end with a walking or a deadhead activity, and the crew can take/leave the bus at any relief point. All crew shifts start with a sign on time and end with a sign off time. These sign on and sign off times are included in the corresponding deadhead or walking activity.

The piece of works (POWs) can be extracted from Figure 2.3 as follows. Based on the definition, a deadhead from the depot to F.Altay, and a d-trip from F.Altay to 10th stop, is POW_1 for *Crew #1* (C_1) performed on *Vehicle #1* (V_1). The other POW are listed as:

- (V_1, C_2, POW_1): d-trip (10th Stop - Konak) and deadhead (Konak - Halkapınar)
- (V_1, C_3, POW_2): d-trip (Halkapınar - F.Altay) and deadhead (F.Altay - Depot)
- (V_2, C_3, POW_1): deadhead (Depot - Halkapınar), d-trip (Halkapınar - F.Altay) and deadhead (F.Altay - Konak)
- (V_2, C_1, POW_2): d-trip (Konak – 12th Stop), d-trip (12th Stop - Karşıyaka) and deadhead (Karşıyaka - Depot)

The crew shifts are constructed using these POWs by connecting them with walking or break activities.

2.2 Problem Definitions

The main problem definitions are provided in this section for the public bus transportation planning process.

2.2.1 Timetabling

The timetabling activity determines alternative frequencies and timetables in order to meet public transport demand. The demand fluctuates during the hours of the day or the days of the week due to changes in the transportation needs of the community. Therefore, the day is usually divided into time intervals, and the frequencies are determined for each time interval. Additionally, the days of the week are commonly considered separately as weekdays and weekends.

The purpose of timetabling is to establish alternative timetables for each line in order to meet variations in public demand. Alternative timetables are determined based on public demand regarding the service quality constraints (Ceder, 2007). During the construction of alternative timetables, several objectives are considered including the minimization of the waiting time of passengers, balancing vehicle utilizations, or minimization of the resources used. The common inputs of the timetabling activity are the routes, the times of first and last trips, average vehicle utilizations (number of passengers onboard the transit vehicle) between adjacent stops, the desired vehicle utilization, the boarding/alighting rate estimates for each stop, available resources and average running times between stops. The output of this activity are the trips, which correspond to the start and end locations as well as start and end times.

2.2.2 Vehicle Scheduling (Bus Scheduling)

Vehicle scheduling involves scheduling a fleet of vehicles to cover a set of trips at minimum cost. The trips are given by specified time intervals as well as start and end termini. The vehicles are grouped into depots according to their locations, and classified by their types in these depots. If there is a single depot, the problem is referred as a Single Depot Vehicle Scheduling Problem (SD-VSP). On the other hand, it is referred as Multi-Depot Vehicle Scheduling Problem (MD-VSP) if there is more than one depot. SD-VSP is relatively easier to solve than MD-VSP.

Another complicating extension is the case of non-identical vehicles. This problem is referred as the Vehicle Scheduling Problem with Multiple Vehicle Types (MVT-VSP). The common vehicle types are midi bus, solo bus and articulated bus. These vehicle types differ in their capacities, average speeds, fuel consumptions, etc. Therefore, not all vehicle types may be able to serve all trips.

In VSP, the vehicle blocks are constructed using a minimum number of vehicles, where each trip is assigned exactly to one vehicle block. The common objectives are minimizing service costs by minimizing the number of vehicles used and minimizing traveling costs by avoiding unnecessary deadheads in vehicle blocks. The problem is solved daily, and the output is the set of vehicle blocks, which are the sequence of trips served by one vehicle that starts and ends in a depot. A detailed problem description will be given in Chapter 4.

2.2.3 Crew Scheduling (Bus Driver Scheduling)

The Crew Scheduling Problem (CSP) consists of the short term scheduling of the crew with the aim of generating a set of daily shifts covering all vehicle blocks. There is a set of tasks (d-trips and deadheads) arising from the vehicle blocks, and the aim is to define a sequence of these tasks as shifts in such a way that every task is assigned to a shift without any overlaps.

The feasibility of a crew shift depends on a set of operator rules and relevant regulations such as the working time limit, the number and length of breaks, the maximum driving time limit, the spread time limit, etc. Additionally, there can be distinct types of crew members having different capabilities and driving licences. In this case, the feasibility of a task-crew shift assignment depends also on crew capabilities. The commonly used solution methodology involves dividing the vehicle block into POWs that start and finish at relief points, and forming a feasible shift through a sequence of POWs satisfying all constraints. A detailed problem description will be provided in Chapter 5.

2.2.4 Crew Rostering (Driver Rostering)

The Crew Rostering Problem (CRP) consists of the long term scheduling of the crew with the aim of generating a set of crew rosters to cover the daily crew shifts. It assigns crew members to constructed shifts over a long planning horizon, generally a month or a half year. It includes days-off scheduling and shift scheduling. Days-off scheduling deals with the assignment of the rest days between working days to the crew over a planning horizon, while shift scheduling deals with the assignment of the crew to shifts. When days-off and shifts are made simultaneously, the procedure is called tour scheduling.

The shifts are usually grouped in morning, day and night shifts. Additionally, the shifts can be grouped according to required crew capabilities. In this case, the shift types are considered during the generation of rosters. Crew rosters can be cyclic or noncyclic. In cyclic schedules all crew members have the same basic schedule starting with a different day, whereas noncyclic schedules are individual. In CRP, there are some hard constraints and soft constraints, where the hard constraints consist of coverage and regulatory requirements and the soft constraints include operational and personal preferences. The coverage requirements guarantee that there are adequate number of crew members on shift at all times. The regulatory requirements ensure that the crew’s work agreement and government rules are regarded. The operational and personnel preferences are also considered for a greater crew and operator satisfaction (Nurmi, 2011).

3 LITERATURE REVIEW

The detailed literature review for the vehicle and crew scheduling problems and the integration approaches for these two problems are provided in the following three sections. Since, both vehicle and crew scheduling problems are formulated as tactical fixed job scheduling problems (TFJSP) in Chapters 4 and 5 respectively, a literature review for this problem is given in Section 3.4. Finally, the contribution of this thesis to existing literature is discussed in Section 3.5.

3.1 Vehicle Scheduling Problem

Daduna and Paixão (1995), discussed modeling approaches and their complexities for several variations of VSP such as using single depot, multiple depot and fixed number of vehicles. Bunte and Kliewer (2009) provided a detailed literature review on VSP and presented the modeling approaches for the basic single depot case and further practical extensions, including multiple depots, multiple vehicle types, time windows and route constraints.

The Single Depot Vehicle Scheduling Problem (SD-VSP) is well known to be solvable with polynomial time algorithms. Freling et al. (2001b) provided a comparison of these algorithms in terms of computational times and complexities. Commonly, this problem has been formulated as a transportation problem or a network flow problem in the literature. These two basic formulations are given in Sections 3.1.1 and 3.1.2, respectively.

VSP is extended with multiple depot and/or multiple vehicle type considerations where each trip must be performed by a subset of depots and/or vehicle types. Bertossi et al. (1987) proved that the Multi-Depot Vehicle Scheduling Problem (MD-VSP) is NP-Hard and Lenstra and Kan (1981) showed that the single depot Vehicle Scheduling Problem with Multiple Vehicle Types (MVT-VSP) is NP-Hard. Obviously, it can be concluded that the combination of these two extensions is also NP-Hard. There are many studies considering these two extensions in the literature.

Common modeling approaches for the MD-VSP are single-commodity model formulations (Carpaneto et al., 1989; Fischetti et al., 1999; Mesquita and Paixão, 1992), multi-commodity model formulations (Forbes et al., 1994; Löbel, 1998; Haghani and

Banihashemi, 2002; Mesquita and Paixão, 1999), multi-commodity model formulations based on time-space network (Kliewer et al., 2006; Gintner et al., 2005) and set partitioning model formulations (Riberio and Soumis, 1994; Hadjar et al., 2006). In multi-commodity models, a multi-graph is generated including independent networks for each depot, while all depots are modeled with a single graph in single-commodity models. Multi-commodity models based on time-space network differ from other multi-commodity models, as they aggregate possible connections between groups of compatible trips and avoid the disadvantage of explicit consideration of all possible connections between compatible trips. Set partitioning models generate all feasible schedules for vehicles usually through a column generation approach.

A common modeling approach for single depot MVT-VSP uses a multi-graph with subnetworks for each vehicle type, which is introduced by Bodin et al. (1983). This approach is also used later by Costa et al. (1995). Hassold and Ceder (2014) considered the same problem and proposed a minimum cost network flow model utilizing a set of Pareto efficient timetables. Gintner et al. (2005) considered the MVT-VSP with multiple depot for public transport bus operators as well as depot capacity restrictions. They proposed a multi-commodity model including time-space networks for each depot and vehicle-type combination, and considered the case that a vehicle may return into another depot than its source depot.

As another approach, Eliiyi et al. (2009) developed a mixed integer programming model and several heuristics for the MVT-VSP, using a TFJSP formulation with spread time limitations while considering the sequence dependent setup times (deadheads) and vehicle capacity restrictions. Furthermore, if there are restrictions on trips such as certain type of trips must be performed with a subset of all vehicle types, problem is further extended including the concept of vehicle type groups. Forbes et al. (1994) and Löbel et al. (1997) also considered this extended problem for multiple depot case but within a multi-commodity modeling approach, while Kliewer et al. (2006) and Kliewer et al. (2008) studied the multi-commodity modeling approach based on time-space networks.

3.1.1 Transportation (Quasi-Assignment) Model

Gavish and Shlifer (1979) formulated the SD-VSP as a transportation problem. A mathematical formulation by Freling et al. (2001b) is presented here.

Let $N = \{1,2,\ldots,n\}$ be the set of trips, numbered due to increasing start times, and $E = \{(i, j) \mid i < j$ compatible, $i \in N, j \in N\}$ be the set of arcs that correspond to deadhead trips. Trips i and j are compatible, if trip j can be served directly after trip i by the same vehicle without any overlaps. Depot is represented by two nodes s and t. The network is defined as an acyclic directed network $G= (V, A)$ with nodes $V= N \cup \{s, t\}$ and arcs $A= E \cup (s \text{ x } N) \cup (N \text{ x } t)$. A path from s to t in the network forms a feasible vehicle schedule for a single vehicle, and a complete feasible vehicle schedule is a set of disjoint paths from s to t such that each node in N is covered. The objective function minimizes operational costs where c_{ij} represents the operational cost of serving trip j after trip i, and commonly a function of travel durations. Additionally, fixed cost of a vehicle can be considered in the objective function. It can be incorporated into the cost of arcs (s, j) or (i, t) for all $i, j \in N$. Therefore, SD-VSP can be formulated by using the binary decision variable x_{ij} , which represents whether a trip j is directly covered after trip i:

$$\text{Minimize} \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{3.0}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A} x_{ij} = 1 \qquad \forall i \in N \tag{3.1}$$

$$\sum_{i:(i,j)\in A} x_{ij} = 1 \qquad \forall j \in N \tag{3.2}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{3.3}$$

The objective (3.0) minimizes the total operational costs. Constraints (3.1) and (3.2) are the flow conservation constraints, ensuring that exactly one deadhead is performed before each trip and one deadhead is performed after each trip. Since they define a totally unimodular restriction matrix, binary variables x_{ij} are commonly relaxed to $x_{ij} \geq 0$ (Freling et al., 2001b).

The problem above is a transportation problem, where the demand/supply of each trip node is one unit. This formulation can be extended for fixed number of vehicles *v* through the following constraints:

$$\sum_{j \in N} x_{sj} = v \tag{3.4}$$

$$\sum_{i \in N} x_{it} = v \tag{3.5}$$

Both the demand/supply amounts of depot nodes *s* and *t* are taken as *v* units as stated in constraints (3.4) and (3.5). In that case, in order to allow a vehicle being idle, an arc between the depot nodes *s* and *t* should be inserted with cost zero. In the literature, Paixão and Branco (1987) referred to this transportation model as a quasi-assignment model, since this model corresponds to a linear assignment model by not considering the nodes *s* and *t*, and the corresponding arcs.

3.1.2 Network Flow Model

Bodin et al. (1983) formulated the SD-VSP as a network flow problem. A network flow model described in Bunte and Kliewer (2009) is presented here. In this model, each trip is represented by two nodes, a start and end node, and these nodes are connected with a trip arc. The depot is represented by two nodes *s* and *t*. Arcs from the depot to trips only have operational costs. The fixed costs are modeled by a single arc leading back from *t* to *s*. A feasible flow from node *s* to *t* is also a feasible vehicle schedule.

Let AT be the set of trip arcs, A the set of arcs and N be the set of nodes. The parameter c_{ij} is the operational cost of serving trip *j* after trip *i* that is usually a function of travel durations. The decision variable x_{ij} represents the flow on the arc (i, j). Thus, SD-VSP can be formulated as a minimum cost flow problem as follows:

$$\text{Minimize} \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{3.6}$$

$$\text{s.t.} \quad \sum_{i:(i,j) \in A} x_{ij} - \sum_{i:(j,i) \in A} x_{ji} = 0 \qquad \forall j \in N \tag{3.7}$$

$$1 \le x_{ij} \le 1 \qquad \forall (i,j) \in AT \tag{3.8}$$

$$x_{ij} \ge 0 \qquad \forall (i,j) \in A \tag{3.9}$$

Objective (3.6) minimizes the total operational costs. Constraint (3.7) states that all nodes are transshipment nodes. In constraint (3.8), the lower and upper bounds on the trip arcs are equal to one in order to ensure that all trips are served. Additionally, this formulation can be extended for the case of fixed number of vehicles by setting the upper bound on the arc leading back from *t* to *s*, to the number of vehicles available (Bunte and Kliewer, 2009).

3.2 Crew Scheduling Problem

Fischetti et al. (1987) proved that the fixed job scheduling problem (FJSP) with spread time constraints is NP-Hard. In a later study, Fischetti et al. (1989) proved that the FJSP with working time constraints is also NP-Hard. As these two basic variants of the CSP with spread and working time constraints are NP-Hard, it can be concluded that the CSP is also NP-Hard. Wren and Rousseau (1995) provided an overview of the Bus Driver Scheduling Problem (BDSP), which is a kind of CSP for public bus transportation, and its several variants. The authors presented constraints and conditions in different environments and proposed various solution approaches for the BDSP. Ernst et al. (2004) provided a detailed bibliography about crew scheduling and crew rostering problems. They classified the studies related to these two problems according to the problem type, application area and solution methodology.

In the literature, one of the most used approaches to the CSP is the column generation that is introduced by Desrochers and Soumis (1989), where the master problem is a set partitioning/covering problem and the subproblem is a Resource-Constrained Shortest Path Problem (RCSPP) (Carraresi et al., 1995; Desrochers et al., 1992; Freling et al., 1999; Freling et al., 2001a, 2003; Friberg and Haase, 1999; Haase et al., 2001; Mesquita and Paias, 2008; Rousseau and Desrosiers, 1995). With this approach, the feasible shifts are generated by solving the RCSPP, where the feasibility constraints such as the maximum working time limit, the number and the length of breaks, the maximum driving duration without a break, etc. are handled as different resources. Then, a crew schedule covering all tasks is determined from the given feasible shifts by solving a set partitioning/covering problem. In some of these studies, different crew types are also considered which differ with respect to shift type such as part-time, full time, etc. Basic formulations for these approaches are given in the following sections.

Furthermore, Boschetti et al. (2004) extended the CSP with multiple depots. They considered the problem of determining the optimal shifts for a set of identical crews divided into several depots, in order to cover a set of trips regarding the working time (total duration of tasks assigned to a duty) and spread time limitations. They proposed an exact method based on a set partitioning formulation with additional constraints, as well as a bounding procedure based on Lagrangian relaxation and column generation. Their model is an extension of MD-VSP, since it corresponds to the MD-VSP when no time limits are imposed on duty duration.

Various heuristic approaches are also proposed in literature for the CSP. Shen and Kwan (2001) proposed a tabu search algorithm for the BDSP, whereas Lourenço et al. (2001) presented multi-objective metaheuristics for the same problem. The authors proposed a Greedy Randomized Adaptive Search Procedure (GRASP), a Tabu Search (TS) and a Genetic Algorithm (GA) as metaheuristics. They integrated these methods in a decision support system. Dias et al. (2002) studied on a GA implementation for the BDSP, through extending the traditional approach of set covering/partitioning formulations by considering several additional criteria. De Leone et al. (2011) proposed a mathematical formulation under special constraints forced by restrictions due to Italian transportation regulations, applied to small- or medium-sized problem instances. For larger instances, the authors proposed GRASP. Kecskeméti and Bilics (2013) proposed an integer programming and evolutionary hybrid algorithm for the BDSP. Toth and Kresz (2013) presented an efficient algorithm based on cut and join approach for the BDSP. The proposed method is divided into two phases. In the first phase, rough shifts are generated that contain only the trips and the travelling activities of the driver. In the second phase, complete shifts are generated which contain all the obligatory activities and the idle activities.

3.2.1 Set Covering / Set Partitioning Model

A set covering formulation of Desrochers and Soumis (1989) is presented below. In this formulation, global constraints such as limitations on number of shift types are also considered as well as covering all tasks. Let $M = \{1,2,\ldots,m\}$ be the set of tasks, N

= {1,2,…,n} be the set of feasible shifts and K = {1,2,…,k} be the set of shift types. The parameter c_j is the cost of shift j, which is a function of travel time and paid breaks. The parameter f_{ij} = 1 if feasible shift j covers the task i, f_{ij} = 0 otherwise. The parameter g_{kj} = 1 if feasible shift j has shift type k, g_{kj} = 0 otherwise and h_k is the upper bound for shift k. The decision variable x_j = 1 if feasible shift j is selected, x_j = 0 otherwise. The master problem of the CSP can be formulated as a set covering formulation as follows:

$$\text{Minimize} \sum_{j=1}^{n} c_j x_j \tag{3.10}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} f_{ij} x_j \geq 1 \qquad \forall i \in M \tag{3.11}$$

$$\sum_{j=1}^{n} g_{kj} x_j \leq h_k \qquad \forall k \in K \tag{3.12}$$

$$x_j \in \{0,1\} \qquad \forall j \in N \tag{3.13}$$

As shown above, the objective is to minimize the total crew costs. Constraint (3.11) ensures that each task i is covered by at least one shift. Constraint (3.12) limits the number of shift types and constraint (3.13) says that all x_j decision variables are binary. Constraint set (3.12) is called as global constraints or sometimes base constraints, and minimum/maximum limitations on shift types (full time, split, etc.) are modeled in these constraints. In the set partitioning formulation, inequalities at constraint (3.11) are replaced with equalities (Desrochers and Soumis, 1989).

3.2.2 Resource Constrained Shortest Path Problem

The Resource Constrained Shortest Path Problem (RCSPP) is a special case of the shortest path problem formulated on a graph G = (N, A), where N is the set of nodes including source s and sink t, A is the set of arcs. Additionally, there is a set of resources, R. The parameter c_{ij} is the cost of each arc (i, j) ∈ A and d_{ij}^h is the consumption of resource h along each arc (i, j) ∈ A based on the resource extension function (REF) of resource h. Irnich (2008) provides a detailed REF descriptions for real life applications.

Let W_j^h be the amount of the h^{th} resource accumulated along the path from source node s to node j. Let the interval $[a_j^h, b_j^h]$ with $a_j^h < b_j^h$ be the resource window for each node $j \in N$. A path from node s to node j is feasible, if and only if $a_j^h \leq W_j^h \leq b_j^h$. Two types of constraints can be modeled with resource windows: time window constraints when h represents time, and capacity constraints when h corresponds to the quantity of resource. Time window constraints can be defined as hard or soft. In hard time window constraints, if node j is visited before a_j^h, waiting is allowed without incurring a cost, where all arrival times after b_j^h are forbidden. In soft time window case, a penalty cost is employed when node j is not visited within the given time window (Pugliese and Guerriero, 2013).

Given a source node s and sink node t, RCSPP aims to find the minimum cost path such that W_j^h respects the feasibility window $[a_j^h, b_j^h]$ for all nodes of the path and all resources. The decision variable y_{ij}= 1 if node j is visited after node i, y_{ij} = 0 otherwise.

$$\text{Minimize} \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{3.14}$$

$$\text{s.t.} \sum_{j:(i,j)\in A} y_{ij} - \sum_{j:(j,i)\in A} y_{ji} = \begin{cases} 1, \text{if } \ i = s \\ -1, \text{if } \ i = t \\ 0, \text{ otherwise} \end{cases} \qquad \forall i \in N \tag{3.15}$$

$$y_{ij}(W_i^h + d_{ij}^h) \leq W_j^h \qquad \forall (i, j) \in A, h \in R \tag{3.16}$$

$$a_i^h \leq W_i^h \leq b_i^h \qquad \forall i \in N, h \in R \tag{3.17}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i, j) \in A \tag{3.18}$$

In the objective function (3.14), total cost of arcs is minimized. Constraint (3.15) indicates that the flow between the source node s and the sink node t must be equal to one by conserving the network flow between other nodes. Constraint (3.16) ensures that, if there is a positive flow between nodes i and j, the value of resource h at node j must be greater than or equal to the sum of the amount of the h^{th} resource accumulated along the path from source node s to node i and the consumption along arc (i, j) of resource h. Constraint (3.17) enforces the resource windows of resource h for all nodes. Finally, constraint (3.18) ensures that all decision variables are binary.

Many of the conditions on crew shifts can be formulated through resources and the constraints regarding the consumption of these resources in a shift. Common resource constraints for crew shifts are spread time, working time, break duration, number and length of piece of works. Pugliese and Guerriero (2013) provided a survey of resource constrained shortest path problems and exact solution approaches.

3.3 Vehicle and Crew Scheduling Problem

In the literature, most of the studies deal with crew scheduling and vehicle scheduling individually, as in the aforementioned review of the VSP and CSP. However, there are studies on vehicle and scheduling problem (VCSP) considering both problems at the same time either by using a sequential approach or an integrated approach. In the sequential approach, VSP and CSP are solved sequentially, while they are solved simultaneously in the integrated one.

Freling et al. (1995) introduced the first exact formulation for integrated vehicle and crew scheduling problem (IVCSP), as well as an optimization-based heuristic approach. Their formulation includes a quasi-assignment structure for vehicle scheduling, set partitioning constraints for crew scheduling and a set of linking constraints between vehicle and crew scheduling. They proposed a column generation algorithm combined with Lagrangian relaxation as a solution approach. The proposed formulation and solution methodology including the computational results for the real world problems can be found in Freling et al. (1999).

Freling et al. (2003) studied on the relaxations and algorithms for the integrated approach to solve the single depot VCSP. They discussed the potential benefits of integration and provided an overview of the literature that considers partial integration. They proposed a mathematical formulation and a solution methodology based on column generation combined with Lagrangian relaxation of the IVCSP. Furthermore, they applied their techniques to real life problems of a public transport operator. According to Freling et al. (2003), three main approaches are pursued for the integration of vehicle and crew scheduling in the literature. The first one is the scheduling of vehicles by a heuristic approach, which is designed by Ball et al. (1983) for the crew scheduling problem. The second approach for the integration is determining the vehicle schedules by taking crew costs into consideration, which is proposed by Darby-Dowman et al. (1988). The third approach is the complete integration of vehicle and crew scheduling, which is proposed by Freling et al. (1995).

Freling et al. (2001a) studied on a practical application of the single depot IVCSP. They considered the bus lines of a public transport operator, where particular constraints on shifts and breaks should be taken care of. They proposed both the sequential and integrated approaches for the VCSP and compared them with each other. As for the sequential approach for the vehicle scheduling, they developed a transportation model where nodes are trips, and the objective is to find the set of disjoint paths between two depots respecting the operator-specific roundtrip condition. They solved this model using subgradient optimization and Lagrangian relaxation of roundtrip condition. After solving this model, they proposed a two-step procedure for the crew scheduling. In the first step, they generated shifts by defining all relief points and generating all feasible pieces of works, sequentially. In the second step, they selected the optimal shifts by solving a set covering model. They proposed a column generation method for solving this problem. As an integrated approach for the VCSP, they proposed a formulation which is the combination of the quasi-assignment formulation for vehicle scheduling and a set partitioning formulation for crew scheduling. The quasi-assignment part assures the feasibility of vehicle schedules, while the set partitioning assures that each trip and deadhead is assigned to a shift if its corresponding deadhead is part of the vehicle schedule. They proposed a column generation algorithm for the integrated model. Furthermore, they investigated the impact of allowing drivers to change vehicle during a break. According to their current operational rules, these changeovers are only allowed in split duties. They compared the sequential and integrated approaches with different variants of allowing changeovers. According to their computational results, the integrated approach gave better results than the sequential approach for all the variants.

Friberg and Haase (1999) introduced the first exact solution approach for the IVCSP, which combines the crew scheduling approach of Desrochers and Soumis (1989) and the vehicle scheduling approach by Riberio and Soumis (1994). They proposed a branch-price-and-cut algorithm, where a column generation is performed for generating both vehicle and crew schedules in order to obtain optimal solutions. However, only small instances including up to 20 trips could be solved to optimality within 3 hours. Later, Haase et al. (2001) proposed an exact approach for the single depot IVCSP with homogenous fleet, which relies on a set partitioning formulation for the CSP that combines side constraints for the vehicle schedules. In their approach, column generation integrated into a branch and bound scheme is applied only for the generation of crew schedules. The side constraints on vehicles guarantee that optimal schedules can be derived afterwards in polynomial time. They also proposed a heuristic approach for larger instances.

Based on the integrated approach of Freling et al. (1995), Huisman (2004) and Huisman et al. (2005) introduced the first formulation for the multiple depot IVCSP. Later, Mesquita and Paias (2008) presented an integer linear programming formulation combining a multi-commodity network flow model with a set partitioning/covering model for the same problem. They proposed a four-step solution approach for the problem. In the first two steps, the set of tasks are defined and initial set of shifts is generated. In the third step, linear programming relaxation of the model is solved using a column generation scheme. In the fourth step, if the resulting solution is not integer, branch and bound techniques are used over the set of generated shifts in order to obtain a feasible solution. Later, Mesquita et al. (2009) developed and compared additional branching strategies for solving the model proposed by Mesquita and Paias (2008). Gintner et al. (2006), Steinzen (2007) and Steinzen et al. (2010) developed a time-space network structure for the multiple depot IVCSP and used the general solution scheme proposed by Freling (1997) in order to solve the model.

Kliewer et al. (2012) extended the multiple depot IVCSP with time windows for scheduled trips. They developed a mathematical model based on a time-space network and solved this model with column generation combined with Lagrangian relaxation. As another extension, Mesquita et al. (2013) studied on the vehicle-crew-roster problem with days-off pattern, which simultaneously determines the minimum cost vehicle and daily crew schedules that cover all trips, and the minimum cost roster that cover all daily crew shifts according to the given predetermined days-off pattern. They proposed an integer linear programming formulation and a heuristic solution approach based on Bender’s decomposition, which iterates between the solution of the IVCSP and the solution of the rostering problem. Gintner et al. (2008) presented a partially integrated approach for the multiple depot VCSP. Contrary to the traditional sequential approach, the crew scheduling in their method is based on a set of optimal vehicle schedules instead of only one optimal schedule. They used a multi-commodity flow formulation based on a time-space network for the MD-VSP, and a column generation approach combined with Lagrangian relaxation for the CSP.

3.4 Tactical Fixed Job Scheduling Problem

As for the studies on the FJSP, Kolen et al. (2007) provided a detailed survey on the Interval Scheduling Problem (ISP), such that each job is started at or after its ready time and completed before its deadline. If the job can be started after its ready time, the problem is defined as the Variable Job Scheduling Problem (VJSP). Conversely, if the job cannot be started after its ready time, the problem is referred as the FJSP. Kovalyov et al. (2007) provided a more recent review on models, applications and algorithms for the FJSP. The problem has two variants according to the objective functions: Operational Fixed Job Scheduling Problem (OFJSP) and Tactical Fixed Job Scheduling Problem (TFJSP). In the OFJSP, the objective is to maximize the weighted number of processed jobs with a given number of processors where each job has a weight. Conversely, in the TFJSP, the objective is the minimization of the total cost of machines required to cover all jobs. Eliiyi and Azizoğlu (2009, 2010, 2011) proposed solution techniques for the OFJSP with eligibility, working time and spread time limitations.

Many studies exist in literature on the TFJSP variants. Fischetti et al. (1989) studied the TFJSP with working time constraints. The authors dealt with the BDSP with working time constraints, where the objective is to find the minimum number of crew required to cover all daily tasks. They proposed a polynomial algorithm for the preemptive version of the problem, and developed a branch and bound algorithm. They also provided some lower bounds and dominance criteria for the problem. Fischetti et al. (1987) studied on the TFJSP with a constraint on the spread time of the bus drivers where each driver has a spread time limit from the start time to the end time of his shift, including the idle times. They proposed polynomial procedures to obtain some lower bounds, as well as dominance criteria and reductions in problem size. They also developed a branch and bound algorithm for the problem.

As another extension for the TFJSP, eligibility constraints are considered where each job can be processed only by a subset of the machine classes. Arkin and Silverberg (1987) showed that the FJSP with eligibility constraints is NP-Complete. Kroon et al. (1997) studied on the TFJSP with eligibility constraints, where the objective is to determine the minimum number of non-identical parallel machines in such a way that each machine can process only jobs from a subset of the job classes and preemption is not allowed. They provided exact and approximation algorithms for the problem. Eliiyi et al. (2009) developed a mixed integer programming model and several heuristics for the vehicle scheduling problem, using a TFJSP formulation with spread time limitations while considering the sequence dependent setup times and vehicle capacity restrictions. Zhou et al. (2014) developed a branch and price algorithm for the TFJSP with spread time and eligibility constraints. Krishnamoorthy et al. (2012) studied on the shift minimization personnel task scheduling problem which is similar to the TFJSP with eligibility constraints, except that the machines have also availability limitations. They provided mathematical formulations and a heuristic approach.

3.5 Discussion

As it has been elaborated in Section 3.1, VSP is commonly formulated as a network flow or transportation model. In the multiple depot and multiple vehicle type cases, multi-commodity model formulations are generally used, where a multi-graph is generated including subnetworks for each depot/vehicle type. In this thesis, a fixed job scheduling approach is proposed for the VSP by considering multiple vehicle types as eligibility constraints.

Furthermore, as discussed in Section 3.4, the common objective of the TFJSP formulations is to minimize the total fixed cost of machines required to cover all jobs. In this thesis, variable cost minimization is also handled within the fixed job scheduling framework. Thus, the proposed formulation is different from the existing fixed job scheduling literature.

In summary, the approach for the VSP in this thesis is different from the existing vehicle scheduling and tactical fixed job scheduling studies in literature due to:

(1) The fixed job scheduling approach to the vehicle scheduling problem,

(2) The consideration of eligibility constraints (multiple vehicle types), and

(3) Minimization of both fixed and variable costs.

As mentioned in Section 3.2, column generation approach is commonly used for the CSP, where the master problem is a set partitioning/covering problem and the subproblem is RCSPP. In studies using this approach, different crew types are considered that differ with respect to shift types such as part-time, full time, etc. However, different crew types with different capabilities are not considered, namely the cases where each task can be assigned to certain subset of crew types. In this thesis, a fixed job scheduling approach is proposed for the CSP considering different crew types with different capabilities as eligibility constraints.

Furthermore, as discussed in Section 3.4, the working time constraint is not considered with sequence dependent setup times in the TFJSP formulations. In this thesis, this constraint is handled by using a TFJSP based approach that includes the sequence dependent setup times, since the drivers must travel between the start and end locations of the tasks in order to fully perform the assigned tasks.

In summary, the approach for the CSP in this thesis is different from the existing crew scheduling and tactical fixed job scheduling studies in the related literature due to:

(1) The fixed job scheduling approach to the crew scheduling problem,

(2) The consideration of many realistic constraints such as working time, spread time and eligibility constraints, and

(3) The handling of the working time constraint in the existence of sequence-dependent setup times.

As stated in Section 3.3, column generation algorithms combined with Lagrangian relaxation are commonly used for the VCSP as a solution approach. Furthermore, multiple vehicle types (eligibility constraints for vehicles) are employed in some of these studies on the multiple depot VCSP (Huisman, 2004; Huisman et al., 2005; Gintner et al. 2006; Steinzen, 2007; Steinzen et al. 2010; Kliewer et al. 2012), by referring to vehicle type as depot. However, all of these studies assume that each crew belongs to a specific depot and can only cover tasks performed by the vehicles from this certain depot. According to this assumption, crew member can be assigned to only one type of vehicle/depot. Thus, eligibility constraints are not considered for crew members in these studies, where each crew member can be assigned to more than one type of task (vehicle) based on his capabilities.

In this thesis, a sequential approach and an integrated formulation are proposed for the VCSP that is different from the existing vehicle and crew scheduling studies in literature due to:

(1) The fixed job scheduling based sequential approach for the VCSP,

(2) A binary programming model for the integrated VCSP,

(3) The consideration of eligibility constraints for both vehicles and crew members, and

(4) The consideration of realistic constraints including working time and spread time for the crew.

4 THE VEHICLE SCHEDULING PROBLEM

This thesis is motivated by a real life vehicle scheduling problem of a public bus transportation authority, which has a heterogeneous fleet of buses. The trips are given by specified time intervals, as well as start and end termini. Furthermore, the trips may require distinct types of vehicles that have different characteristics such as capacity, average speed, fuel consumption, etc. Therefore, there are several vehicle classes corresponding to eligibility constraints in the problem. The objective includes the minimization of total fuel consumption cost arising from deadheads and trips, as well as the fixed cost of vehicles.

Based on these characteristics, the VSP in this thesis can be formulated as a Tactical Fixed Job Scheduling Problem (TFJSP), as the ready times and deadlines of the trips are fixed in advance, and the objective is to minimize the total fixed cost of vehicles to cover all trips and the total variable costs resulting from deadheads and trips. In the basic Fixed Job Scheduling Problem (FJSP), there are n independent tasks ready to be processed on m parallel resources. The time window of each task is defined by a ready time and a deadline. The tasks cannot be delayed after their ready times, meaning that the processing times of the tasks are equal to their corresponding time windows. The TFJSP is a variant of the FJSP where the objective is to minimize the total fixed cost of the resources required to process all tasks. In this thesis, additional eligibility constraints further complicating the problem are considered. Moreover, the existence of the additional objective function of minimizing the total variable costs arising from deadheads and trips diverge the problem from a typical TFJSP, further complicating the structure. A TFJSP-based binary programming model is formulated for the real life VSP mentioned above. Before presenting the mathematical formulation, the problem notation and the assumptions are stated.

4.1 Assumptions and Notation

The trips can demand different types of vehicles such as mini bus, solo bus, or articulated bus according to their demand densities and physical constraints of the routes. For instance, a trip requiring articulated buses cannot be performed with a solo bus. Therefore, each vehicle is eligible to perform only a subset of trips.

The VSP can be defined as a TFJSP where the ready times and deadlines of the trips are fixed in advance, and the objective is to minimize the total fixed cost of the vehicles and the total variable costs resulting from the deadheads and the trips. All trips must be covered satisfying the eligibility constraints. The notation for the problem formulation is given in Table 4.1.

Table 4.1 VSP Notation.

Sets	
K	Set of trips
D	Set of trip/vehicle classes
V	Set of vehicles (buses)
V_d	Set of buses belonging to class $d \in D$; $V_d \subset V$
K_d	Set of trips belonging to class $d \in D$; $K_d \subset K$
IV_i	Set of incompatible trips for trip $i \in K$; $IV_i \subset K$
AV_d	Set of trips that can be performed by a bus belonging to class $d \in D$; $K_d \subseteq AV_d$
BV_d	Set of buses that can perform trips belonging to class $d \in D$; $V_d \subseteq BV_d$
Parameters	
r_i	Ready time of trip $i \in K$
e_i	Deadline of trip $i \in K$
a_{ij}	Setup time between trip i and j, $i, j \in K$
o_v	Trip based fuel consumption cost of vehicle $v \in V$, per time unit
s_v	Deadhead based fuel consumption cost of vehicle $v \in V$, per time unit
t_v	Fixed cost of vehicle $v \in V$
Decision Variables	
x_i^v	1 if eligible trip i is assigned to vehicle v, 0 otherwise
z_v	1 if vehicle v is used, 0 otherwise
w_i^v	1 if eligible trip i is first trip of vehicle v, 0 otherwise
p_i^v	1 if eligible trip i is last trip of vehicle v, 0 otherwise
h_{ij}^v	1 if eligible trip j is covered consecutively after eligible trip i by vehicle v, 0 otherwise

The assumptions are listed below.

- The problem is solved daily. The timetabling problem is assumed to be solved in advance; hence, the ready times and deadlines of the trips are fixed and known. The deadlines of the trips are defined as their ready times plus their constant processing times.
- Trip preemption is not allowed. A trip must be performed wholly by a single bus.
- Delay and cancellation are not possible, all trips must be performed/covered.
- There are several trip classes that require certain vehicle types. There is a predefined subset of eligible trips for each vehicle class. These sets are inclusive; for any two vehicle classes d and $f \in D$, either $AV_d \subseteq AV_f$ or $AV_f \subseteq AV_d$.
- Each vehicle has a daily fixed cost. When a vehicle is assigned to a trip, its fixed cost is incurred and the privilege to use the vehicle for a day is obtained. Fixed

costs are different for each vehicle class and determined based on the daily depreciation cost of vehicles by the operator.

- Each vehicle type has different trip and deadhead fuel consumption costs. It is assumed that vehicles consume more fuel during deadheads compared to trips, as the deadheads are performed at higher speeds. Hence, deadhead fuel consumption cost is higher than the trip cost for all vehicles.
- There are sufficient number of vehicles in each class to perform all trips.
- Sequence-dependent setup (deadhead) times between trips are deterministic.
- The setup time between any two trips is determined by the shortest path between them. There is no setup time before first trip and after the last trip in a vehicle schedule.

The trips are numbered in increasing order based on their ready times, meaning that $r_i \leq r_j$ for all trips $i < j$. Sets of vehicles belonging to class d (V_d) are disjoint subsets of the vehicle set V. To make sure that there are sufficient number of vehicles in each class to perform all trips, an upper bound is computed for the number of vehicles belonging to each class through a simple greedy heuristic, where the trips are grouped based on their classes and the number of vehicles to cover all trips in each class is determined in a greedy manner with extra allowances.

Sets of trips belonging to class d (K_{d}) are disjoint subsets of the trip set K. However, the set of trips that can be performed by a vehicle belonging to class d (AV_d) is an inclusive set, meaning that a vehicle can perform trips of all lower classes beside its own class. For instance, a solo bus cannot be assigned to trips requiring articulated buses, but an articulated bus can be assigned to both trip classes. The decision variables are defined only for eligible vehicle-trip assignments. Similarly, the set of vehicles that can perform trips belonging to class d (BV_d) is an inclusive set, meaning that a trip can be performed by a bus of a higher class beside its own class. For instance, a trip requiring a solo bus can be assigned to a solo bus or an articulated bus. However, a trip requiring an articulated bus can be assigned to only an articulated bus. An incompatibility set for each vehicle is defined in order to handle the time overlaps between trips. The reason for any trip pair (i,j) not to be assigned to the same vehicle is that, the time windows of the two trips overlap ($r_i \leq r_j < e_i$), or the setup time (deadhead) in between ($r_i \leq r_j < e_i + a_{ij}$). According to these observations, the set of incompatible trips for trip $i \in K$, namely IV_i, is defined as follows:

$$IV_i = \left\{ j \in K, i < j; r_j < e_i + a_{ij} \right\}$$

4.2 Formulation

The binary programming model of the proposed VSP is given below:

$$\text{Minimize} \sum_{d \in D} \sum_{v \in V_d} (t_v z_v + o_v (\sum_{i \in AV_d} (e_i - r_i) x_i^v) + s_v (\sum_{i \in AV_d} \sum_{j \in AV_d | j > i} a_{ij} h_{ij}^v)) \qquad (4.0)$$

s.t. $$\sum_{v \in BV_d} x_i^v = 1 \qquad \forall i \in K_d, d \in D \qquad (4.1)$$

$$x_i^v + x_j^v \leq 1 \qquad \forall i \in AV_d, j \in IV_i \cap AV_d, v \in V_d, d \in D \qquad (4.2)$$

$$x_i^v \leq z_v \qquad \forall i \in AV_d, v \in V_d, d \in D \qquad (4.3)$$

$$\sum_{v \in BV_d} w_j^v + \sum_{f \in D} \sum_{v \in (V_f \cap BV_d)} \sum_{i \in AV_f | i < j} h_{ij}^v = 1 \qquad \forall j \in K_d, d \in D \qquad (4.4)$$

$$\sum_{v \in BV_d} p_i^v + \sum_{f \in D} \sum_{v \in (V_f \cap BV_d)} \sum_{j \in AV_f | j > i} h_{ij}^v = 1 \qquad \forall i \in K_d, d \in D \qquad (4.5)$$

$$\sum_{i \in AV_d | i < j} h_{ij}^v \leq x_j^v \qquad \forall j \in AV_d, v \in V_d, d \in D \qquad (4.6)$$

$$\sum_{j \in AV_d | j > i} h_{ij}^v \leq x_i^v \qquad \forall i \in AV_d, v \in V_d, d \in D \qquad (4.7)$$

$$\sum_{i \in AV_d} w_i^v = z_v \qquad \forall v \in V_d, d \in D \qquad (4.8)$$

$$\sum_{i \in AV_d} p_i^v = z_v \qquad \forall v \in V_d, d \in D \qquad (4.9)$$

$$w_i^v \leq x_i^v \qquad \forall i \in AV_d, v \in V_d, d \in D \qquad (4.10)$$

$$p_i^v \leq x_i^v \qquad \forall i \in AV_d, v \in V_d, d \in D \qquad (4.11)$$

$$x_i^v, z_v, w_i^v, p_i^v, h_{ij}^v \in [0,1], \quad \forall i \in AV_d, j \in AV_d \mid j > i, v \in V_d, d \in D \qquad (4.12)$$

The objective function (4.0) minimizes the total fixed and variable costs of the vehicles. Constraint set (4.1) ensures that all trips are covered by an eligible vehicle. Constraint set (4.2) guarantees that incompatible trip pairs are not assigned to the same vehicle. Constraint set (4.3) ensures that a trip can be assigned to a vehicle only if it is used. Constraint sets (4.4) and (4.5) guarantee that all trips must be scheduled to an eligible vehicle; that is, there must be an arrival to and a departure from each trip. Constraint sets (4.6) and (4.7) ensure that there can be a deadhead between any two trips only if both of these trips are assigned to same vehicle. Constraint sets (4.8) and (4.9) guarantee that there must be a first and a last trip for each used vehicle. Constraint sets (4.10) and (4.11) ensure that a trip can be the first/last trip of an eligible vehicle only if it is assigned to that vehicle. Finally, (4.12) defines the decision variables.

4.3 Computational Study

In this section, a preliminary computational experiment is described to evaluate the performance of the proposed formulation in Section 4.2. A similar setting to the one described in Fischetti et al. (1987) is used for computational experimentation. Random test problems are generated for trip sets with K = 20, 40, 80 and 120.

It is assumed that there are 200 time units in a day. There are three trip/vehicle classes due to vehicle types: midi bus, solo bus and articulated bus. A midi bus can be assigned to trips only requiring midi buses. However, a solo bus can be assigned to trip classes requiring midi and solo buses. An articulated bus is assumed to be eligible to perform all trips. There are two settings for trip-class assignments. In the first setting (c=1), 25% of the trips belong to class 1 (midi bus), 50% of the trips belong to class 2 (solo bus) and the remaining 25% belong to class 3 (articulated bus). In the second setting (c=2), 20% of the trips belong to class 1, 40% of the trips belong to class 2 and the remaining 40% belong to class 3.

Two sets of ready times are generated. In the first set (r=1), the ready times follow a discrete uniform distribution in the range [0,200]. In the second set (r=2), a peak time is considered where 25% of ready times follow a discrete uniform distribution in the range [30, 50], 25% come from the range [120,160], and the remaining 50% from ranges [0,29], [51,119] and [161,200].

Two sets of distributions are used for the processing times of trips; (p=1) discrete uniform distribution in the range [4,13] and (p=2) triangular distribution in the range [4,9,13]. The setup times (deadheads) are uniform in the range [0,5]. Fixed cost of vehicles in each class are set as 60, 80 and 120 monetary units, respectively. Trip fuel consumption costs of vehicles in each class are taken as 3.5, 5 and 7 monetary units per time unit, whereas the deadhead fuel consumption costs of vehicles in each class are assumed to be 4.5, 6.5 and 9 monetary units per time unit, respectively.

For each problem combination, 10 random test problems are generated, summing up to 320 problem instances in total. The prime modulus multiplicative linear congruential generator (PMMLCG) by Law and Kelton (1991, pp. 449-457) is used with default seeds to generate the random numbers. The binary programming formulation in Section 4.2 is modeled in C++ programming language on Microsoft Visual Studio platform, and IBM ILOG Concert Technology is used for solving the model with IBM ILOG CPLEX 12.6. All test problems are optimally solved on a Core i7, 2.60 GHz, 8 GB RAM computer. Computational results for each problem combination are reported in Table 4.2. The table represents the minimum, average and maximum solution times over 10 instances for each setting. Bus set sizes for each vehicle class are generated to be nonbinding for all instances.

As shown in Table 4.2, the proposed model formulation is quite effective in terms of solution time for instances with up to 120 trips. For all instances with up to 80 trips, the optimal solutions are obtained in less than 30 seconds. Among instances with 120 trips, the maximum solution time is around 4 minutes.

As observed from the table, the average solution times for settings (1,2,1), (1,2,2), (2,2,1) and (2,2,2) are higher than others. Especially, settings (1,2,1) and (1,2,2) have the highest solution times. Hence, it can be said that the instances with ready times considering peak hours (r = 2) are relatively harder to solve. Furthermore, it can be concluded that proposed formulation performs robustly under different vehicle type combinations and processing time distributions.

Table 4.2 Computational Results for the VSP model.

c	*r*	*p*	*K*	Runtime (Sec.)		
				Min.	**Avg.**	**Max.**
1	1	1	20	0.235	0.322	0.500
			40	0.657	0.997	1.609
			80	6.093	9.572	22.594
			120	42.265	55.731	79.469
1	1	2	20	0.250	0.352	0.563
			40	0.656	0.908	1.406
			80	6.719	9.770	14.453
			120	34.547	57.139	89.032
1	2	1	20	0.375	0.500	0.829
			40	0.891	1.203	2.141
			80	10.234	19.638	28.391
			120	75.547	110.459	222.562
1	2	2	20	0.297	0.335	0.516
			40	0.578	0.964	1.610
			80	7.563	10.775	13.672
			120	53.281	70.736	111.812
2	1	1	20	0.234	0.380	0.656
			40	0.610	0.827	1.516
			80	5.625	9.905	27.688
			120	37.313	53.916	65.546
2	1	2	20	0.235	0.316	0.593
			40	0.594	0.713	1.140
			80	6.047	8.616	15.625
			120	31.969	53.255	145.703
2	2	1	20	0.234	0.369	0.453
			40	0.641	0.995	1.625
			80	4.875	8.244	10.468
			120	45.937	66.833	92.109
2	2	2	20	0.266	0.316	0.578
			40	0.750	0.978	1.672
			80	6.671	9.890	22.687
			120	45.906	59.644	71.437

5 THE CREW SCHEDULING PROBLEM

In the Crew Scheduling Problem (CSP) of the bus operator, each driver has a spread time limit from the start time to the end time of his/her shift including the idle times. Furthermore, a driver cannot exceed the maximum total working time limit. The processing times of the tasks assigned to drivers are included in their working times, as well as the sequence-dependent setup times. The tasks require different types of vehicles and different crew capabilities. Therefore, several crew classes exist based on competencies required to use certain vehicle types, corresponding to the eligibility constraints in the problem. The objective of the CSP is to minimize the total fixed cost of the crew such that all tasks are covered satisfying the operational constraints.

The CSP studied in this thesis is formulated as a Tactical Fixed Job Scheduling Problem (TFJSP), where the ready times and deadlines of the tasks are fixed in advance, and the objective is to minimize the total cost of crew to cover all tasks satisfying the eligibility constraints, working and spread time limitations. Before presenting the mathematical formulation, problem notation and the assumptions are stated briefly in Section 5.1.

5.1 Assumptions and Notation

Each driver has a spread time limit from the start time to the end time of his/her shift. This limitation is defined as an upper bound on the total time elapsing from the start time of the first task assigned to that driver until the end time of the last task. The drivers can be idle at some time intervals during their shift; these idle times are also included in the spread time. Furthermore, a driver cannot exceed the maximum total working time limit. The processing times of the tasks assigned to the crew are included in the working time, as well as the sequence-dependent setup times, as the drivers must travel between the start and end locations of the tasks in order to perform the assigned tasks.

Different types of vehicles require different crew capabilities. For instance, a driver licensed to drive solo buses may not be licensed to drive an articulated bus. Therefore, each driver is eligible to perform only a subset of tasks. The notation for the problem formulation is given in Table 5.1.

Table 5.1 CSP Notation.

Sets	
L	Set of tasks
D	Set of crew/task classes
Q	Set of crew members (drivers)
Q_d	Set of drivers belonging to class $d \in D$; $Q_d \subset Q$
L_d	Set of tasks belonging to class $d \in D$; $L_d \subset L$
I_i	Set of incompatible tasks for task $i \in L$; $I_i \subset L$
A_d	Set of tasks that can be performed by a crew member belonging to class $d \in D$; $L_d \subseteq A_d$
B_d	Set of crew members that can perform tasks belonging to class $d \in D$; $Q_d \subseteq B_d$
Parameters	
r_i	Ready time of task $i \in L$
e_i	Deadline of task $i \in L$
a_{ij}	Setup time between task i and j, $i, j \in L$
W	Working time limit
S	Spread time limit
c_k	Fixed cost of driver $k \in Q$
Decision Variables	
x_i^k	1 if eligible task i is assigned to driver k, 0 otherwise
y_k	1 if driver k is used, 0 otherwise

The assumptions for the VSP are valid for the CSP, as well. The additional assumptions are listed below.

- The problem is solved daily. The crew rostering problem is assumed to be solved in advance. Hence, the crew availabilities during the working week are known, along with days-off schedules and shifts.
- The vehicle scheduling problem is assumed to be solved. Task preemption is not allowed. A task must be performed wholly by a single crew member.
- There are several task and crew classes based on vehicle requirements and driver licences. There is a predefined subset of eligible tasks for each crew class. These sets are inclusive; for any two crew classes d and $f \in D$, either $A_d \subseteq A_f$ or $A_f \subseteq A_d$.
- Only full time drivers are considered. Each driver has a daily fixed cost. When a driver is assigned to a task, his/her fixed cost is incurred and the privilege to use the driver for a day is obtained. Fixed costs are different for each crew class.
- Delay and cancellation are not possible. All tasks should be covered by the crew and there are sufficient drivers in each class.
- Sequence-dependent setup (travel) times between tasks are deterministic, and is determined by the shortest path between the tasks. There is no setup time before first task and after the last task in a crew schedule.
- Spread and working time limits are constant, known and the same for all drivers.

The tasks are numbered in increasing order based on their ready times, meaning that $r_i \leq r_j$ for all tasks $i < j$. Sets of drivers belonging to class d (Q_d) are disjoint subsets of driver set Q. It is assumed that a sufficient number of drivers exist in each crew class. The upper bounds on the number of drivers belonging to class d are determined as nonbinding values by using a simple greedy heuristic approach. For this purpose, the tasks are grouped based on their task classes, and the required number of drivers to cover all tasks in each task class is determined in a greedy manner with extra allowances.

Sets of tasks belonging to class d (L_d) are disjoint subsets of the task set L. However, the set of tasks that can be performed by a crew belonging to class d (A_d) is an inclusive set, meaning that a crew member can perform tasks of lower classes beside its own class. For instance, a driver licensed to drive a solo bus cannot be assigned to tasks requiring articulated buses. However, a driver licensed to drive an articulated bus can be assigned to any task class. Decision variables are defined only for eligible crew-task assignments. Additionally, the set of drivers that can perform tasks belonging to class d (B_d) is an inclusive set, meaning that a task can be performed by a driver of a higher class beside its own class. For instance, a task with a solo bus can be assigned to a driver licensed to drive a solo bus or one licensed to drive an articulated bus.

An incompatibility set for each task is defined in order to handle the spread time constraints and the overlaps. The set of tasks that cannot be performed by the same driver are determined through these sets. There may be two reasons for any task pair (i,j) not to be assigned to the same crew. The first reason is that, the time windows of the two tasks overlap ($r_i \leq r_j < e_i$), or a crew cannot perform the two tasks consecutively due to the setup time in between ($r_i \leq r_j < e_i + a_{ij}$). The second reason could be that the difference between the ready time of first task and the deadline of second task exceeds the total spread time limit ($e_j - r_i > S$). According to these observations, the set of incompatible tasks for task $i \in L$, namely I_i, is defined as follows:

$$I_i = \left\{ j \in L, j > i; r_j < e_i + a_{ij} \;\; or \;\; e_j - r_i > S \right\}$$

5.2 Formulation

Based on the above assumptions and definitions, the binary programming model of the CSP is given below:

$$\text{Minimize} \sum_{k \in Q} c_k y_k \tag{5.0}$$

$$\text{s.t.} \quad \sum_{k \in B_d} x_i^k = 1 \qquad \forall i \in L_d, d \in D \tag{5.1}$$

$$x_i^k + x_j^k \leq 1 \qquad \forall i \in A_d, j \in I_i \cap A_d, k \in Q_d, d \in D \tag{5.2}$$

$$\sum_{i \in A_d} (e_i - r_i) x_i^k \leq W y_k \qquad \forall k \in Q_d, d \in D \tag{5.3}$$

$$x_i^k, y_k \in [0,1], \forall i \in A_d, k \in Q_d, d \in D \tag{5.4}$$

The objective function (5.0) minimizes the total fixed crew cost. Constraint set (5.1) ensures that all tasks are covered. Constraint set (5.2) guarantees that incompatible task pairs are not assigned to the same driver. Note that the setup times between tasks are considered in the definition of the incompatibility set. Thus, the setup time parameter (a_{ij}) and the spread time parameter (S) do not appear in the formulation. Constraint set (5.3) ensures that the sum of the processing times of tasks assigned to a driver does not exceed the total working time limit. Finally, constraint set (5.4) defines the necessary binary variables.

As mentioned before, the approach in this thesis is different from the crew scheduling studies in literature due to the handling of the working time constraint in the existence of sequence-dependent setup times. As shown in the formulation, the sum of the processing times of tasks assigned to a driver is not allowed to exceed the total working time limit. However, in practice the sequence-dependent setup times should also be included in the working time. In order to avoid defining an additional sequence variable, which increases the model size and the solution time considerably, an *iterative valid inequality generation schem*e is proposed, which eliminates the task sequences violating the total working time when the setup times are included. The procedure is based on repetitive solutions of the simpler and fast model above, where possible infeasible solutions are removed iteratively, as they are encountered. The model is solved initially with working time constraints including only the processing times of the tasks. Then, valid inequalities are generated (if needed) for infeasible task sequences exceeding the total working time limit when sequence-dependent setup times are included. The generated inequalities ensure that the tasks forming an infeasible sequence cannot be assigned to same driver; they are added to the formulation and the model is re-solved. The procedure is automatically repeated until there is no infeasible sequence in the solution.

The approach for valid inequality generation is illustrated through an example given in Table 5.2. When the model is solved with a working time limit of 80 time units, an infeasible task sequence of 1-2-3-4-5-7-8 is obtained for a driver, with a total working time of 92 time units when the setup times are added. Therefore, a valid inequality for this sequence is generated, and added to the model. In addition, to avoid any infeasibilities that may occur in further iterations of the procedure, all infeasible subsequences of the original infeasible sequence (namely, 1-2-3-4-5-6, 2-3-4-5-6-7, and 3-4-5-6-7-8) are also eliminated at this same iteration through generating additional cuts, as shown in Table 5.2.

Table 5.2 Valid Inequality Generation Example for the CSP.

Original Infeasible Sequence: 1-2-3-4-5-7-8, Total Working Time: 92 and Working Time Limit: 80					
Subsequence	**Total Working Time (Time Units)**	**Subsequence**	**Total Working Time (Time Units)**	**Subsequence**	**Total Working Time (Time Units)**
1-2	25	2-3	30	3-4	27
1-2-3	42	2-3-4	44	3-4-5	45
1-2-3-4	56	2-3-4-5	62	3-4-5-6	57
1-2-3-4-5	74	2-3-4-5-6	74	3-4-5-6-7	68
1-2-3-4-5-6	**86**	**2-3-4-5-6-7**	**85**	**3-4-5-6-7-8**	**84**
Inequalities are generated for subsequences: (1-2-3-4-5-6); (2-3-4-5-6-7); (3-4-5-6-7-8)					

5.3 Computational Study

In this section, the computational experiment is described to evaluate the performance of the proposed solution procedure for the CSP. A similar setting to the one described in Fischetti et al. (1987) is used. Random test problems are generated for task sets with L = 40, 80 and 120.

It is assumed that there are 200 time units in a day, and a task can be either a d-trip or a deadhead. There are three task/crew classes due to vehicle types; midi bus, solo bus and articulated bus. A driver licensed to drive a midi bus can be assigned to tasks only with midi buses. However, a driver licensed to drive a solo bus can be assigned to two classes; with midi and solo buses. A driver licensed to drive an articulated bus is eligible to drive midi, solo and articulated buses.

There are two settings for task-class assignments. In the first setting (c=1), 25% of the tasks belong to class 1 (midi bus), 50% of the tasks belong to class 2 (solo bus) and the remaining 25% belong to class 3 (articulated bus). In the second setting (c=2), 40% of the tasks belong to class 1, 40% of the tasks belong to class 2 and the remaining 20% belong to class 3. There are two settings for the task types. In the first setting (t=1) 40% of the tasks are deadhead and the remaining 60% are d-trips, while in the second setting (t=2) 20% of the tasks are deadhead and the remaining 80% are d-trips. Two sets of ready times are generated. In the first set (r=1), the ready times follow a discrete uniform distribution in the range [0,200]. In the second set (r=2), a peak time is considered where 25% of ready times follow a discrete uniform distribution in the range [30, 50], 25% come from the range [120,160], and the remaining 50% from ranges [0,29], [51,119] and [161,200]. Two sets of distributions are used for the processing times of d-trips; (p=1) discrete uniform distribution in the range [3,13] and (p=2) triangular distribution in the range [3,8,13]. The duration of deadheads in all sets are uniform in the range [1,5], and setup times are uniform in the range [0,5]. Time limits are set as S=100 and W=80. Fixed cost of drivers in each class are determined as 160, 180 and 200 monetary units, respectively.

For each problem combination, 10 random test problems are generated, summing up to 480 problem instances in total. The prime modulus multiplicative linear congruential generator (PMMLCG) by Law and Kelton (1991, pp. 449-457) is used with default seeds to generate random numbers. The binary programming formulation in Section 5.2 is modeled in C++ programming language on Microsoft Visual Studio platform, and IBM ILOG Concert Technology is used for solving the model with IBM ILOG CPLEX 12.6. As explained in the Section 5.2, the model is resolved with added cuts at each iteration. All test problems are optimally solved on a Core i7, 2.60 GHz, 8 GB RAM computer. Computational results for each problem combination are reported in Table 5.3.

Table 5.3 Computational Results for the CSP.

c	*t*	*r*	*p*	*L*	Runtime (sec.)			# of Iterations			Total # of Cuts		
					Min.	**Avg.**	**Max.**	**Min.**	**Avg.**	**Max.**	**Min.**	**Avg.**	**Max.**
1	1	1	1	40	0.703	0.920	1.359	0	0	0	0	0	0
				80	4.250	8.385	13.125	0	0	1	0	23	23
				120	36.984	97.317	311.328	0	0	1	0	29	30
1	1	1	2	40	0.531	0.667	0.875	0	0	0	0	0	0
				80	4.140	6.553	11.704	0	0	1	0	24	24
				120	21.641	80.499	125.172	0	1	2	0	58	140
1	1	2	1	40	0.563	0.733	0.937	0	0	0	0	0	0
				80	3.500	7.150	11.531	0	0	0	0	0	0
				120	28.359	61.384	129.906	0	0	1	0	34	34
1	1	2	2	40	0.531	0.775	0.907	0	0	0	0	0	0
				80	2.219	5.289	7.109	0	0	0	0	0	0
				120	30.609	76.933	197.531	0	0	1	0	52	68
1	2	1	1	40	0.625	0.764	1.062	0	0	0	0	0	0
				80	3.500	9.047	22.577	0	1	3	0	36	75
				120	32.875	147.949	219.609	0	1	2	0	73	224
1	2	1	2	40	0.516	0.697	0.938	0	0	0	0	0	0
				80	2.797	7.303	15.313	0	0	1	0	26	27
				120	50.718	208.394	784.735	0	1	3	0	64	150
1	2	2	1	40	0.547	0.738	1.141	0	0	1	0	18	18
				80	5.328	8.045	14.391	0	0	1	0	23	23
				120	22.344	81.531	306.610	0	0	2	0	49	78
1	2	2	2	40	0.468	0.717	0.891	0	0	0	0	0	0
				80	3.453	7.955	12.781	0	0	1	0	38	56
				120	30.672	92.002	263.266	0	0	2	0	75	105
2	1	1	1	40	0.609	0.864	1.110	0	0	0	0	0	0
				80	1.968	7.811	19.656	0	0	0	0	0	0
				120	41.875	207.610	1053.160	0	1	2	0	55	87
2	1	1	2	40	0.672	0.858	1.344	0	0	0	0	0	0
				80	3.890	7.605	13.985	0	0	0	0	0	0
				120	31.969	74.478	134.312	0	1	2	0	63	120
2	1	2	1	40	0.469	0.671	0.906	0	0	0	0	0	0
				80	3.359	5.798	10.953	0	0	0	0	0	0
				120	26.891	54.044	83.438	0	0	1	0	34	34
2	1	2	2	40	0.687	0.938	1.188	0	0	0	0	0	0
				80	4.719	8.289	13.000	0	0	1	0	27	27
				120	31.953	80.352	242.953	0	0	0	0	0	0
2	2	1	1	40	0.625	0.967	1.375	0	0	0	0	0	0
				80	3.344	9.027	16.828	0	0	1	0	26	26
				120	34.688	196.736	568.015	0	1	3	0	67	180
2	2	1	2	40	0.703	0.985	1.579	0	0	0	0	0	0
				80	5.453	11.198	21.359	0	0	1	0	32	52
				120	36.454	156.816	364.813	0	1	2	0	100	165
2	2	2	1	40	0.515	0.786	1.672	0	0	1	0	18	18
				80	3.375	6.469	11.422	0	0	1	0	68	108
				120	31.016	69.071	140.641	0	0	1	0	34	35
2	2	2	2	40	0.500	0.594	0.671	0	0	0	0	0	0
				80	4.407	6.804	9.906	0	0	1	0	30	30
				120	21.703	91.436	404.375	0	0	1	0	54	72

The table represents the minimum, average and maximum solution times over 10 instances from each setting, along with the number of iterations and the total number of added cuts. Driver set sizes for each crew class are generated to be nonbinding. As shown in the table, the proposed solution procedure is quite efficient in terms of solution time for instances with up to 120 tasks. For almost all instances with 40 tasks, the optimal solutions are obtained with no repetition of solutions, in less than one second. For instances with 80 tasks, the optimal solutions are obtained with at most one iteration, except for setting (1,2,1,1), and the maximum solution time is around 23 seconds. For instances with 120 tasks, the solutions are obtained with at most two iterations, except for settings (1,2,1,2) and (2,2,1,1), and the maximum solution time is 18 minutes.

As shown in the table, the average solution times for settings (1,2,1,1), (1,2,1,2), (2,2,1,1), (2,2,1,2) and (2,1,1,1) are higher than others. In other words, if the number of d-trip tasks is increased (t = 2) the solution time increases accordingly. Additionally, it can be said that the settings with ready times considering peak hours (r = 2) are relatively easier to solve than the settings without peak hours (r = 1). When the processing times of the d-trips are concerned, there is no significant change in solution performance due to changes in the distribution.

6 THE VEHICLE AND CREW SCHEDULING PROBLEM

The Vehicle and Crew Scheduling Problem (VCSP) is considered as a combination of the VSP and CSP in this chapter. An integrated approach is presented first, followed by a sequential one.

6.1 An Integrated Approach for the VCSP

In this section, an integrated approach is proposed for the VCSP, where the vehicles and the drivers are assigned to trips simultaneously. In the integrated VCSP, the aim is to cover all d-trips with an eligible driver and an eligible vehicle, where the ready times and deadlines of the d-trips are fixed in advance, and the objective is to minimize the total fixed cost of drivers, the total fixed cost of vehicles and the total variable costs resulting from deadheads and d-trips. Each d-trip must be assigned to an eligible vehicle and an eligible driver without any overlaps. Additionally, the d-trips belonging to the same trip must be assigned to the same vehicle. Each deadhead arising from the vehicle-d-trip assignments must be covered by an eligible driver. The processing times of the tasks (deadheads and d-trips) assigned to the crew are included in their working time, as well as the sequence-dependent setup times between tasks. Problem notation and the assumptions are stated in Section 6.1.1.

6.1.1 Assumptions and Notation

The notation for the problem formulation is given in Table 6.1. To provide a better understanding of travel times, an illustrative example is given in Figure 6.1. The driving (deadhead) times between d-trips are indicated with parameter a_{ij}, while the driver movement (without driving a vehicle) times between d-trips are specified with parameter b_{ij}. Additionally, as shown in the figure, parameter γ_{ij} is used to indicate the driver movement time between a deadhead and d-trip pair, parameter β_{ij} is used to specify the driver movement time between a deadhead and deadhead pair, while parameter α_{ij} is used to indicate the driver movement time between a d-trip and deadhead pair.

Table 6.1 VCSP Notation.

Sets	
K	Set of trips
D	Set of crew/vehicle/d-trip classes
N	Set of d-trips
Q	Set of crew members (drivers)
V	Set of vehicles (buses)
N_d	Set of d-trips belonging to class $d \in D$; $N_d \subset N$
Q_d	Set of drivers belonging to class $d \in D$; $Q_d \subset Q$
V_d	Set of vehicles belonging to class $d \in D$; $V_d \subset V$
DT_l	Set of d-trips belonging to trip $l \in K$; $DT_l \subset N$
IV_i	Set of incompatible d-trips for d-trip $i \in N$; $IV_i \subset N$
VC_i	Set of eligible vehicles that can perform d-trip $i \in N$; $VC_i \subset V$
QC_i	Set of eligible crew members that can perform d-trip $i \in N$; $QC_i \subset Q$
A_d	Set of d-trips that can be performed by a crew member belonging to class $d \in D$; $N_d \subseteq A_d$
AV_d	Set of d-trips that can be performed by a vehicle belonging to class $d \in D$; $N_d \subseteq AV_d$
ID_d	Set of ineligible crew classes for vehicle class (type) $d \in D$; $ID_d \subseteq D$
Parameters	
r_i	Ready time of d-trip $i \in N$
e_i	Deadline of d-trip $i \in N$
W	Working time limit
S	Spread time limit
c_k	Fixed cost of driver $k \in Q$
t_v	Fixed cost of vehicle $v \in V$
o_v	Trip based fuel consumption cost of vehicle v, per time unit
s_v	Deadhead based fuel consumption cost of vehicle v, per time unit
a_{ij}	Driving time between end location of d-trip i and start location of d-trip j, $i, j \in N$
b_{ij}	Driver movement time between end location of d-trip i and start location of d-trip j, $i, j \in N$
γ_{ij}	Driver movement time between start location of d-trip i and start location of d-trip j, $i, j \in N$
β_{ij}	Driver movement time between start location of d-trip i and end location of d-trip j, $i, j \in N$
α_{ij}	Driver movement time between end location of d-trip i and end location of d-trip j, $i, j \in N$
dc_{ij}	1 if the driving time between end location of d-trip i and start location of d-trip j ($i, j \in N$) is zero, 0 otherwise
Decision Variables	
x_i^{kv}	1 if eligible d-trip i is assigned to driver k and vehicle v, 0 otherwise
y_k	1 if driver k is used, 0 otherwise
z_v	1 if vehicle v is used, 0 otherwise
w_i^v	1 if eligible d-trip i is first d-trip of vehicle v, 0 otherwise
p_i^v	1 if eligible d-trip i is last d-trip of vehicle v, 0 otherwise
h_{ij}^{kv}	1 if eligible d-trip j is covered consecutively after d-trip i by vehicle v and it is assigned to driver k, 0 otherwise
gd_i^k	1 if eligible d-trip i is first task of driver k, 0 otherwise
gt_i^k	1 if eligible d-trip i is last task of driver k, 0 otherwise
g_{ij}^k	1 if eligible deadhead i-j is first task of driver k, 0 otherwise
gc_{ij}^k	1 if eligible deadhead i-j is last task of driver k, 0 otherwise
sb_{ijm}^k	1 if eligible d-trip m is covered consecutively after deadhead i-j by driver k, 0 otherwise
sc_{mij}^k	1 if eligible deadhead i-j is covered consecutively after d-trip m by driver k, 0 otherwise
u_{ijmn}^k	1 if eligible deadhead m-n is covered consecutively after deadhead i-j by driver k, 0 otherwise
uc_{ij}^k	1 if eligible d-trip j is covered consecutively after d-trip i by driver k, 0 otherwise
ar_{ij}^v	1 if eligible d-trip j is covered consecutively after d-trip i by vehicle v with zero deadhead time, 0 otherwise

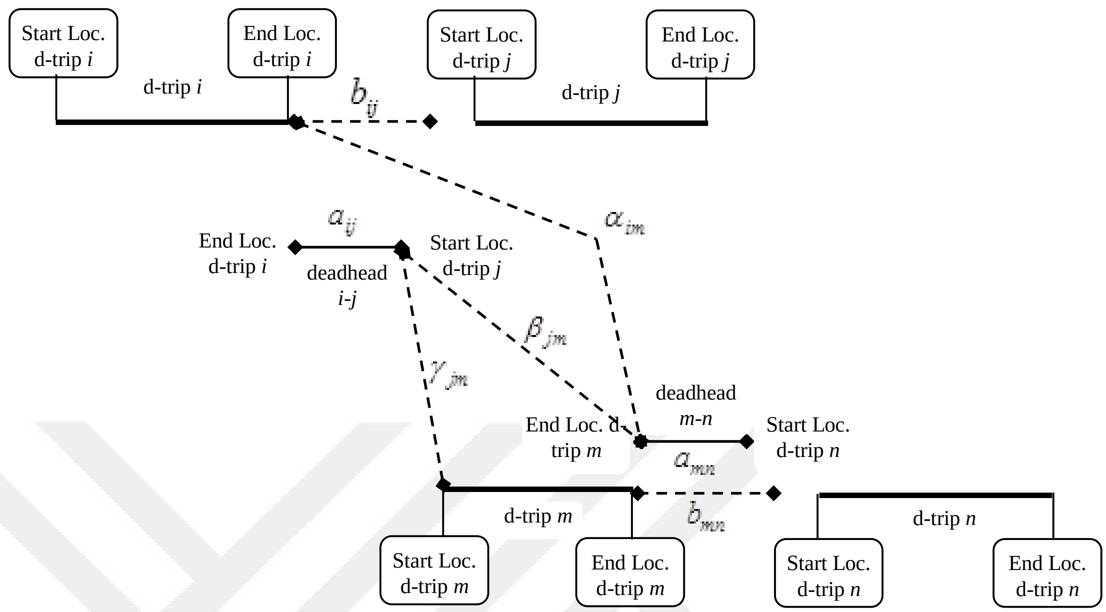

Figure 6.1 Travel Times between Tasks in the VCSP.

The complete assumptions of the VCSP are listed below.

- The problem is solved daily. The timetabling and crew rostering problems are assumed to be solved in advance; hence, the ready times and deadlines of the trips are fixed and known.
- Relief points are known in advance as well as the trips that allow driver change. Therefore, d-trips are given by specified time intervals along with start and end termini. The deadlines of d-trips are defined as their ready times plus their constant processing times.
- A deadhead between d-trips *i* and *j* start immediately after d-trip *i*. Therefore, the ready time of deadhead *i-j* is taken as the deadline of d-trip *i*.
- Preemption is not allowed. A d-trip must be performed wholly by a single crew member and a single vehicle, and d-trips belonging to the same trip must be performed by the same vehicle.
- Delay and cancellation is not possible, all d-trips must be covered.
- Only full time drivers are included.
- Each driver (and vehicle) has a daily fixed cost. When a driver is assigned to a task (a vehicle assigned to a d-trip), the fixed cost is incurred and the privilege to use the driver (vehicle) for a day is obtained. Fixed costs are different for each crew (vehicle) class.
- Spread and working time limits are constant, known and the same for all drivers.
- Sequence-dependent setup (travel) times between tasks are deterministic. There is no setup time before first task and after the last task in a crew schedule.

Similarly, there is no setup time before first d-trip and after the last d-trip in a vehicle schedule.

- Each vehicle has d-trip and deadhead fuel consumption costs. Both deadhead and d-trip fuel consumption costs are defined as different for each vehicle class.
- There are sufficient numbers of drivers and vehicles in each class.
- There are several d-trip classes that require certain vehicle types and several crew classes based on the competencies required to use certain vehicle types. There is a predefined subset of eligible d-trips for each vehicle class. These sets are inclusive for any two vehicle classes d and $f \in D$, either $AV_d \subseteq AV_f$ or $AV_f \subseteq AV_d$. Similarly, there is a predefined subset of eligible d-trips for each crew class. These sets are also inclusive, for any two crew classes d and $f \in D$, either $A_d \subseteq A_f$ or $A_f \subseteq A_d$.
- Each crew can cover tasks performed by certain subset of vehicle classes (types). A task which is assigned to vehicle belonging to class $d \in D$ cannot be performed by a driver belonging to class $f \in ID_d$.

The d-trips are numbered in increasing order based on their ready times, meaning that $r_i \leq r_j$ for all d-trips $i < j$. Sets of drivers belonging to class d (Q_d) are disjoint subsets of driver set Q and sets of vehicles belonging to class d (V_d) are disjoint subsets of vehicle set V. It is assumed that a sufficient number of drivers and vehicles exist in each crew/vehicle class. Hence, the upper bounds on the number of drivers and vehicles belonging to class d, are determined as nonbinding values by using a simple greedy heuristic approach, where the d-trips are grouped based on their d-trip classes and the required number of drivers and vehicles to cover all drips in each d-trip class is determined in a greedy manner with extra allowances.

Sets of d-trips belonging to class d (N_d) are disjoint subsets of the d-trip set N. However, the set of d-trips that can be performed by a vehicle belonging to class d (AV_d) is an inclusive set, meaning that a vehicle can perform d-trips of another class beside its own class. For instance, a solo bus cannot be assigned to d-trips requiring articulated buses. However, an articulated bus can be assigned to both d-trip classes. Also, the set of d-trips that can be performed by a crew belonging to class d (A_d) is an inclusive set, where a crew member can perform d-trips of another class beside its own class. Therefore, decision variables are defined only for eligible crew-vehicle-d-trip assignments.

As mentioned before, each driver is eligible to drive certain subset of vehicle types, correspondingly eligible to perform tasks requiring these certain vehicle types.

However, a task (deadhead or d-trip) can be assigned to a driver only if this driver is eligible to drive the vehicle that is assigned to this task. For instance, if a solo bus is assigned to a d-trip requiring midi bus, a driver only licensed to drive midi buses will not be capable for this d-trip anymore, even if it belongs to set of drivers that can perform d-trips requiring midi buses, since he/she is not eligible to drive solo buses.

An incompatibility set for each d-trip is defined in order to handle the overlaps. The set of d-trips that cannot be performed by the same vehicle are determined through these sets. There may be two reasons for any d-trip pair (*i*,*j*) not to be assigned to the same vehicle. The first reason is that the time windows of the two d-trips overlap ($r_i \leq r_j < e_i$). The second reason could be that a vehicle cannot perform the two d-trips consecutively due to the driving time in between ($r_i \leq r_j < e_i + a_{ij}$). According to these observations, the set of incompatible d-trips for d-trip *i*, namely IV_i, is defined as follows:

$$IV_i = \left\{ j \in N, i < j; r_j < e_i + a_{ij} \right\}.$$

6.1.2 Formulation

Based on the above assumptions and definitions, the binary programming model of the VCSP is given below:

$$Minimize \sum_{v \in V} t_v z_v + \sum_{i \in N} \sum_{v \in VC_i} o_v \Big(\sum_{k \in QC_i} (\mathrm{e}_i - r_i)\, \mathrm{x}_i^{kv} \Big) + \sum_{i \in N} \sum_{j \in N | j > i} \sum_{v \in (VC_i \cap VC_j)} s_v \Big(\sum_{k \in (QC_i \cap QC_j)} a_{ij} h_{ij}^{kv} \Big) + \sum_{k \in Q} c_k y_k \qquad (6.0)$$

$$\text{s.t.} \quad \sum_{k \in QC_i} x_i^{kv} \leq z_v \qquad \forall i \in N, v \in VC_i \qquad (6.1)$$

$$\sum_{v \in VC_i} \sum_{k \in QC_i} x_i^{kv} = 1 \qquad \forall i \in N \qquad (6.2)$$

$$\sum_{k \in QC_i} x_i^{kv} + \sum_{k \in QC_j} x_j^{kv} \leq 1 \qquad \forall i \in N, j \in IV_i, v \in (VC_i \cap VC_j) \qquad (6.3)$$

$$\sum_{k \in QC_i} x_i^{kv} = \sum_{k \in QC_j} x_j^{kv} \qquad \forall i \in DT_l, j \in DT_l \mid j \neq i, l \in K, v \in (VC_i \cap VC_j) \qquad (6.4)$$

$$\sum_{v\in VC_j} w_j^v + \sum_{i\in N|i<j} \sum_{v\in(VC_i\cap VC_j)} (((1-dc_{ij}) \sum_{k\in(QC_i\cap QC_j)} h_{ij}^{kv}) + dc_{ij} ar_{ij}^v) = 1 \qquad \forall j\in N \qquad (6.5)$$

$$\sum_{v\in VC_i} p_i^v + \sum_{j\in N|j>i} \sum_{v\in(VC_i\cap VC_j)} (((1-dc_{ij}) \sum_{k\in(QC_i\cap QC_j)} h_{ij}^{kv}) + dc_{ij} ar_{ij}^v) = 1 \qquad \forall i\in N \qquad (6.6)$$

$$\sum_{i\in AV_d|i<j} \sum_{k\in(QC_i\cap QC_j)} h_{ij}^{kv} \le \sum_{k\in QC_j} x_j^{kv} \qquad \forall j\in AV_d, v\in V_d, d\in D \qquad (6.7)$$

$$\sum_{j\in AV_d|j>i} \sum_{k\in(QC_i\cap QC_j)} h_{ij}^{kv} \le \sum_{k\in QC_i} x_i^{kv} \qquad \forall i\in AV_d, v\in V_d, d\in D \qquad (6.8)$$

$$\sum_{i\in AV_d|i<j} ar_{ij}^v \le \sum_{k\in QC_j} x_j^{kv} \qquad \forall j\in AV_d, v\in V_d, d\in D \qquad (6.9)$$

$$\sum_{j\in AV_d|j>i} ar_{ij}^v \le \sum_{k\in QC_i} x_i^{kv} \qquad \forall i\in AV_d, v\in V_d, d\in D \qquad (6.10)$$

$$\sum_{j\in N|j>i} \sum_{v\in(VC_i\cap VC_j)} (\sum_{k\in(QC_i\cap QC_j)} h_{ij}^{kv} + ar_{ij}^v) \le 1 \qquad \forall i\in N \qquad (6.11)$$

$$\sum_{i\in AV_d} w_i^v = z_v \qquad \forall v\in V_d, d\in D \qquad (6.12)$$

$$\sum_{i\in AV_d} p_i^v = z_v \qquad \forall v\in V_d, d\in D \qquad (6.13)$$

$$w_i^v \le \sum_{k\in QC_i} x_i^{kv} \qquad \forall i\in AV_d, v\in V_d, d\in D \qquad (6.14)$$

$$p_i^v \le \sum_{k\in QC_i} x_i^{kv} \qquad \forall i\in AV_d, v\in V_d, d\in D \qquad (6.15)$$

$$\sum_{k\in QC_n} gd_n^k + \sum_{i\in N|i<n} \sum_{j\in N|j>i} \sum_{k\in(QC_i\cap QC_j\cap QC_n)} sb_{ijn}^k + \sum_{m\in N|m<n} \sum_{k\in(QC_n\cap QC_m)} uc_{mn}^k = 1 \qquad \forall n\in N \qquad (6.16)$$

$$\sum_{k\in QC_i} gt_i^k + \sum_{n\in N|n>i} \sum_{m\in N|m<n} \sum_{k\in(QC_i\cap QC_m\cap QC_n)} sc_{imn}^k + \sum_{j\in N|j>i} \sum_{k\in(QC_i\cap QC_j)} uc_{ij}^k = 1 \qquad \forall i\in N \qquad (6.17)$$

$$\sum_{k\in(QC_n\cap QC_m)} g_{mn}^{k} + \sum_{i\in N|i<n,i\neq m}\ \sum_{j\in N|j>i}\ \sum_{k\in(QC_i\cap QC_j\cap QC_m\cap QC_n)} u_{ijmn}^{k} + \sum_{i\in N|i<n}\ \sum_{k\in(QC_n\cap QC_m\cap QC_i)} sc_{imn}^{k} = \sum_{v\in(VC_m\cap VC_n)}\ \sum_{k\in(QC_n\cap QC_m)} h_{mn}^{kv} \quad \forall m\in N, n\in N\mid n>m \tag{6.18}$$

$$\sum_{k\in(QC_i\cap QC_j)} gc_{ij}^{k} + \sum_{n\in N|n>i}\ \sum_{m\in N|m<n,m\neq i}\ \sum_{k\in(QC_i\cap QC_j\cap QC_m\cap QC_n)} u_{ijmn}^{k} + \sum_{n\in N|n>i}\ \sum_{k\in(QC_i\cap QC_j\cap QC_n)} sb_{ijn}^{k} = \sum_{v\in(VC_i\cap VC_j)}\ \sum_{k\in(QC_i\cap QC_j)} h_{ij}^{kv} \quad \forall i\in N, j\in N\mid j>i \tag{6.19}$$

$$\sum_{i\in A_d} gd_i^k + \sum_{m\in A_d}\ \sum_{n\in A_d|n>m} g_{mn}^{k} = y_k \quad \forall k\in Q_d, d\in D \tag{6.20}$$

$$\sum_{j\in A_d} gt_j^k + \sum_{m\in A_d}\ \sum_{n\in A_d|n>m} gc_{mn}^{k} = y_k \quad \forall k\in Q_d, d\in D \tag{6.21}$$

$$\sum_{v\in VC_i} x_i^{kv} \le y_k \quad \forall i\in N, k\in QC_i \tag{6.22}$$

$$\sum_{i\in A_d|i<j}\ \sum_{v\in(VC_i\cap VC_j)} h_{ij}^{kv} \le y_k \quad \forall j\in A_d, k\in Q_d, d\in D \tag{6.23}$$

$$gd_i^k \le \sum_{v\in VC_i} x_i^{kv} \quad \forall i\in N, k\in QC_i \tag{6.24}$$

$$gt_i^k \le \sum_{v\in VC_i} x_i^{kv} \quad \forall i\in N, k\in QC_i \tag{6.25}$$

$$g_{ij}^k \le \sum_{v\in(VC_i\cap VC_j)} h_{ij}^{kv} \quad \forall i\in N, j\in N\mid j>i, k\in(QC_i\cap QC_j) \tag{6.26}$$

$$gc_{ij}^k \le \sum_{v\in(VC_i\cap VC_j)} h_{ij}^{kv} \quad \forall i\in N, j\in N\mid j>i, k\in(QC_i\cap QC_j) \tag{6.27}$$

$$\sum_{m\in A_d|m>i} sb_{ijm}^k \le \sum_{v\in(VC_i\cap VC_j)} h_{ij}^{kv} \quad \forall i\in A_d, j\in A_d\mid j>i, k\in Q_d, d\in D \tag{6.28}$$

$$\sum_{i\in A_d|i<m}\ \sum_{j\in A_d|j>i} sb_{ijm}^k \le \sum_{v\in VC_m} x_m^{kv} \quad \forall m\in A_d, k\in Q_d, d\in D \tag{6.29}$$

$$\sum_{j\in A_d|j>m}\ \sum_{i\in A_d|i<j} sc_{mij}^k \le \sum_{v\in VC_m} x_m^{kv} \quad \forall m\in A_d, k\in Q_d, d\in D \tag{6.30}$$

$$\sum_{m \in A_d | m < j} sc_{mij}^{k} \le \sum_{v \in (VC_i \cap VC_j)} h_{ij}^{kv} \qquad \forall i \in A_d, j \in A_d \mid j > i, k \in Q_d, d \in D \qquad (6.31)$$

$$\sum_{n \in A_d | n > i} \sum_{m \in A_d | m < n, m \ne i} u_{ijmn}^{k} \le \sum_{v \in (VC_i \cap VC_j)} h_{ij}^{kv} \quad \forall i \in A_d, j \in A_d \mid j > i, k \in Q_d, d \in D \qquad (6.32)$$

$$\sum_{i \in A_d | i < n, i \ne m} \sum_{j \in A_d | j > i} u_{ijmn}^{k} \le \sum_{v \in (VC_m \cap VC_n)} h_{mn}^{kv} \quad \forall m \in A_d, n \in A_d \mid n > m, k \in Q_d, d \in D \qquad (6.33)$$

$$\sum_{j \in A_d | j > i} uc_{ij}^{k} \le \sum_{v \in VC_i} x_i^{kv} \qquad \forall i \in A_d, k \in Q_d, d \in D \qquad (6.34)$$

$$\sum_{i \in A_d | i < j} uc_{ij}^{k} \le \sum_{v \in VC_j} x_j^{kv} \qquad \forall j \in A_d, k \in Q_d, d \in D \qquad (6.35)$$

$$uc_{ij}^{k} + \sum_{v \in (VC_i \cap VC_j)} h_{ij}^{kv} \le 1 \qquad \forall i \in A_d, j \in A_d \mid j > i, k \in Q_d, d \in D \qquad (6.36)$$

$$e_i + a_{ij} \le r_j + (1 - \sum_{v \in (VC_i \cap VC_j)} h_{ij}^{kv})M \qquad \forall i \in N, j \in N \mid j > i, k \in (QC_i \cap QC_j) \qquad (6.37)$$

$$e_i + b_{ij} \le r_j + (1 - uc_{ij}^{k})M \qquad \forall i \in N, j \in N \mid j > i, k \in (QC_i \cap QC_j) \qquad (6.38)$$

$$e_i + a_{ij} + \gamma_{jm} \le r_m + (1 - sb_{ijm}^{k})M \qquad (6.39)$$
$$\forall i \in N, j \in N \mid j > i, m \in N \mid m > i, k \in (QC_i \cap QC_j \cap QC_m)$$

$$e_i + a_{ij} + \beta_{jm} \le e_m + (1 - u_{ijmn}^{k})M \qquad (6.40)$$
$$\forall m \in N, n \in N \mid n > m, i \in N \mid i < n, i \ne m, j \in N \mid j > i, k \in (QC_i \cap QC_j \cap QC_m \cap QC_n)$$

$$e_m + \alpha_{mi} \le e_i + (1 - sc_{mij}^{k})M \qquad (6.41)$$
$$\forall m \in N, j \in N \mid j > m, i \in N \mid i < j, k \in (QC_i \cap QC_j \cap QC_m)$$

$$e_j(\sum_{v \in VC_j} x_j^{kv}) - r_i(\sum_{v \in VC_i} x_i^{kv}) \le S + (1 - \sum_{v \in VC_i} x_i^{kv})M \qquad (6.42)$$
$$\forall i \in N, j \in N \mid j > i, k \in (QC_i \cap QC_j)$$

$$e_m(\sum_{v \in VC_m} x_m^{kv}) - e_i(\sum_{v \in (VC_i \cap VC_j)} h_{ij}^{kv}) \le S + (1 - \sum_{v \in (VC_i \cap VC_j)} h_{ij}^{kv})M \qquad (6.43)$$
$$\forall m \in N, i \in N \mid i < m, j \in N \mid j > i, k \in (QC_i \cap QC_j \cap QC_m)$$

$$(e_m + a_{mn})(\sum_{v\in(VC_m \cap VC_n)} h_{mn}^{kv}) - e_i(\sum_{v\in(VC_i \cap VC_j)} h_{ij}^{kv}) \le S + (1 - \sum_{v\in(VC_i \cap VC_j)} h_{ij}^{kv})M \tag{6.44}$$

$$\forall i \in N, j \in N \mid j > i, n \in N \mid n > i, m \in N \mid m < n, k \in (QC_i \cap QC_j \cap QC_m \cap QC_n)$$

$$(e_i + a_{ij})(\sum_{v\in(VC_i \cap VC_j)} h_{ij}^{kv}) - r_m(\sum_{v\in VC_m} x_m^{kv}) \le S + (1 - \sum_{v\in VC_m} x_m^{kv})M \tag{6.45}$$

$$\forall i \in N, j \in N \mid j > i, m \in N \mid m < j, k \in (QC_i \cap QC_j \cap QC_m)$$

$$\sum_{i\in A_d}(e_i - r_i)(\sum_{v\in VC_i} x_i^{kv}) + \sum_{i\in A_d}\sum_{j\in A_d \mid j>i}(a_{ij}(\sum_{v\in(VC_i\cap VC_j)} h_{ij}^{kv}) + b_{ij}uc_{ij}^k + (\sum_{m\in A_d \mid m>i}\gamma_{jm} sb_{ijm}^k) +$$
$$(\sum_{n\in A_d \mid n>i}\sum_{m\in A_d \mid m<n, m\neq i}\beta_{jm}u_{ijmn}^k) + (\sum_{m\in A_d \mid m<j}\alpha_{mi} sc_{mij}^k)) \le W y_k \tag{6.46}$$
$$\forall k \in Q_d, d \in D$$

$$\sum_{d\in D}\sum_{v\in V_d}\sum_{i\in AV_d}\sum_{f\in ID_d}\sum_{k\in(Q_f\cap QC_i)} x_i^{kv} = 0 \tag{6.47}$$

$$\sum_{d\in D}\sum_{v\in V_d}\sum_{i\in AV_d}\sum_{j\in AV_d \mid j>i}\sum_{f\in ID_d}\sum_{k\in(QC_i\cap QC_j\cap Q_f)} h_{ij}^{kv} = 0 \tag{6.48}$$

$$\begin{aligned}
& x_i^{kv} \in [0,1], \forall i \in N, v \in VC_i, k \in QC_i, y_k \in [0,1], \forall k \in Q, z_v \in [0,1], \forall v \in V \\
& w_i^v, p_i^v \in [0,1], \forall i \in N, v \in VC_i \\
& h_{ij}^{kv} \in [0,1], \forall i \in N, j \in N \mid j > i, v \in (VC_i \cap VC_j), k \in (QC_i \cap QC_j) \\
& ar_{ij}^v \in [0,1], \forall i \in N, j \in N \mid j > i, v \in (VC_i \cap VC_j) \\
& gd_i^k, gt_i^k \in [0,1], \forall i \in N, k \in QC_i \\
& g_{ij}^k, gc_{ij}^k, uc_{ij}^k \in [0,1], \forall i \in N, j \in N \mid j > i, k \in (QC_i \cap QC_j) \\
& sb_{ijm}^k \in [0,1], \forall i \in N, j \in N \mid j > i, m \in N \mid m > i, k \in (QC_i \cap QC_j \cap QC_m) \\
& sc_{mij}^k \in [0,1], \forall m \in N, j \in N \mid j > m, i \in N \mid i < j, k \in (QC_m \cap QC_i \cap QC_j) \\
& u_{ijmn}^k \in [0,1], \\
& \forall m \in N, n \in N \mid n > m, i \in N \mid i < n, i \neq m, j \in N \mid j > i, k \in (QC_m \cap QC_n \cap QC_i \cap QC_j)
\end{aligned} \tag{6.49}$$

The objective function minimizes the total fixed cost of drivers, the total fixed cost of vehicles and total variable costs arising from deadheads and d-trips. Constraint set (6.1) ensures that a d-trip can be assigned to an eligible vehicle only if it is used. Constraint set (6.2) guarantees that each d-trip must be covered by an eligible vehicle and an eligible driver. Constraint set (6.3) says that incompatible d-trips cannot be assigned to the same vehicle. Constraint set (6.4) ensures that the d-trips belonging to the same trip must be assigned to the same vehicle. Constraint sets (6.5) and (6.6) guarantee that all d-trips must be scheduled to an eligible vehicle, that is, there must be an arrival to and a departure from each d-trip. These constraints also ensure that an eligible driver can be assigned to a deadhead only if the deadhead duration is greater than zero, otherwise the deadhead is only assigned to an eligible vehicle.

Constraint sets (6.7), (6.8), (6.9) and (6.10) ensure that there can be a deadhead between any two d-trips only if both of these d-trips are assigned to same vehicle. Particularly, constraint sets (6.7) and (6.8) consider the deadheads which have durations greater than zero, while constraint sets (6.9) and (6.10) consider the deadheads that have zero duration. Constraint set (6.11) guarantees that two types of deadhead variables cannot be equal to 1 for any d-trip pair at the same time. Constraint sets (6.12) and (6.13) guarantee that there must be a first and a last d-trip for each used vehicle. Constraint sets (6.14) and (6.15) ensure that a d-trip can be a first/last d-trip of an eligible vehicle only if it is assigned to that vehicle.

Constraint sets (6.16) and (6.17) guarantee that all d-trips must be scheduled to an eligible driver, that is, there must be an arrival to and a departure from each d-trip. Similarly, constraint sets (6.18) and (6.19) guarantee that all deadheads must be scheduled to an eligible driver, that is, there must be an arrival to and a departure from each deadhead. The constraint sets (6.16), (6.17), (6.18) and (6.19) consider the all consecutive task pair alternatives; d-trip-d-trip, d-trip-deadhead, deadhead-d-trip and deadhead-deadhead task pairs. Constraint sets (6.20) and (6.21) ensure that there must be a first and a last task (deadhead or d-trip) for each used driver. Constraint set (6.22) ensures that a d-trip can be assigned to an eligible driver only if it is used. Similarly, constraint set (6.23) guarantees that a deadhead can be assigned to an eligible driver only if it is used. Constraint sets (6.24) and (6.25) ensure that a d-trip can be a first/last d-trip of an eligible driver only if it is assigned to that driver. Similarly, constraint sets (6.26) and (6.27) guarantee that a deadhead can be a first/last deadhead of an eligible driver only if it is assigned to that driver.

Constraint sets (6.28) and (6.29) ensure that a d-trip can be consecutively covered after a deadhead only if both of them are assigned to same driver. Similarly, constraint sets (6.30) and (6.31) guarantee that a deadhead can be consecutively covered after a d-trip only if both of them are assigned to same driver. Constraint sets (6.32) and (6.33) ensure that there can be a driver movement between any two deadheads only if both of these deadheads are assigned to same driver. In same way, constraint sets (6.34) and (6.35) guarantee that there can be a driver movement between any two d-trips only if both of these d-trips are assigned to same driver, where the driver movement refers to travel activity without driving a vehicle. Constraint set (6.36) ensures that a driver can cover any two d-trips consecutively, either by driving vehicle or without driving a vehicle.

Constraint set (6.37) ensures that there cannot be a deadhead between any two d-trips if the time windows of these d-trips overlap or the driver cannot perform these d-trips consecutively due to the driving time in between. Similarly, constraint set (6.38) guarantees that there cannot be a driver movement between any two d-trips if the time windows of these d-trips overlap or the driver cannot perform these d-trips consecutively due to the driver movement time in between. Constraint set (6.39) ensures that a d-trip can be consecutively covered after a deadhead only if the time windows of these tasks do not overlap regarding the driver movement time in between. Similarly, constraint set (6.40) guarantees that a deadhead can be consecutively covered after a deadhead only if the time windows of these tasks do not overlap regarding the driver movement time in between. In same way, constraint set (6.41) ensures that a deadhead can be consecutively covered after a d-trip only if the time windows of these tasks do not overlap regarding the driver movement time in between.

Constraint sets (6.42), (6.43), (6.44) and (6.45) ensure that the difference between the deadline of second task and the ready time of first task do not exceed the total spread time limit for all task pairs assigned to a driver. Particularly, constraint set (6.42) considers the all d-trip-d-trip pairs, constraint set (6.43) considers the all deadhead-d-trip pairs, constraint set (6.44) considers the all deadhead-deadhead pairs, and constraint set (6.45) considers the all d-trip-deadhead pairs. Constraint set (6.46) ensures that total working time of a driver cannot exceed the total working time limit. The d-trip durations, the deadhead durations and the driver movement durations between tasks are included in working time. Constraint set (6.47) guarantees that a d-trip can be assigned to a driver only if this driver is eligible to drive the vehicle that is assigned to this d-trip. Similarly, constraint set (6.48) ensures that a deadhead can be assigned to a driver only if this driver is eligible to drive the vehicle that is assigned to this deadhead.

All decision variables are defined as binary by (6.49) considering only eligible vehicle-driver-d-trip assignments. Particularly, the eligible vehicle set for a deadhead between d-trip *i* and d-trip *j* is defined as the intersection set of the eligible vehicle set for d-trip *i* and the eligible vehicle set for d-trip *j*, since the vehicle assigned to this deadhead should be eligible to perform both of these d-trips. Similarly, the eligible driver sets for driver movements and deadheads between tasks are defined in same way, since the driver assigned to a deadhead/driver movement should be eligible to drive both of the vehicles assigned to these tasks.

The proposed binary programming model is optimally solved for small-sized problem instances and the results are presented in Section 6.3. However, larger instances cannot be solved within reasonable time limits due to exponentially increasing solution times. Therefore, a sequential approach is proposed in the following section.

6.2 A Sequential Approach for the VCSP

In the sequential approach proposed in this thesis, the vehicles are firstly assigned to trips and the crew members are assigned to tasks, i.e. the generated vehicle blocks. Namely, the CSP presented in Chapter 5 is solved using the output of the VSP in Chapter 4. The solution procedure including the *iterative valid inequality generation scheme* is used for the CSP in order to cover tasks resulting from the computational results of the VSP in Section 4.3. The detailed description, assumptions, formulation and computational results for both models were presented in Chapters 4 and 5.

6.2.1 Computational Study

In this section, a computational experiment is described to evaluate the performance of the sequential approach for the VCSP. The vehicle blocks generated by the computational study for VSP in Section 4.3 are used as input for the crew scheduling part of the problem. In order to schedule the drivers, tasks are defined based on the vehicle blocks. Based on the obtained vehicle schedules, the trips and deadheads are defined, and the initial trip types are updated according to vehicle assignments. For instance, if a solo bus has been assigned to a trip originally requiring a midi bus, a driver only licensed to drive midi buses will not be eligible for this trip anymore, and this trip now requires a driver licensed to drive solo buses for the solution of the CSP.

As mentioned before, a task refers to deadhead or d-trip where each task must be performed by a single driver. In VSP, the trips must be performed by a single vehicle. However, in CSP, a trip can be divided into d-trips where each d-trip must be performed by a single driver. Hence, d-trips are defined from the solutions of the VSP by dividing 50% of the trips which are longer than equal to 8 time units. The trips are divided into 2 d-trips at a point that follows a discrete uniform distribution between 40% and 60% of the trip duration. Hence, the durations of d-trips are within the range [3, 13] similar to Section 5.3.

It is assumed that a deadhead between d-trips *i* and *j* start immediately after *i*. Therefore, the ready time of deadhead *i-j* is taken as the deadline of *i*. Additionally, the durations of deadheads are within the range [1, 5] similar to Section 5.3, as the deadheads having zero duration are not included. The setup time between any two tasks is determined by the shortest path between these tasks. Furthermore, the setup time between consecutive tasks of a vehicle schedule is assumed to be zero, as tasks performed by the same vehicle can also be performed by the same driver. Based on these assumptions, the setup times are generated uniformly in the range [0,5]. The time limits and costs are taken as in Chapter 5.

Regarding the above assumptions, the solutions of the test problems in Section 4.3 are converted to appropriate input data for the CSP. The percentages of deadheads, the percentages of tasks for each class, and the percentages of tasks having ready times within peak hours are shown in Table 6.2. As shown in the table, approximately 25% of the tasks are deadheads for all settings.

Table 6.2 Crew Scheduling Problem Data Analysis.

VSP Settings			**Deadhead Tasks**		**Class 1 Tasks**		**Class 2 Tasks**		**Class 3 Tasks**		**Peak Hour [30-50]**		**Peak Hour [120-160]**	
c	***r***	***p***	**Avg. (%)**	**Std. Dev.**	**Avg. (%)**	**Std. Dev.**	**Avg. (%)**	**Std. Dev.**	**Avg. (%)**	**Std. Dev.**	**Avg. (%)**	**Std. Dev.**	**Avg. (%)**	**Std. Dev.**
1	1	1	27	0.03	19	0.05	50	0.08	31	0.06	10	0.06	22	0.06
1	1	2	26	0.03	22	0.03	48	0.04	30	0.03	12	0.06	22	0.05
1	2	1	25	0.05	19	0.06	50	0.07	31	0.04	25	0.03	25	0.03
1	2	2	24	0.04	20	0.07	50	0.06	31	0.05	24	0.03	24	0.03
2	1	1	27	0.03	15	0.06	40	0.07	45	0.04	12	0.06	21	0.06
2	1	2	26	0.03	16	0.04	39	0.05	45	0.06	11	0.05	21	0.06
2	2	1	25	0.05	16	0.06	40	0.07	44	0.04	24	0.04	24	0.02
2	2	2	24	0.04	17	0.05	38	0.06	45	0.05	23	0.02	25	0.03

As mentioned in Section 4.3, in the first setting (*c*=1), 25% of the trips belong to class 1 (midi bus), 50% of the trips belong to class 2 (solo bus) and the remaining 25% belong to class 3 (articulated bus). After solving the VSP, approximately 20% of the tasks belong to class 1 (midi bus), 50% of the tasks belong to class 2 (solo bus) and the remaining 30% to belong class 3 (articulated bus). In the second setting (*c*=2), 20% of the trips belong to class 1, 40% of the trips belong to class 2 and the remaining 40% belong to class 3. After solving the VSP, approximately 15% of the tasks belong to class 1, 40% of the tasks belong to class 2 and the remaining 45% to belong class 3. For both settings (*c*=1, *c*=2), the percentage distribution of the tasks generally reflect the percentages of trips in terms of class assignments, except that the tasks belonging to class 1 are decreased by 5% and the tasks belonging to class 3 are increased by 5%. As mentioned in Section 4.3, in the first setting (*r*=1), the ready times follow a discrete uniform distribution in the range [0, 200]. After solving the VSP, approximately 10% of the tasks are in the range [30, 50] while 20% of the tasks are in the range [120, 160]. The percentages of tasks fit the discrete uniform distribution for the range [0, 200]. In the second setting (*r*=2), a peak time is considered where 25% of ready times follow a discrete uniform distribution in the range [30, 50], 25% come from the range [120,160], and the remaining 50% from ranges [0-29], [51,119] and [161,200]. After solving the VSP, approximately 25% of the tasks are in the range [30, 50] while 25% of the tasks are in the range [120, 160], which are similar to the percentages in the VSP. For both settings (*r*=1, *r*=2), it can be concluded that ready times of tasks reflects the ready time settings used for VSP.

6.2.2 Computational Results

As in Section 4.3, random test vehicle scheduling problems were generated for trip sets with *K* = 20, 40, 80 and 120. For each setting and trip set combination, 10 random test problems were generated, summing up to 320 problem instances in total. In this section, the CSP of Chapter 5 is solved based on the results of these problem instances. Computational results for these problem combinations are reported in Table 6.3. The table represents the minimum, average and maximum solution times over 10 instances from each setting, along with the number of tasks, the number of iterations and the total number of added cuts. The proposed solution procedure is quite effective in terms of solution time for instances with up to 84 tasks (*K*=20 and 40). For instances with up to 44 tasks (*K*=20), the optimal solutions are obtained with no repetition of solutions, in less than 1.5 seconds. For instances having tasks between 66 and 84

(*K*=40), the optimal solutions are obtained with at most 2 iterations, except for setting (1,1,2), and the maximum solution time is around 32 seconds. For instances having tasks between 128 and 163 (*K*=80), the solutions are obtained with at most 6 iterations except settings (1,1,2) and (2,2,1), and the maximum solution time is 88 minutes. It can be concluded that the optimal solutions are obtained with at most 9 iterations for all settings.

As shown in the table, the solution times increase exponentially due to the increasing number of tasks. Additionally, it can be said that the settings with ready times considering peak hours (r = 2) are relatively easier to solve than settings without peak hours (r = 1), as the solution times of the settings considering peak hours are lower than the others. This outcome is also consistent with the outcome of the computational study for the CSP in Section 5.3. Furthermore, as shown in the table, the solution times of the settings (c=2) are slightly higher than the settings (c=1).

Table 6.3 Computational Results (sequential) for *K* = 20, 40 and 80.

c	*r*	*p*	*K*	# of Tasks (*L*)			Runtime (Sec.)			# Iterations		# Cuts	
				Min.	Avg.	Max.	Min.	Avg.	Max.	Avg.	Max.	Avg.	Max.
1	1	1	20	33	37	42	0.44	0.76	1.36	0	0	0	0
			40	66	72	79	2.36	5.66	12.08	0	1	17	17
			80	132	141	152	72.59	606.02	2245.41	1	6	149	435
1	1	2	20	37	40	44	0.42	0.63	0.95	0	0	0	0
			40	75	79	84	2.11	9.84	32.50	1	5	80	105
			80	150	155	163	74.88	937.47	2079.48	2	8	140	480
1	2	1	20	35	37	41	0.42	0.56	0.83	0	0	0	0
			40	66	73	80	2.89	5.68	15.44	0	0	0	0
			80	132	140	148	62.69	216.62	694.78	0	2	52	75
1	2	2	20	38	40	43	0.49	0.72	1.20	0	0	0	0
			40	73	77	81	3.33	9.40	15.88	0	0	0	0
			80	144	146	152	241.97	543.62	1187.48	1	1	44	90
2	1	1	20	33	37	41	0.39	0.53	0.72	0	0	0	0
			40	69	74	81	2.25	5.16	11.09	0	1	18	19
			80	139	143	153	147.19	998.80	5302.08	2	9	185	783
2	1	2	20	36	39	42	0.44	0.61	1.03	0	0	0	0
			40	75	80	83	3.16	9.88	26.91	0	1	40	40
			80	147	153	161	136.56	762.01	2201.02	1	3	91	130
2	2	1	20	35	38	40	0.36	0.60	0.75	0	0	0	0
			40	66	73	79	2.33	5.88	12.88	0	2	38	38
			80	128	139	148	68.03	273.48	642.38	1	2	49	81
2	2	2	20	38	40	43	0.47	0.65	1.05	0	0	0	0
			40	72	75	82	3.63	5.86	9.47	0	1	36	36
			80	135	148	156	178.42	744.37	2807.95	1	4	95	300

As mentioned in Section 4.3, two sets of distributions are used for the processing times of trips; (p=1) discrete uniform distribution in the range [4,13] and (p=2) triangular distribution in the range [4,9,13]. It can be concluded that the settings which are based on trips following a triangular distribution (p=2) are relatively harder to solve than the other settings which are based on trips following a discrete uniform distribution (p=1), since the solution times of the settings following a triangular distribution are higher than the others. This result is expected since the number of trips that will be divided into d-trips, and correspondingly the number of tasks are increased when the processing times of trips follow a triangular distribution.

Test problems for K = 120 are solved under a 3-hour run time limit on a Core i7, 2.60 GHz, 8 GB RAM computer. Computational results are reported in Table 6.4. The table represents the minimum, average and maximum solution times over 10 instances from each setting, along with the number of tasks, the number of iterations and the total number of added cuts. Number of optimally solved instances within the time limit (out of 10) are also reported. As mentioned before, driver set sizes for each crew class are generated to be nonbinding for all instances.

Table 6.4 Computational Results (sequential) for *K* = 120.

c	***r***	***p***		**# of Tasks (*L*)**			**Runtime (Sec.)**			**# Iterations**		**# Cuts**	
			Opt.	**Min.**	**Avg.**	**Max.**	**Min.**	**Avg.**	**Max.**	**Avg.**	**Max.**	**Avg.**	**Max.**
1	1	1	6	201	208	216	1902.20	7236.64	10800.00	1	2	128	272
1	1	2	3	210	221	227	7836.41	10214.41	10800.00	1	3	213	408
1	2	1	8	183	198	211	1940.89	5756.61	10800.00	1	2	78	152
1	2	2	4	202	214	220	2881.86	8718.68	10800.00	1	2	133	216
2	1	1	9	201	208	215	1236.89	6457.97	10800.00	2	6	257	576
2	1	2	3	211	222	231	3772.26	9752.29	10800.00	1	4	254	416
2	2	1	8	188	197	205	2742.19	7879.66	10800.00	1	3	148	252
2	2	2	2	210	216	226	2818.73	9460.60	10800.00	0	1	88	152

As shown in Table 6.4, number of tasks changes in between 183 and 231, and some of the instances cannot be optimally solved within 3 hours in all settings. For only settings (1,2,1), (2,1,1) and (2,2,1), almost all instances are optimally solved within the time limit. When the processing times of trips are concerned, the same conclusion can be made for these instances; the settings which are based on trips following a triangular distribution (p=2) are relatively harder to solve than other settings, which are based on trips following a discrete uniform distribution (p=1).

6.3 Comparison

In this section, the sequential approach is compared with the integrated approach for the VCSP. Regarding the complexity of the problem, random test problems are generated only for trip sets with K = 12 and K = 15. The same setting described in Section 4.3 is used for vehicle scheduling. There are two settings for trip-class assignments. In the first setting (c=1), 25% of the trips belong to class 1 (midi bus), 50% of the trips belong to class 2 (solo bus) and the remaining 25% belong to class 3 (articulated bus). In the second setting (c=2), 20% of the trips belong to class 1, 40% of the trips belong to class 2 and the remaining 40% belong to class 3.

Two sets of ready times are generated. In the first set (r=1), the ready times follow a discrete uniform distribution in the range [0,200]. In the second set (r=2), a peak time is considered where 25% of ready times follow a discrete uniform distribution in the range [30, 50], 25% come from the range [120,160], and the remaining 50% from ranges [0-29], [51,119] and [161,200]. Two sets of distributions are used for the processing times of trips; (p=1) discrete uniform distribution in the range [4,13] and (p=2) triangular distribution in the range [4,9,13]. Fixed cost of vehicles in each class are determined as 60, 80 and 120 monetary units, respectively. Trip based fuel consumption cost of vehicles in each class are determined as 3.5, 5 and 7 monetary units per time unit, respectively. Deadhead based fuel consumption cost of vehicles in each class are determined as 4.5, 6.5 and 9 monetary units per time unit, respectively.

The d-trips are defined by dividing 50% of the trips which are longer than equal to 8 time units. The trips are divided into 2 d-trips at the point that follows a discrete uniform distribution between 40% and 60% of the trip duration. Hence, the durations of d-trips are within the range [3, 13]. The deadhead duration (driving time) between end location of d-trip i and start location of d-trip j is determined by the shortest path between them and they are uniform in the range [0,5]. The deadhead duration between d-trips of same trip is assumed to be zero. There are 3 crew classes due to vehicle types; midi bus, solo bus and articulated bus. A driver licensed to drive a midi bus can be assigned to tasks only with midi buses. However, a driver licensed to drive a solo bus can be assigned to two classes; with midi and solo buses. A driver licensed to drive an articulated bus is eligible to drive midi, solo and articulated buses. Time limits for drivers are set as S=100 and W=80. Fixed cost of drivers in each class are determined as 160, 180 and 200 monetary units, respectively.

In the sequential approach, it is assumed that the setup (driver movement) time between any two tasks is determined by the shortest path between these tasks. In order to be able compare the results of integrated approach to the sequential one, this assumption is regarded in all setup times which are used in sequential approach. However, this assumption is not valid for other generated setup times which are not used in sequential approach. Consequently, setup times between d-trips i and j (b_{ij}, γ_{ij}, β_{ij}, α_{ij}) are generated uniformly in the range [0,5]. For each setting and trip set, a random test problem is generated, summing up to 16 problem instances. The binary programming formulations of the sequential approach which are described in Sections 4.2 and 5.2 are modeled in C++ programming language on Microsoft Visual Studio platform, and IBM ILOG Concert Technology is used for solving the models with IBM ILOG CPLEX 12.6. The binary programming formulation of integrated approach which is described in Section 6.1.2 is modeled on IBM ILOG CPLEX Optimization Studio platform and solved with IBM ILOG CPLEX 12.6.

All test problems having 12 trips are optimally solved with integrated approach, on a Core i7, 2.60 GHz, 8 GB RAM computer. The results of the sequential approach are obtained by optimally solving the VSP and CSP sequentially on the same computer. Bus and driver set sizes for each d-trip class are generated to be nonbinding. The integrated and sequential approach results for each problem instance having 12 trips are reported in Table 6.5, as well as the number of d-trips. The sequential approach results are compared with the optimal solutions (integrated approach) in terms of objective function values and run times. The percent gaps from the optimal solution are calculated as:

$$Gap\ \% = \frac{(Obj(\text{sequential approach}) - Obj(\text{optimal}))}{Obj(\text{optimal})} * 100\ ,$$

where *Obj(sequential approach)* is the result obtained by optimally solving the models of sequential approach, and *Obj(optimal)* is the optimal objective function value of the model of integrated approach. As shown in Table 6.5, the optimal solutions are obtained in less than 1 second for 6 of the 8 instances using the sequential approach, while they are obtained in more than 20 minutes with the integrated approach. Note that, the optimal results of the integrated approach are obtained in more than 3.5 hours for 3 instances even for these small problem instances. As shown in the table, 2 instances cannot be optimally solved using the sequential approach, and the maximum gap is 7.03%. However, the solution times of the sequential approach clearly outperform the solution times of the integrated one. Hence, it can be concluded that our sequential approach is quite efficient and effective in finding near optimal solutions within reasonable computation times.

Table 6.5 Comparison of the models for *K* = 12.

c	*r*	*p*	# d-trips (*N*)	Sequential Runtime (sec.)	Integrated Runtime (sec.)	Gap %
1	1	1	17	0.718	21640.390	0.00
1	1	2	18	0.749	16264.160	0.00
1	2	1	15	0.516	4768.000	7.03
1	2	2	16	0.516	13186.380	0.98
2	1	1	16	0.595	4598.060	0.00
2	1	2	17	0.563	4814.090	0.00
2	2	1	16	0.531	5233.030	0.00
2	2	2	16	0.594	1725.270	0.00

For some of the additional generated instances having 12 trips, it has been observed that the optimal (integrated approach) solutions cannot be obtained within 10 hours of computation time. Furthermore, for larger instances having more than 12 trips, it has been observed that the solution times increase exponentially. The results of the sequential and integrated approaches for the random test problems having 15 trips are reported in Table 6.6, as well as the number of dtrips. These test problems are solved on same computer. However, they are solved with integrated approach under a 1-hour run time limit due to exponentially increasing solution times. The sequential approach results are compared with the feasible solutions of integrated approach in terms of objective function values and run times. The percent gaps from the feasible solution are calculated as:

$$Gap\ \% = \frac{(Obj(\text{sequential approach}) - Obj(\text{integrated approach}))}{Obj(\text{integrated approach})} * 100,$$

where Obj(sequential approach) is the result obtained by optimally solving the models of sequential approach, and Obj(integrated approach) is the feasible objective function value of the model of integrated approach. Furthermore, the percent gaps from the optimal solutions are reported for the results of the integrated approach, which are obtained when the runtime limit is reached.

As shown in Table 6.6, feasible solutions cannot be obtained for two instances and optimal solutions cannot be obtained for all instances using integrated approach under a 1 hour runtime limit. However, feasible solutions are obtained for all instances using the sequential approach in less than 1 second, and these solutions are clearly better than the results obtained using the integrated approach.

Table 6.6 Comparison of the models for *K* = 15.

c	***r***	***p***	**# d-trips (*N*)**	**Sequential Runtime (sec.)**	**Integrated Runtime (sec.)**	**Gap %**	**Optimality Gap of Integrated Result %**
1	1	1	22	0.562	3600	-24.54	26.71
1	1	2	22	0.671	3600	-	-
1	2	1	21	0.688	3600	-0.26	15.48
1	2	2	23	0.782	3600	-	-
2	1	1	21	0.656	3600	-31.23	31.71
2	1	2	22	0.547	3600	-18.86	29.76
2	2	1	21	0.640	3600	-1.54	22.03
2	2	2	22	0.657	3600	-49.42	44.77

Hence, it can be concluded that larger instances of the complete problem cannot be solved within reasonable time limits using the integrated approach, due to exponentially increasing solution times. The sequential approach is clearly superior when the low solution times and quality solutions are considered.

7 CONCLUSION

In this thesis, both vehicle and crew scheduling problems are studied, where the objective is to determine the optimal number of different types of vehicles and crew members with a minimum operating cost to cover a given set of trips and corresponding deadheads subject to the working and spread time limitations of drivers. We treat first, vehicle and crew scheduling problems individually.

A Tactical Fixed Job Scheduling Problem (TFJSP)-based binary programming formulation is proposed for the vehicle scheduling problem considering multiple vehicle types as eligibility constraints. In the proposed formulation, the objective includes the minimization of total fuel consumption cost arising from deadheads and trips, as well as the fixed cost of vehicles. The additional eligibility constraints and the existence of additional objective function that minimizes total variable costs diverge our problem from a typical TFJSP, further complicating the structure. When the related literature is considered, our approach for the vehicle scheduling problem in this thesis is different from the existing vehicle scheduling and tactical fixed job scheduling studies in the literature due to: (1) fixed job scheduling approach to the vehicle scheduling problem; (2) eligibility constraints (multiple vehicle types); and (3) minimization of both fixed and variable costs. The performance of the developed model is investigated through a set of detailed experimentations and the numerical results are reported. The results show that the proposed model is quite effective in terms of solution time for instances with up to 120 trips. For all instances with up to 80 trips, the optimal solutions are obtained in less than 30 seconds. Particularly, optimal solutions of the instances with up to 40 trips are obtained in less than 2 seconds while the maximum solution time is around 4 minutes for instances with 120 trips.

A TFJSP-based binary programming model is proposed for crew scheduling problem as well considering different crew types with different capabilities as eligibility constraints and regarding operational constraints as working and spread time limitations, in the existence of sequence-dependent setup times. In the formulation, only processing times of tasks are considered as working time. In order to handle sequence dependent setup times in working time, an iterative valid inequality generation scheme is developed, which cuts off task sequences that exceed the total working time when setup times are included. When the related literature is considered, our approach for crew scheduling problem in this thesis is different from the existing crew scheduling and tactical fixed job scheduling studies in the related literature due to: (1) fixed job scheduling approach to the crew scheduling problem; (2) many realistic constraints such as working time, spread time and eligibility constraints; and (3) handling of the working time constraint in the existence of sequence-dependent setup times. The performance of the developed solution procedure is investigated through a detailed experimentation and the numerical results are reported. The results show that the proposed solution procedure is quite effective in terms of solution time for instances with up to 120 tasks. Particularly, the solution procedure returns optimal solutions for problem instances with up to 80 tasks within only seconds. It is worth noting that the optimal solutions are obtained with at most three iterations for all settings.

Furthermore, sequential and integrated approaches are proposed for the overall problem. As an integrated approach, a binary programming model is formulated and optimally solved for small-sized problem instances. However, larger instances cannot be solved within reasonable time limits due to exponentially increasing solution times. Therefore, a sequential approach is proposed. When the related literature is considered, the approaches for Vehicle and Crew Scheduling Problem (VCSP) in this thesis are different from the existing vehicle and crew scheduling studies in literature due to: (1) fixed job scheduling based sequential approach for VCSP; (2) binary programming model for integrated VCSP; (3) eligibility constraints for both vehicles and crew members; and (4) realistic constraints including working time and spread time for crew members. The performance of the sequential approach is investigated through a detailed experimentation, and results show that the sequential approach is quite efficient for instances with up to 120 trips. Additionally, the sequential approach is compared with the integrated one in terms of small sized instances and results indicate that sequential approach is quite effective in finding near optimal solutions within reasonable computation times.

As addressing for further studies, alternative operational constraints can be added into the formulations regarding more realistic representation of the problem. For example, the addition of part-time or split-shift drivers with corresponding costs may be a useful and practically meaningful extension of the problem. This extension can be handled by differentiating the fixed costs, as well as the working and spread times of different driver types. Alternatively, additional constraints on number and length of breaks can be included.

The trip/task times are assumed to be deterministic in this study. If the variability in trip/task durations due to traffic congestion or demand are taken into account, the crew scheduling, as well as the vehicle scheduling becomes much more complex. A simulation optimization approach may be worth studying as a future research topic when stochasticity is introduced into the problem.

Within the scope of this thesis, detailed experimentations are carried out using randomly generated instances and the performances of the proposed approaches are evaluated. For future work, computational studies are planned using real data obtained from the public bus transportation authority. Besides, the experimentations are carried out for instances with only up to 120 trips. For larger instances up to 1000 trips, metaheuristic approaches can definitely turn out to be quite useful in practice. Thus, the development of such heuristics is one of the fruitful research direction to follow.

REFERENCES

Arkin, E.M., Silverberg, E.B., 1987. Scheduling jobs with fixed start and end times. Discrete Applied Mathematics 18, 1–8.

Ball, M., Bodin, L., Dial, R., 1983. A matching based heuristic for scheduling mass transit crews and vehicles. Transportation Science 17, 4-31.

Bertossi, A.A., Carraresi, P., Gallo, G., 1987. On some matching problems arising in vehicle scheduling models. Networks 17, 271-281.

Bodin, L., Golden, B., Assad, A., Ball, M., 1983. Routing and scheduling of vehicles and crews: the state of the art. Computers & Operations Research 10 (2), 63-211.

Boschetti, M.A., Mingozzi, A., Ricciardelli, S., 2004. An exact algorithm for the simplified multiple depot crew scheduling problem. Annals of Operations Research 127, 177-201.

Bunte, S., Kliewer, N., 2009. An overview on vehicle scheduling models. Public Transport 1 (4), 299-317.

Carpaneto, G., Dell' Amico, M., Fischetti, M., Toth, P., 1989. A branch and bound algorithm for the multiple depot vehicle scheduling problem. Networks 19, 531-548.

Carraresi, P., Nonato, M., Girardi, L., 1995. Network models, Lagrangean relaxation and subgradient bundle approach in crew scheduling problem, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Daduna, J. R., Branco, I., Paixão, J.M.P. (Eds.). Springer, Berlin, Vol.430, pp. 188–212.

Ceder, A., 2007. Public Transit Planning and Operation: Theory, Modeling and Practice. Elsevier, Butterworth-Heinemann, pp. 4.

Costa, A., Branco, I., Paixão, J.M.P., 1995. Vehicle scheduling problem with multiple type of vehicles and a single depot, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Daduna, J. R., Branco, I., Paixão, J.M.P. (Eds.). Springer, Berlin, Vol.430, pp. 115–129.

Daduna, J. R., Mojsilovic, M., 1988. Computer aided vehicle and duty scheduling using HOT programme system, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Daduna, J. R. and Wren A. (Eds.). Springer-Verlag, Berlin, Vol.308, pp. 133-146.

Daduna, J.R., Paixão, J.M.P., 1995. Vehicle scheduling for public mass transit – an overview, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and*

Mathematical Systems". In: Daduna, J. R., Branco, I., Paixão, J.M.P. (Eds.). Springer, Berlin, Vol.430, pp. 76-90.

Darby-Dowman, K., Jachnik, J.K., Lewis, R.L., Mitra, G., 1988. Integrated decision support systems for urban transport scheduling: discussion of implementation and experience. , in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Daduna, J. R. and Wren A. (Eds.). Springer-Verlag, Berlin, pp. 226-239.

De Leone, R., Festa, P., Marchitto, E., 2011. Bus driver scheduling problem: a new mathematical model and a GRASP approximate solution. Journal of Heuristics 17(4), 441–466.

Desrochers, M., Soumis, F., 1989. A column generation approach to the urban transit crew scheduling problem. Transportation Science 23, 1–13.

Desrochers, M., Gilbert, J., Sauvé, M., Soumis, F., 1992. Crew-OPT: subproblem modelling in a column generation approach to urban crew scheduling, in "*Computer-Aided Transit Scheduling*". In: Desrochers, M., Rousseau, J. (Eds.). Springer, Berlin, pp. 395–406.

Dias, T.G., De Sousa, J.P., Cunha, F., 2002. Genetic algorithms for the bus driver scheduling problem: a case study. The Journal of the Operational Research Society 53(3), 324–335.

Eliiyi, D. T., Ornek, A., Karakütük, S. S., 2009. A vehicle scheduling problem with fixed trips and time limitations. International Journal of Production Economics 117, 150–161.

Eliiyi, D. T., Azizoğlu, M., 2009. A fixed job scheduling problem with machine-dependent job weights. International Journal of Production Research 47(9), 2231–2256.

Eliiyi, D. T., Azizoğlu, M., 2010. Working time constraints in operational fixed job scheduling. International Journal of Production Research 48(21), 6211–6233.

Eliiyi, D. T., Azizoğlu, M., 2011. Heuristics for operational fixed job scheduling problems with working and spread time constraints. International Journal of Production Economics 132(1), 107–121.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., Sier, D., 2004. An annotated bibliography of personnel scheduling and rostering. Annals of Operations Research 127, 21–144.

ESHOT, 2016**.** *Ulaşım Planlama Dairesi Başkanlığı, Monthly Status Report,* February 2016. (not publicly accessible, obtained from ESHOT İstatistik Şube Müdürlüğü)

Fischetti, M., Martello, S., Toth, P., 1987. The fixed job schedule problem with spread-time constraints. Operations Research 35, 849–858.

Fischetti, M., Martello, S., Toth, P., 1989. The fixed job schedule problem with working-time constraints. Operations Research 37, 395–403.

Fischetti, M., Lodi, A., Toth, P., 1999. A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. Technical Report, Universita di Bologna.

Forbes, M., Holt, J.N., Watts, A. M., 1994. An exact algorithm for multiple depot bus scheduling. European Journal of Operational Research 72, 115-124.

Freling, R., Boender, G., Paixão, A., 1995. An integrated approach to vehicle and crew scheduling. Report 9503/A, Erasmus University, Rotterdam, The Netherlands.

Freling R., 1997. Models and techniques for integrating vehicle and crew scheduling. Thesis Publishers, Amsterdam.

Freling, R., Wagelmans, P., Paixão, J.M.P., 1999. An overview of models and techniques for integrating vehicle and crew scheduling, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Wilson, N. (Ed.). Springer, Berlin, Vol. 471, pp. 441–460.

Freling, R., Huisman, D., Wagelmans, A., 2001a. Applying and integrated approach to vehicle and crew scheduling in practice, in "*Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems*". In: Voβ, S., Daduna, J.R. (Eds.). Springer, Berlin, Vol. 505, pp. 73–90.

Freling R., Wagelmans A., Paixão J.M.P., 2001b. Models and algorithms for single depot vehicle scheduling. Transportation Science 35(2), 165-180.

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

Friberg, C., Haase, K., 1999. An exact algorithm for the vehicle and crew scheduling problem, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Wilson, N. (Ed.). Springer, Berlin, Vol. 471, pp. 63–80.

Gavish, B., Shlifer, E., 1979. An approach for solving a class of transportation scheduling problems. European Journal of Operational Research 3, 122-134.

Gintner, V., Kliewer, N., Suhl, L., 2005. Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice. OR Spektrum 27(4), 507-523.

Gintner, V., Steinzen, I., Suhl, L., 2006. A time-space network based approach for integrated vehicle and crew scheduling in public transport, in "*Proceedings of the EURO Working Group on Transportation (EWGT) Joint Conferences 2006*", pp. 371-377.

Gintner, V., Kliewer, N., Suhl, L., 2008. A crew scheduling approach for public transit enhanced with aspects from vehicle scheduling, in "*Computer Aided Systems in Public Transport*". Springer, Berlin, Heidelberg, pp. 25-42.

GOAL Systems, 2016**.** GoalBus. www.goalsystems.com/en/goalbus/. (Access Date: August 2016)

Haase, K., Desaulniers, G., Desrosiers, J., 2001. Simultaneous vehicle and crew scheduling in urban mass transit systems. Transportation Science 35(3), 286–303.

Hadjar, A., Marcotte, O., Soumis, F., 2006. A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. Operations Research 54(1), 130-149.

Haghani, A., Banihashemi, M., 2002. Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints. Transportation Research Part A: Policy and Practice 36, 309-333.

Hassold, S., Ceder, A., 2014. Public transport vehicle scheduling featuring multiple vehicle types. Transportation Research Part B 67, 129-143.

Huisman, D., 2004. Integrated and dynamic vehicle and crew scheduling. PhD thesis, Erasmus University of Rotterdam, The Netherlands.

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

Irnich, S., 2008. Resource extension functions: properties, inversion and generalization to segments. OR Spectrum 30, 113-148.

IVU Traffic Technologies AG, 2016. IVU Suite. www.ivu.com/products-and-solutions/ivusuite.html. (Access Date: August 2016)

Kecskeméti, B., Bilics, A., 2013. Bus driver duty optimization using an integer programming and evolutionary hybrid algorithm. CEJOR 21, 745-755.

Kliewer, N., Mellouli, T., Suhl, L., 2006. A time-space network based exact optimization model for multi-depot bus scheduling. European Journal of Operational Research 175(3), 1616–1627.

Kliewer, N., Gintner, V., Suhl, L., 2008. Line change considerations within a time-space network based multi-depot bus scheduling model, in "*Computer-Aided Systems in Public Transport, Lecture Notes in Economics and Mathematical Systems*". In: Hickman, M., Mirchandani, P., Voβ, S. (Eds.). Springer, Berlin, Vol. 600, pp. 57-70.

Kliewer, N., Amberg, B., Amberg, B., 2012. Multiple depot vehicle and crew scheduling with time windows for scheduled trips. Public Transport 3 (3), 213-244.

Kolen, A.W.J., Lenstra, J.K., Papadimitriou, C.H., Spieksma, F.C.R., 2007. Interval scheduling: a survey. Naval Research Logistics 54, 530–543.

Kovalyov, M.Y., Ng, C.T., Cheng, T.C.E., 2007. Fixed interval scheduling: models, applications, computational complexity and algorithms. European Journal of Operational Research 178, 331–342.

Krishnamoorthy, M., Ernst, A.T., Baatar, D., 2012. Algorithms for large scale shift minimisation personnel task scheduling problems. European Journal of Operational Research 219, 34-48.

Kroon, L.G., Salomon, M., Van Wassenhove, L.N., 1997. Exact and approximation algorithms for the tactical fixed interval scheduling problem. Operations Research 4, 624–638.

Kwan, A.S.K, Kwan, R.S.K., Parker, M.E., Wren, A., 1999. Producing train driver schedules under different operating strategies, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Wilson, N. (Ed.). Springer, Berlin, Vol. 471, pp. 129-154.

Lenstra, J., Kan A.R., 1981. Complexity of vehicle routing and scheduling problems. Networks 11(2), 221-227.

Lourenço, H.R., Paixão, J.M.P., Portugal, R., 2001. Multiobjective metaheuristics for bus driver scheduling problem. Transportation Science 35(3), 331–343.

Löbel, A., 1997. Optimal vehicle scheduling in public transit. PhD Thesis, Technische Universitat Berlin.

Löbel. A., 1998. Vehicle scheduling in public transit and Lagrangean pricing. Management Science 44(12), 1637-1650.

Mesquita, M., Paixão, J.M.P., 1992. Multiple depot vehicle scheduling problem: a new heuristic based on quasi-assignment algorithms, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Desrochers, M., Rousseau, J.M., (Eds.). Springer, Berlin, Vol. 386, pp. 167-180.

Mesquita, M., Paixão, J.M.P., 1999. Exact algorithms for the multiple-depot vehicle scheduling problem based on multicommodity network flow type formulations, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Wilson, N. (Ed.). Springer, Berlin, Vol. 471, pp. 221-243.

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

Mesquita, M., Paias, A., Respicio, A., 2009. Branching approaches for integrated vehicle and crew scheduling. Public Transportation 1(1), 21-37.

Mesquita, M., Moz, M., Paias, A., Pato, M., 2013. A decomposition approach for the integrated vehicle-crew-roster problem with days-off pattern. European Journal of Operational Research 229(2), 318-331.

Nurmi, K., Kyngas, J., Post, G., 2011. Driver rostering for bus transit companies. Engineering Letters International Association of Engineers 19(2), 125-132.

Paixão, J. M., Branco, I., 1987. A quasi-assignment algorithm for bus scheduling. Networks 17, 249-269.

Pugliese, L., Guerriero, F., 2013. A survey of resource constrained shortest path problems: exact solution approaches. Networks 62(3), 183-200.

Riberio, C. C., and Soumis, F., 1994. A column generation approach to the multiple depot vehicle scheduling problem. Operations Research 42, 41-52.

Rousseau, J. M., Blais, J. Y., 1985. HASTUS: An interactive system for buses and crew scheduling, in "*Computer Scheduling of Public Transport 2*". In: Rousseau, J. M. (Ed.). North Holland, Amsterdam, pp. 473-491.

Rousseau, J., Desrosiers, J., 1995. Results obtained with CREW-OPT, a column generation method for transit crew scheduling. Lecture Notes in Economics and Mathematical Systems 430, 349–358.

Shen, Y., Kwan, R.S.K., 2001. Tabu search for driver scheduling, in "*Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems*". In: Voβ, S., Daduna, J.R. (Eds.). Springer, Berlin, Vol. 505, pp. 121–135.

Smith, B. M., Wren, A., 1988. A bus crew scheduling system using a set covering formulation. Transportation Research Part A 22, 97-108.

Steinzen, I., 2007. Topics in integrated vehicle and crew scheduling in public transit. PhD Thesis, University of Paderborn, Germany.

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

Toth, A., Kresz, M., 2013. An efficient solution approach for real word driver scheduling problems in urban bus transportation. Central European Journal of Operations Research 21(1), 75-94.

Wren, A., Rousseau, J. M., 1995. Bus driver scheduling – an overview, in "*Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*". In: Daduna, J. R., Branco, I., Paixão, J.M.P. (Eds.). Springer, Berlin, Vol. 430, pp. 173–187.

Zhou, S., Zhang, X., Chen, B. Velde, S., 2014. Tactical fixed job scheduling with spread time constraints. Computers & Operations Research 47, 53–60.

CURRICULUM VITAE

Hande Öztop was born in İzmir. She received her B.Sc. degree from Yaşar University and started her graduate studies in 2014. She is currently a research assistant in the Department of Industrial Engineering at Yaşar University, where she has been working since 2014.

During her graduate studies, she has taken many courses including Optimization Models and Algorithms, Probabilistic Analysis, Sequencing and Scheduling Theory, System Simulation, Logistics and Transportation Systems, Statistical Analysis and Heuristic Optimization. Her research interests include combinatorial optimization, public transportation operations, scheduling problems and vehicle routing problems.

Her publications are listed below:

Oztop, H., Turan, B., Eliiyi, D.T., Kandiller, L., An application of multi-compartment vehicle routing problem in a Turkish feed delivery company, *Proceedings of the 1st International Conference on Agrifood SCM & Green Logistics*, Vol. 1, Halkidiki, Greece, 71-80, May 2015.

Oztop, H., Eliiyi, D.T., Eliiyi, U., Timetabling of bus lines through discrete event simulation, *ISCO 2016, 4th International Symposium on Combinatorial Optimization-Book of Abstracts*, 79-80, Vietri sul Mare, Italy, May 2016.